

УДК 004.056.52

Способ и алгоритм поиска уязвимостей в протоколах объектов критической информационной инфраструктуры

Васинев Д. А. ✉, Соловьев М. В.

Академия Федеральной службы охраны Российской Федерации,
Орел, 302020, Российская Федерация

Постановка задачи. В связи с постоянным появлением новых угроз и уязвимостей значительные усилия специалистов по информационной безопасности посвящены разработке способов и алгоритмов поиска уязвимостей в телекоммуникационных протоколах, программном обеспечении и информационных системах. Таким образом, актуальной задачей является разработка автоматических и автоматизированных методов, уменьшающих рутинные функции специалиста по информационной безопасности, связанные с поиском уязвимостей. **Целью работы** является повышение эффективности поиска уязвимостей в протоколах стека TCP/IP для критической информационной инфраструктуры. **Используемые методы:** методы теории систем, теории вероятностей, стохастические методы поиска на основе Марковских цепей, генетические алгоритмы, методы и алгоритмы тестирования и технологии тестирования, поиска уязвимостей в программном обеспечении, обрабатывающем протокольные конструкции стека TCP/IP в критической информационной инфраструктуре. **Новизна:** разработанный способ, алгоритм и программное средство поиска уязвимостей в протоколах позволяют целенаправленно формировать тестовые конструкции на основе стохастического случайного поиска уязвимостей в заданном пространстве состояний протоколов, получать тестовые последовательности на основе Марковских цепей и модифицировать их генетическими алгоритмами. **Результат:** применение предлагаемого способа и алгоритмов поиска уязвимостей позволяет исследовать заданное пространство состояний протоколов, функционирующих в критической информационной инфраструктуре, уменьшить время формирования тестовых конструкций по сравнению с методом полного перебора. Благодаря особенностям разработанного алгоритма реализовано решение для поиска уязвимостей, возникающих из-за нарушения не только подачи непредусмотренного вида информации, но и структуры трафика, поступающего в обрабатывающую систему. **Теоретическая значимость:** данное исследование является вкладом в теорию информационной безопасности в области усовершенствования методов тестирования протокольных конструкций на основе стохастического случайного поиска в пространстве протоколов и их параметров, а также модификации тестовых последовательностей, получаемых методами генетических алгоритмов, и доведении предлагаемых решений до алгоритмов реализации. **Практическая значимость:** разработан универсальный автоматизированный программный фаззер для поиска уязвимостей программного обеспечения стека TCP/IP в критической информационной инфраструктуре, функционирующий на основе параметров пространства состояний целевого объекта.

Библиографическая ссылка на статью:

Васинев Д. А., Соловьев М. В. Способ и алгоритм поиска уязвимостей в протоколах объектов критической информационной инфраструктуры // Информационные технологии и телекоммуникации. 2024. Т. 12. № 2. С. 1–15. DOI: 10.31854/2307-1303-2024-12-2-01-15. EDN: XYFVBO

Reference for citation:

Vasinev D., Solovlev M. Method and Algorithm for Vulnerability Detection in the Protocols of Critical Information Infrastructure Objects. *Telecom IT*. 2024. Vol. 12. Iss. 2. PP. 1–15 (in Russian). DOI: 10.31854/2307-1303-2024-12-2-01-15. EDN: XYFVBO

Ключевые слова: фаззинг, тестирование уязвимостей, фаззинг протоколов, генетический алгоритм, динамический анализ, тестирование методом «черного ящика», стохастический случайный поиск, генетический алгоритм, методика тестирования, алгоритм тестирования

Введение

Распределенные информационные системы (ИС), информационно-телекоммуникационные сети (ИТС), автоматизированные системы управления технологическими процессами, элементы центров обработки данных (ЦОД) являются составными частями распределенных информационных инфраструктур организаций, выполняющих функции транспортировки данных, предоставления услуг с заданным качеством обслуживания, резервирования, отказоустойчивости и информационной безопасности. Для наиболее важных отраслей деятельности государства такие инфраструктуры получили статус критических информационных инфраструктур (КИИ), к которым предъявляются особые требования, в том числе и по обеспечению их информационной безопасности.

Анализ способов построения КИИ позволяет сделать вывод об иерархичности, вложенности таких конструкций, которые формируются на основе применения различных видов виртуальных частных сетей (VPN, аббр. от англ. Virtual Private Network) [1, 2]. Существующие воздействия нарушителя на объекты КИИ классифицируются Национальным координационным центром по компьютерным инцидентам (см. Бюллетени НКЦКИ: новые уязвимости ПО // Безопасность пользователей сети Интернет. URL: <https://safe-surf.ru/specialists/bulletins-nkcki>), при этом степень деструктивных действий в отношении коммуникационной инфраструктуры говорит о сетевых угрозах преимущественно высокого и критического уровней, проявляющихся с применением протоколов канального, сетевого, транспортного, прикладного уровней. Сложившиеся условия функционирования ИС, ИТС, ЦОД, когда вложенные иерархические конструкции являются основой построения коммуникационных инфраструктур на основе протоколов канального, сетевого, транспортного уровней, подвержены воздействию нарушителя. При этом в связи с разнообразием коммуникационных инфраструктур пространство состояний для воздействия нарушителя меняется в диапазоне от $2^{18+20+20}$ байт (Ethernet+IP+TCP/UDP) до 2^{36} байт (MPLS TE over GRE), что представляет собой достаточно большое пространство состояний для воздействия нарушителя на элементы коммуникационной инфраструктуры объекта КИИ. В заданных условиях функционирования объекта КИИ верификация его протокольных конструкций и политики информационной безопасности является актуальным направлением исследования.

