ENGENSEC
Educating the Next generation experts in Cyber Security

Tempus

The project has been funded by the European Commission. The Education, Audiovisual and Culture Executive program (EACEA), TEMPUS IV. The content of this presentation reflects the opinion of the author.

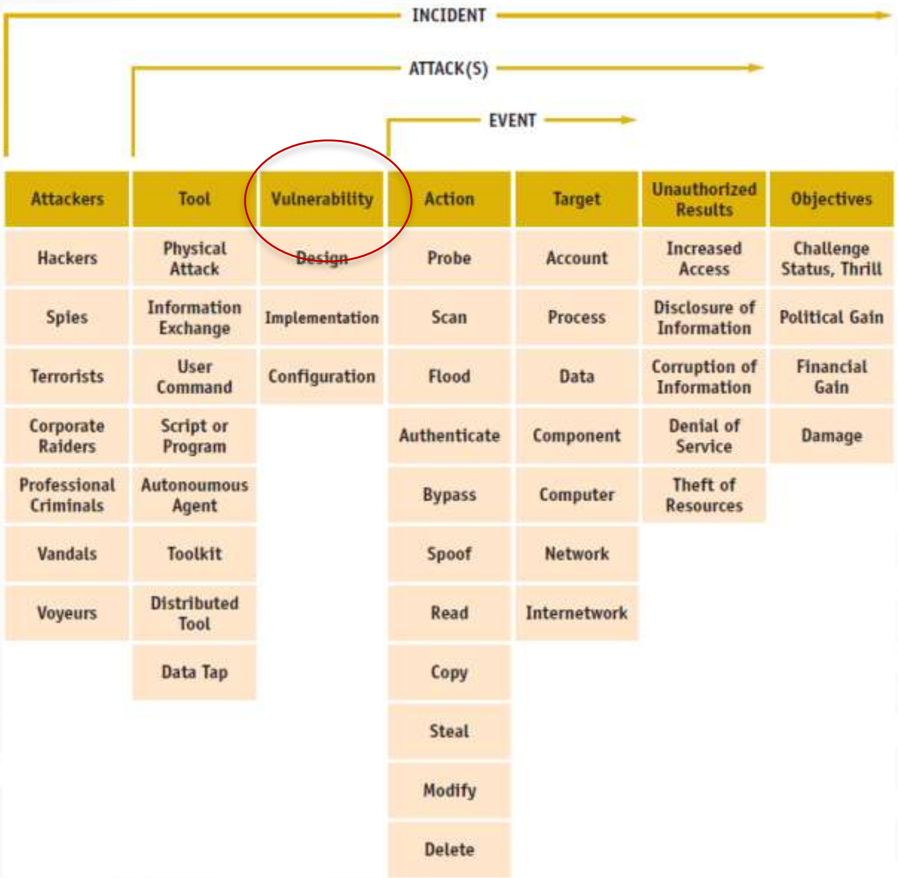
Module 3. Vulnerability Analysis

Penetration testing course

Vulnerability

A security vulnerability is a weakness in a product that could allow an attacker to compromise the integrity, availability, or confidentiality of that product.

Place of vulnerability



Attackers	Tool	Vulnerability	Action	Target	Unauthorized Results	Objectives
Hackers	Physical Attack	Design	Probe	Account	Increased Access	Challenge Status, Thrill
Spies	Information Exchange	Implementation	Scan	Process	Disclosure of Information	Political Gain
Terrorists	User Command	Configuration	Flood	Data	Corruption of Information	Financial Gain
Corporate Raiders	Script or Program		Authenticate	Component	Denial of Service	Damage
Professional Criminals	Autonoumous Agent		Bypass	Computer	Theft of Resources	
Vandals	Toolkit		Spoof	Network		
Voyeurs	Distributed Tool		Read	Internetwork		
	Data Tap		Copy			
			Steal			
			Modify			
			Delete			