Целью исследования является повышение эффективности поиска уязвимостей в протоколах стека TCP/IP за счет ограничения размерности пространства состояний, алгоритмов направленного случайного поиска в заданном пространстве параметров протоколов и снижения времени верификации протокольных конструкций в протоколах объектов КИИ.

Частные задачи при этом целесообразно разделить на две группы. К первой относится поиск уязвимостей в протокольных конструкциях, существующих алгоритмах работы программного обеспечения, функционирующего на канальном, сетевом, транспортном уровне, на основе методов тестирования протоколов, известных как фаззинг-тестирование. Вторая группа задач основана на таких методах тестирования протоколов, как верификации политики информационной безопасности объекта КИИ.

Анализ исследований в предметной области позволяет выделить отличительные признаки методов тестирования программного обеспечения, являющихся наиболее близкими аналогами для процесса тестирования протоколов [3]. По результатам анализа таковым является программное средство AFL (*аббр. от англ. American Fuzzy Lop* – не переводимое буквально название свободного программного фаззера, использующего генетические алгоритмы для эффективного увеличения покрытия кода тестовых случаев).

Обобщенное представление функциональных возможностей средства AFL сводится к формированию вектора тестовых посылок в пространстве параметров тестируемых протоколов. При этом формируется множество тестовых данных, случайное в пространстве параметров протокола, применяется мутационный подход к формированию тестовых данных. Пространство состояний процесса формирования вектора тестовых данных для рассматриваемого программного средства представлено на рисунке 1.

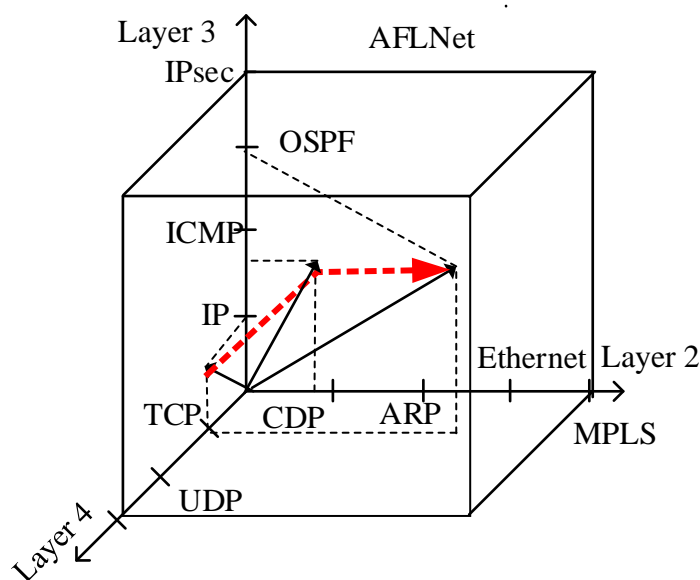


Рис. 1. Формирование вектора тестовых данных в пространстве состояний протокольных конструкций

Достаточно разнообразные функциональные возможности тестирования протоколов позволяют выделить и недостатки, связанные с неконтролируемым процессом случайного поиска. Поскольку пространство состояний протокольных конструкций достаточно большое, то такой подход может привести к про-

пуску значимых тестовых конструкций. Другим недостатком является несовершенство функции контроля пригодности тестового набора данных, контроля самого тестируемого объекта.

Программное средство работы с протоколами Scapy позволяет решать задачи, связанные с необходимостью формирования пакетов по определенным правилам. Анализ функциональных возможностей Scapy позволяет графически отобразить результаты его работы в форме плоскости для тестирования заданной области протоколов (рисунок 2).

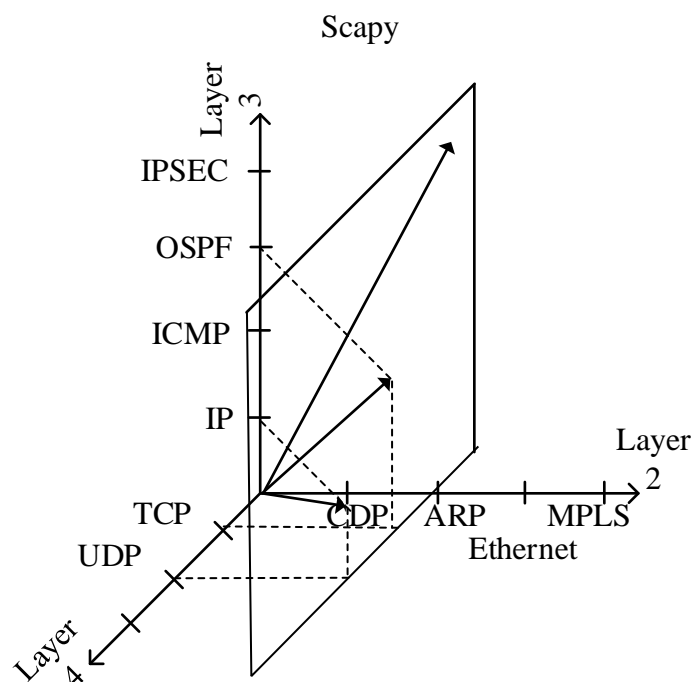


Рис. 2. Процесс тестирования пространства состояний протоколов в виде плоскости

Недостатками процесса формирования тестов в программном средстве Scapy является ограничение процедуры тестирования плоскостью параметров одного протокола.

Направлением развития рассматриваемого класса средств тестирования протоколов является расширение функциональных возможностей за счет:

- процедуры управления стохастическим случайным поиском как во множестве протоколов, так и их параметрах, как в автоматическом, так и автоматизированном режимах работы;
- применения генетических алгоритмов модификации параметров протоколов и применения методов направленного случайного поиска в пространстве состояний;
- контроля пригодности тестовых посылок, времени отклика принимаемых данных, заданного целевого тестируемого домена.

Графически процесс тестирования в форме направленного случайного поиска в пространстве состояний представлен на рисунке 3.

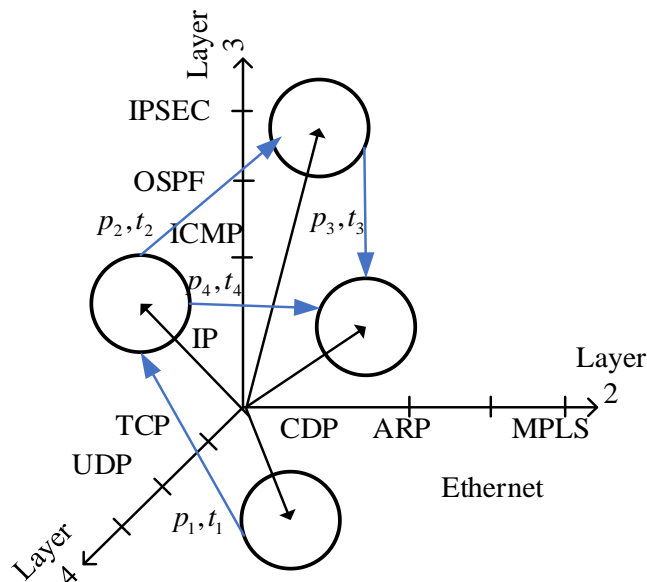


Рис. 3. Управление пространством состояний протоколов на основе параметров матрицы переходных вероятностей и времени пребывания в состоянии

Отличительным признаком предлагаемого решения является управление формированием тестовых посылок на основе аналитических методов (Марковских процессов) за счет управления параметрами переходных вероятностей и времени пребывания в заданном пространстве состояний. Применение предлагаемых аналитических методов допустимо как при формировании множества тестируемых протокольных блоков данных (PDU, аббр. от англ. Packet Data Unit – блок пакетных данных) – «Корпус», так и при формировании параметров, которые модифицируются в блоках «Генетический алгоритм», «Мутационный алгоритм» в функциональной модели, представленной на рисунке 4.

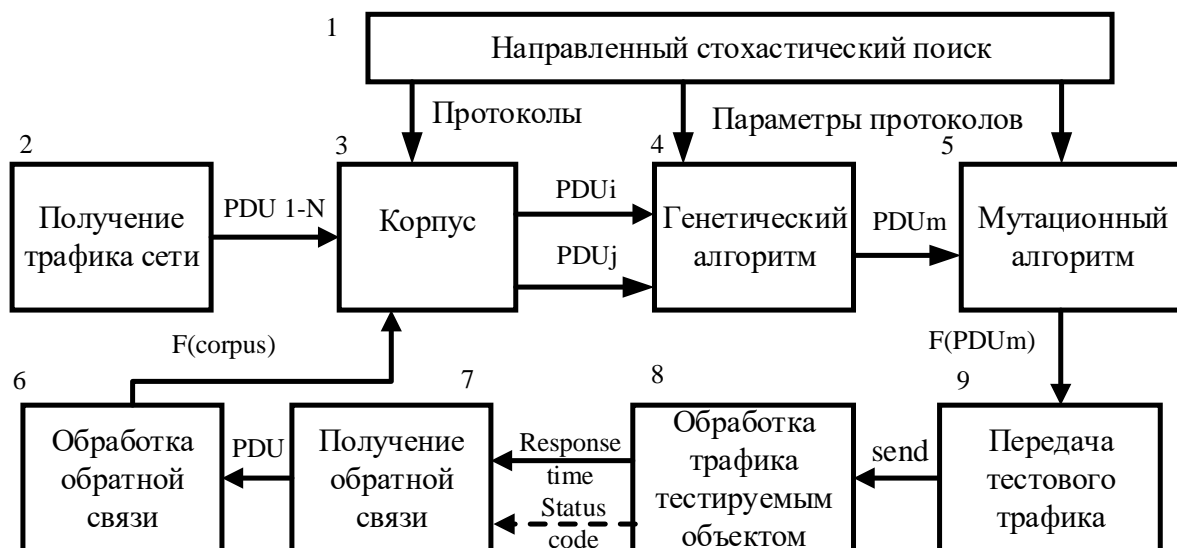


Рис. 4. Функциональная модель тестирования протоколов

Получение трафика сети (блок 2) осуществляется на основе анализа проходящих в сегменте сети данных. В блоке 3 формируется подмножество дан-

ных – «Корпус», – которое необходимо для выполнения процедур «Генетического алгоритма» (блок 4), а также «Мутационного алгоритма» (блок 5). Формирование множеств протоколов в блоке 3, выбор PDU для процедур генетического алгоритма в блоке 4, мутация PDU в блоке 5 осуществляются с учетом направленного стохастического случайного поиска (блок 1). Полученные тестовые пакеты (блок 9) отправляются на тестируемый объект (блок 8). Осуществляется контроль обратной связи для каждой группы тестовых пакетов (блок 7). Включение наиболее значимых пакетов в множество «Корпус» (блок 3) позволяет учитывать тестовые пакеты, имеющие максимальную полезность по критерию роста функции приспособленности.

Предлагаемая модель позволяет формировать множество тестовых пакетов с учетом направленного случайного поиска, а также учитывать в блоке 3 тестовые множества, дающие наибольший вклад в формирование значимых тестовых комбинаций на основе блока 7.

Концептуальные положения, заложенные в функциональной модели, легли в основу модификации эволюционного алгоритма [3] в виде алгоритма генерации данных на основе семантически верного подхода (рисунок 5) и с использованием популяции (рисунок 6).

Разработка способа и алгоритмов поиска уязвимостей программного обеспечения стека TCP/IP в критической информационной инфраструктуре

Для автоматизации процесса генерации тестовых фреймов предлагается работать с их заголовками в качестве базовых единиц. Сгенерированный пакет отправляется тестируемой системе, причем для реализации данной операции необходимо выбирать программные средства, позволяющие создавать «неправильные» по спецификации пакеты (изменение порядка следования заголовков, несоответствие фактической длины пакета и значения поля). Эмпирической функцией эффективности предлагается выбрать время ответа тестируемой системы, следовательно, чем больше тестируемый PDU влияет на время обработки PDU, время ответа, количество отправляемых PDU, тем более он эффективен.

В качестве исходных данных для развития способа тестирования был принят алгоритм реализации универсального тестирования на основе генетических методов [3]. В ходе практической реализации предложенного алгоритма в [3] было принято решение усовершенствовать его модернизацию в двух направлениях:

- 1) семантически неверный подход, реализуемый на основе алгоритма, представленного в [3];
- 2) семантически верный подход, представленный на рисунке 5.

При реализации семантически неверного метода предполагалось, что формирование протокольных конструкций, не соответствующих спецификации, может привести к реализации уязвимости, как, например, в Double Tagging Attack (атаке с двойным тегированием), на основе формирования PDU с двумя метками виртуальной сети с целью переноса данных в закрытый сегмент [4].

Однако эффективность предлагаемого решения мала, поскольку высок процент потерь пакетов, связанная с алгоритмом работы протокола. В связи с этим было принято решение модифицировать процесс генерации тестовых единиц данных в соответствии с алгоритмом, представленным на рисунке 5.

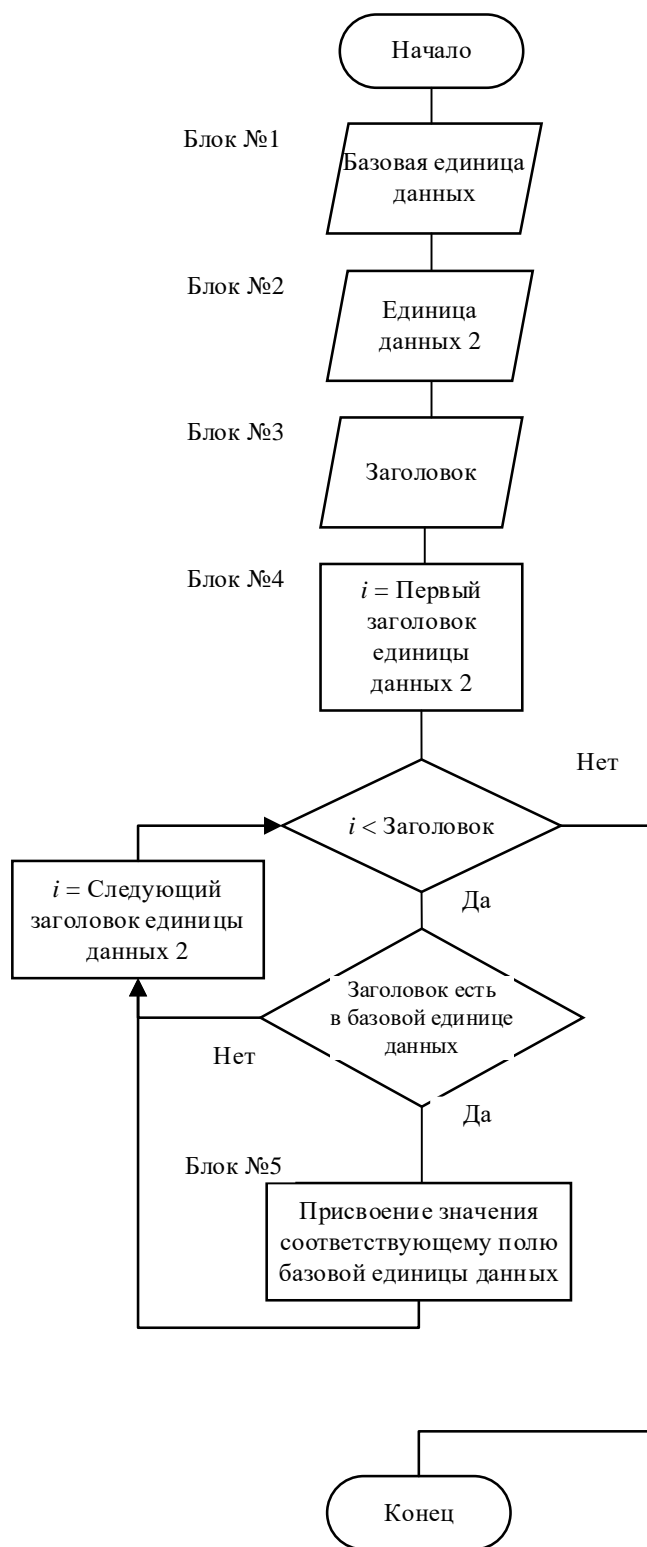


Рис. 5. Алгоритм генерации данных согласно эволюционному алгоритму, семантически верный подход

Алгоритм реализует мутацию, поэтому ему на вход подаются выбранные на предыдущих этапах единицы данных блоков №№ 1 и 2. Одна из этих конструкций считается базовой единицей данных блока № 1, значения заголовка которой будут изменены в процессе функционирования алгоритма мутации протокольных конструкций.

В целях ограничения параметров тестируемого протокола помимо единиц данных передается заголовок (блок № 3), вплоть до значений которого будет производиться переопределение значений полей единиц данных.

В ходе дальнейшей работы алгоритма перебираются значения полей единицы данных в блоке № 2 и, в случае если они также присутствуют в базовой единице данных, их значения присваиваются значениям заголовков базовой единицы данных (блок № 5). Следует также отметить, что адресные поля эта операция не затрагивает.

Одним из способов увеличения эффективности генетического фаззинга является преобразование корпуса тестирования (множества, содержащего исключительно тестовые единицы данных), в популяцию (множество тестовых единиц данных, каждой из которых соответствует значение функции приспособленности) [5].

Функция приспособленности (Fitness Function) [5] в контексте генетического фаззинга используется для оценки эффективности и качества сгенерированных единиц данных. Она помогает определить, насколько хорошо конкретный входной тестовый вектор способен вызывать нежелательное поведение тестируемой системы. В ходе реализации фаззера для функции приспособленности был выбран показатель – время ответа, которое при стабильной сетевой задержке позволит характеризовать время обработки тестовых последовательностей.

Таким образом, алгоритм работы фаззера с учетом функции приспособленности изменится, как показано на рисунке 6.

После захвата пакетов из них формируется первоначальная популяция, каждый пакет которой соотносится со значением функции приспособленности, (блок № 1). Из популяции в соответствии с этими значениями выбирается 2 PDU (блок № 2) для применения к ним алгоритма формирования тестовых данных (после инициализации популяции выбор будет равновероятен). Получившаяся после конкатенации полей тестовых данных PDU (блок № 3) и мутации полей PDU (блок № 4) конструкция отправляется к тестируемому объекту (блок № 5). При наличии ответного PDU высчитывается время ответа (блок № 6).

Предлагается использовать время ответа в качестве аргумента для функции приспособленности, допускается также, что значение функции равно ее аргументу. Последним этапом цикла тестирования является добавление полученного тестового PDU и соответствующего ему значения функции приспособленности в популяцию (блок № 8).

Помимо модификации генерации данных представляется возможным модифицировать способ выбора единиц данных из популяции с учетом вероятности нахождения уязвимости в протоколе или группе протоколов. В этом случае эксперименту, реализующему фаззинг-тестирование, нет необходимости тестировать все множество перехваченных протоколов. Отдельным протокольным конструк-

циям уделяется большее предпочтение: например, они более доступны возможному нарушителю. В таком случае возможно одно или несколько из этих предпочтений учесть при формировании политики тестирования (фаззинга).

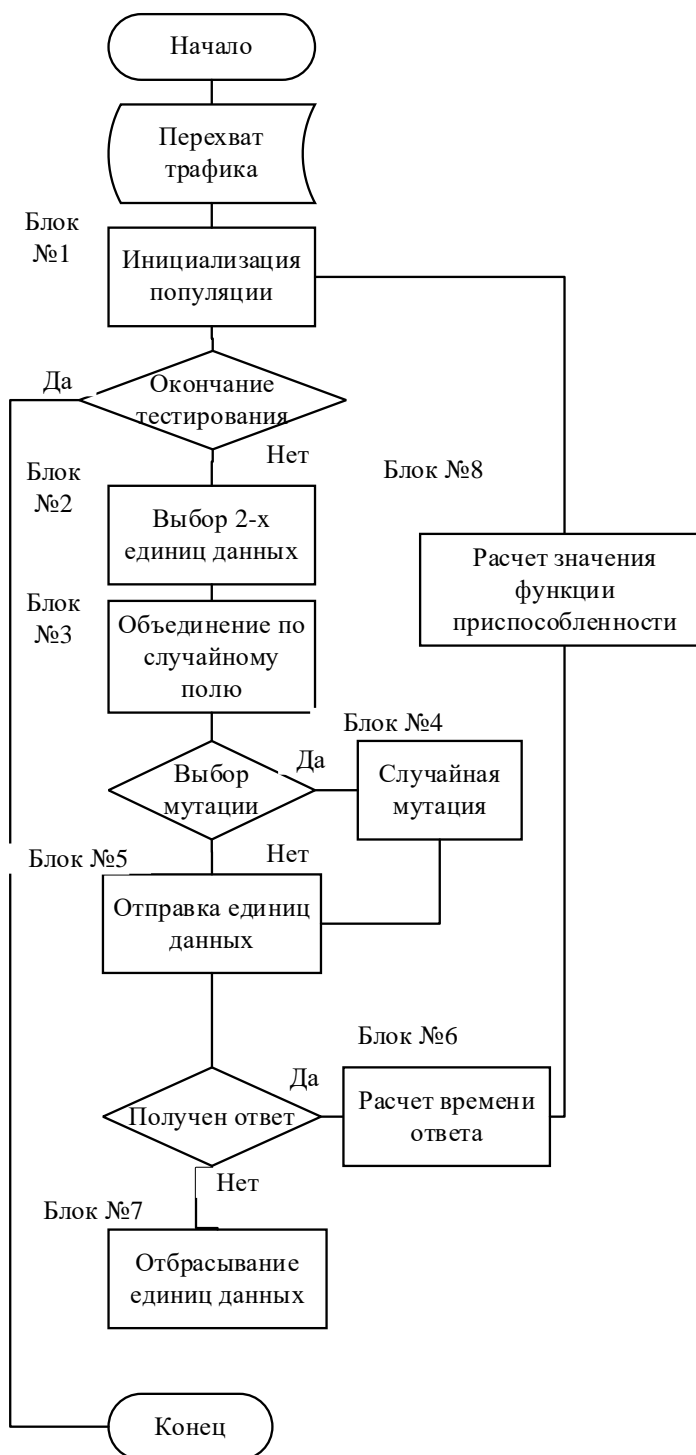


Рис. 6. Алгоритм генерации данных согласно эволюционному алгоритму с использованием популяции

Предположим, что злоумышленник тестирует протоколы, выбирая их в соответствии с предполагаемой вероятностью появления. Таким образом, существует возможность представить последовательность тестируемых протоколов

в виде дискретной стохастической цепи Маркова. Последовательность тестируемых протоколов (или поведение потенциального злоумышленника) может быть задана на основе известного распределения или предпочтений эксперта в матрице переходных вероятностей.

На первом этапе тестирования эксперт получает из перехваченного трафика набор протоколов, единицы данных которых содержатся в дампе. Путем задания каждому протоколу определенной вероятности появления эксперт изменяет значение функции приспособленности при инициализации популяции. Протокольные единицы с высокой долей вероятности в начальном состоянии таблицы получают большие значения функции приспособленности, следовательно, с наибольшей вероятностью будут подвергнуты мутации.

Единица протокольных данных представляет собой множество заголовков протокольных блоков данных:

$$PDU_n = \{header_1, header_2, \dots, header_n\},$$

где $header_n$ – заголовок PDU . Тогда начальная популяция состоит из множества принятых единиц протокольных данных:

$$POP_n = \{PDU_1, PDU_2, \dots, PDU_n\}.$$

Рассмотрим начальную популяцию, сформированную из перехваченных $PDUPOP_n = \{PDU_1, PDU_2, \dots, PDU_n\}$, и составим множество уникальных протоколов, используемых в перехваченном трафике:

$$Protocols = \{proto(PDU_1), proto(PDU_2), \dots, proto(PDU_n)\},$$

где $proto(PDU_n)$ – протокол соответствующей единицы данных.

Тогда сформируем множество вероятностей использования каждого выявленного протокола в трафике, где матрицей задают начальные состояния множества протоколов:

$$Prob = \{P(proto_1), P(proto_2), \dots, P(proto_n)\},$$

где $P(proto_n)$ – вероятность появления протокола $proto_n$.

Из $Prob$ выделим все уникальные протоколы, сформируем матрицу вероятностей начальных состояний для формирования тестового множества протоколов:

$$P_i^0 = \{P(proto_1), P(proto_2), \dots, P(proto_n)\}.$$

Сформируем матрицу переходных вероятностей для тестирования протоколов на основе известного распределения или экспертного решения о необходимости проверок в определенной области:

$$P_{i,j} = \begin{pmatrix} p_{1,1} & p_{1,2} & p_{1,j} \\ p_{2,1} & p_{2,2} & p_{1,j} \\ p_{i,1} & p_{i,2} & p_{i,j} \end{pmatrix},$$

где $p_{i,j}$ – матрица переходных вероятностей протоколов для формирования множества единиц данных для тестирования.

Тогда следующее состояние, позволяющее выбрать протоколы для формирования тестовых множеств, будет получено на основе выражения:

$$P_{i,j}(k) = \sum_{i=1}^n P_i(k-1) \cdot P_{i,j}^k,$$

где $P_{i,j}^k$ – значение переходных вероятностей для k шага.

Анализ предметной области позволил сделать вывод о том, что существующие инструменты тестирования используют порождающий или мутационный методы генерации тестовых данных. В связи с этим было предложено дополнить уже существующие методики, реализовав генетический алгоритм. Метод генерации данных предлагается улучшить путем реализации функции приспособленности по обратной связи, эффективность тестирования предполагается увеличить благодаря изменению способа выбора единиц данных из популяции со случайного на стохастический дискретный контролируемый выбор протокола с учетом предполагаемой оператором вероятности нахождения уязвимости в протоколе или группе протоколов. Направлением дальнейшего развития является написание программного фаззера.

Разработка программного обеспечения для поиска уязвимостей в протоколах стека TCP/IP в критической информационной инфраструктуре

Итогом исследования стала практическая реализация экземпляра фаззера [6]. В качестве языка программирования был выбран Python, в качестве фреймворка для низкоуровневого взаимодействия с протокольными конструкторами – Scapy; данный выбор позволил реализовать: автоматический расчет контрольных сумм при изменении данных в перехваченных единицах данных, фаззинг полей протоколов, основанный на их типе данных, составление не соответствующих спецификациям протокольных конструкций, поддержку большого числа поддерживаемых протоколов.

В ходе работы над проектом на основе предоставляемой фреймворком возможности действий с протокольными единицами данных как с объектами языка программирования была разработана первая известная специализированная библиотека для проведения универсального многопротокольного фаззинг-тестирования, в полном объеме предоставляющая разработчикам функционал, необходимый для создания фаззера протоколов, а также на ее основе был разработан экземпляр программы для тестирования протоколов и проведено тестирование, выявившее ранее известную уязвимость реализации протокола 802.1Q (Double Vlan Tagging).

В модуле для Python реализованы функции, представленные в таблице 1. Программный код отдельного модуля программы для тестирования поддается масштабированию, позволяет реализовать тестирование согласно разработанным алгоритмам и нетребователен к квалификации инженерного персонала.

Таблица 1 – Реализованный функционал в программном модуле

Функциональность	Название метода	Реализуемые действия
Перехват трафика для получения множества «Корпус»	intercept	Перехват исходящего с тестирующего устройства системы трафика
	sniff	Копирование сетевого трафика
	AsyncSniffer	Асинхронное копирование сетевого трафика
	Sniff_and_bridge	Перехват трафика из одного сетевого интерфейса и передача в другой после модификации
Мутационный фаззинг	scapy_def_rand	Функционал функции фаззинга из фреймворка Scapy
	scapy_fix_rand	Измененный функционал scapy_def_rand для корректной работы с перехваченным трафиком
	raw_mutations	Реализация случайной мутации из AFL
Генетический фаззинг	crossover	Кроссовер PDU по случайному байту (семантически неверный)
	layer_crossover	Кроссовер заголовков PDU (семантически верный)

В ходе проверки эффективности был реализован программно-аппаратный стенд, представленный на рисунке 7. Целью исследования было тестирование передачи данных между виртуальными сетями, не предусматривающими политики информационной безопасности сетевого оборудования.

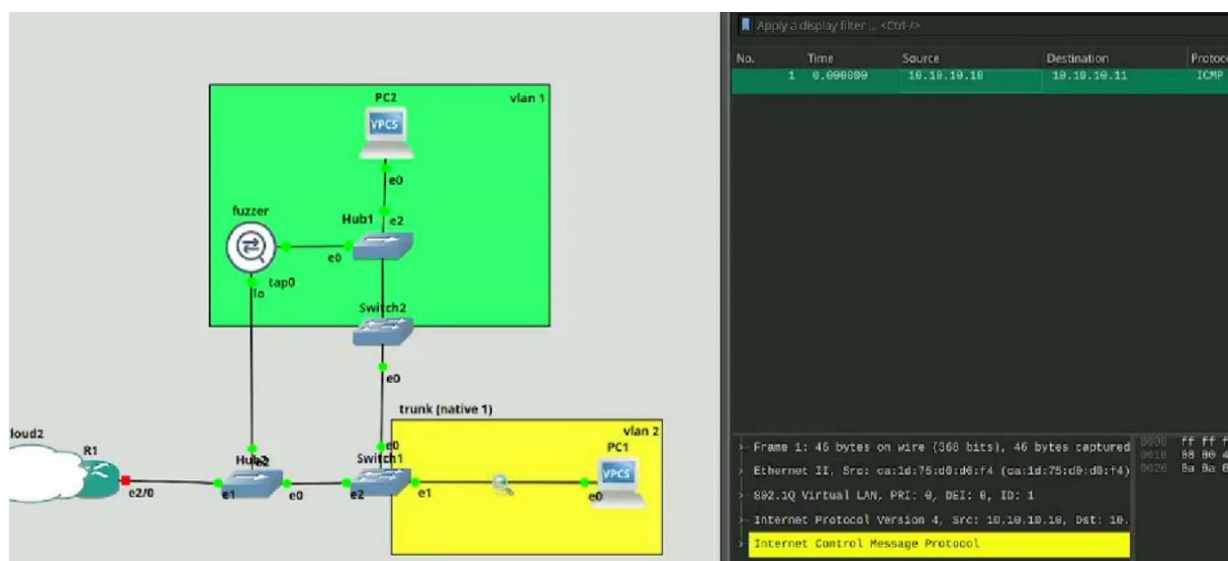


Рис. 7. Сгенерированный в первой виртуальной сети трафик попадает во вторую виртуальную сеть

Согласно эксперименту, программа копировала фрейм 802.1Q из первой виртуальной сети и с помощью генетического алгоритма генерации данных формировала тестовый фрейм, нарушающий политику безопасности и переходящий из второй виртуальной сети в первую, реализуя таким образом сетевую атаку. Из рисунка следует, что тестовый трафик, сформированный программным средством, перенаправляется в другую виртуальную сеть, создавая предпосылки для атак нарушителя широкого класса, такие как отказ в обслуживании, перенаправление трафика.

По результатам исследования успешно реализован и проверен алгоритм формирования тестовых запросов; разработано средство тестирования и проверено путем поиска уязвимостей в стандарте 802.1Q; помимо программы-фаззера разработан модуль на языке программирования Python, существенно упрощающий работу по управлению, модификации алгоритма функционирования и дальнейшему улучшению фаззера, в том числе и инженерным персоналом.

Разработанное программное средство функционирует на основе подключения к тестируемому объекту как в локальной, так и в распределенной сетях. Данное свойство успешно продемонстрировано в ходе эксперимента, ожидаемый результат получен в полном объеме.

В ходе дальнейшей работы разработан способ и программно-аппаратный комплекс для оценки защищенности телекоммуникационного и оконечного оборудования КИИ с использованием алгоритма стохастического случайного поиска в заданном пространстве состояний на основе генетического алгоритма и цепей Маркова [7].

Выводы

В работе исследованы современные средства фаззинга протоколов. Их анализ показал, что тестирование в общем случае предполагает работу исключительно с исходным кодом объекта. Используемые решения не предполагают автоматического тестирования и требуют тщательной предварительной настройки. Также было отмечено отсутствие программных модулей, облегчающих разработку средств фаззинг-тестирования на языке программирования – Python.

Разработан и модифицирован с использованием функции приспособленности алгоритм генетической генерации тестовых данных для фаззинга. В целях повышения эффективности формирования тестовых последовательностей предлагается применять математический аппарат цепей Маркова для стохастического случайного формирования тестовых последовательностей протокола, что позволяет использовать как автоматический, так и автоматизированный режимы работы.

Разработан и апробирован модуль для проведения фаззинг-тестирования. Верификация реализованной программы-фаззера успешно обнаружила уязвимость на основе протокола 802.1Q на виртуальном программно-аппаратном стенде.

Разработан программно-аппаратный комплекс для оценки защищенности телекоммуникационного и оконечного оборудования КИИ, позволяющий проводить автоматическое и автоматизированное тестирование объектов сетевой инфраструктуры вплоть до подсетей.

Направлениями дальнейших исследований являются: поиск эффективного способа определения статуса загрузки тестируемой системы по обратной связи; верификация программно-аппаратного фаззера на коммуникационной инфраструктуре различного типа, поиск CVE; разработка эргономичного пользовательского интерфейса.

Литература

1. Запечников С. В., Милославская Н. Г., Толстой А. И. Основы построения виртуальных частных сетей: учебное пособие для вузов. 2-е изд. М.: Горячая линия-Телеком, 2011. 249 с.

2. Захватов М. А. Построение виртуальных частных сетей на базе технологии MPLS. М.: Cisco Systems, 2001. 52 с.

3. Васинев Д. А., Соловьев М. В. Предложения по построению универсального фаззера протоколов // Труды учебных заведений связи. 2023. Т. 9. № 6. С. 59–67. DOI: 10.31854/1813-324X-2023-9-6-59-67. EDN: AABMEE

4. Bull R., Matthews J., Trumbull K. VLAN Hopping, ARP Poisoning & Man-In-The-Middle Attacks in Virtualized Environments. 2016. URL: <https://infocon.org/cons/DEF%20CON/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEF%20CON%2024%20-%20Bull-Matthews-Trumbull-VLAN-Hopping-ARP-MITM-in-Virtualized-UPDATED.pdf>

5. Andreas Z., Rahul G., Marcel B. The Fuzzing Book. Tools and Techniques for Generating Software Tests. URL: <https://www.fuzzingbook.org> (дата обращения 06.09.2024).

6. Васинев Д. А., Соловьев М. В., Бочков М. В., Сизых В. В. Программа для фаззинг-тестирования сетевых протоколов // Свидетельство о регистрации программы для ЭВМ RU 2024615254, опубликовано 05.03.2024. EDN: RENOCZ

7. Васинев Д. А., Соловьев М. В., Бочков М. В., Кирьянов А. В., Полехин А. А. и др. Программно-аппаратный комплекс для оценки защищенности телекоммуникационного и оконечного оборудования критической информационной инфраструктуры // Патент на изобретение RU2831928C1, опубликовано 16.12.2024.

Статья поступила 27 ноября 2024 г.

Одобрена после рецензирования 10 декабря 2024 г.

Принята к публикации 23 декабря 2024 г.

Информация об авторах

Васинев Дмитрий Александрович – кандидат технических наук, сотрудник Академии Федеральной службы охраны Российской Федерации. E-mail: vda33@academ.msk.rsnet.ru

Соловьев Михаил Викторович – сотрудник Академии Федеральной службы охраны Российской Федерации. E-mail: saintsdertr@gmail.com

Method and Algorithm for Vulnerability Detection in the Protocols of Critical Information Infrastructure Objects

D. Vasinev ✉, M. Solovev

Academy of the Federal Guard Service of the Russian Federation,
Oryol, 302020, Russian Federation

Problem statement. Due to the constant emergence of new threats and vulnerabilities, significant efforts of IS specialists are devoted to the development of methods and algorithms for finding vulnerabilities in telecommunication protocols, software, and information systems. Thus, the actual task is to automate the process of vulnerability search, to develop automatic and automated methods that reduce the routine functions of the IS specialist associated with the search for vulnerabilities. The aim of the work is to increase the efficiency of vulnerability search in TCP/IP stack protocols for critical information infrastructure. Methods used: methods of systems theory, probability theory, stochastic search methods based on Markov chains, genetic algorithms, methods and algorithms of testing and testing technologies, and search for vulnerabilities in software that processes protocol constructs of TCP/IP stack for critical information infrastructure. Novelty: a method, algorithm and software tool for searching for vulnerabilities in protocols, allowing to form test constructs based on stochastic random search for vulnerabilities, have been developed. The proposed solution allows to purposefully form test constructs in a given state space of protocols, obtain test sequences based on Markov chains, and modify them with genetic algorithms. **Result:** The application of the proposed method and algorithms of vulnerability search allows to investigate a given state space of protocols functioning in CII and allows to reduce the time of formation of test designs in comparison with the method of complete enumeration. Due to the peculiarities of the developed algorithm, the solution for the search for vulnerabilities based on the violation of the structure of the incoming traffic to the processing system, and not only the supply of an unintended type of information, is realized. **Theoretical significance:** contribution to the theory of information security in the field of improving methods of testing protocol designs based on stochastic random search in the space of protocols and their parameters, as well as modification of test sequences by methods of genetic algorithms, bringing the proposed solutions to implementation algorithms. **Practical significance:** a universal automated software fuzzer for searching vulnerabilities of TCP/IP stack software in critical information infrastructure has been developed, functioning on the basis of state space parameters of the target object.

Key words: fuzzing, vulnerability testing, protocol fuzzing, genetic algorithm, dynamic analysis, black box fuzzing, stochastic random search, genetic algorithm testing methodology, testing algorithm

Information about Authors

Vasinev Dmitry – Ph. D. of Engineering Sciences, employee (Academy of the Russian Federal Guard Service). E-mail: vda33@academ.msk.rsnet.ru

Solovev Mikhail – employee (Academy of the Russian Federal Guard Service). E-mail: saintsdertr@gmail.com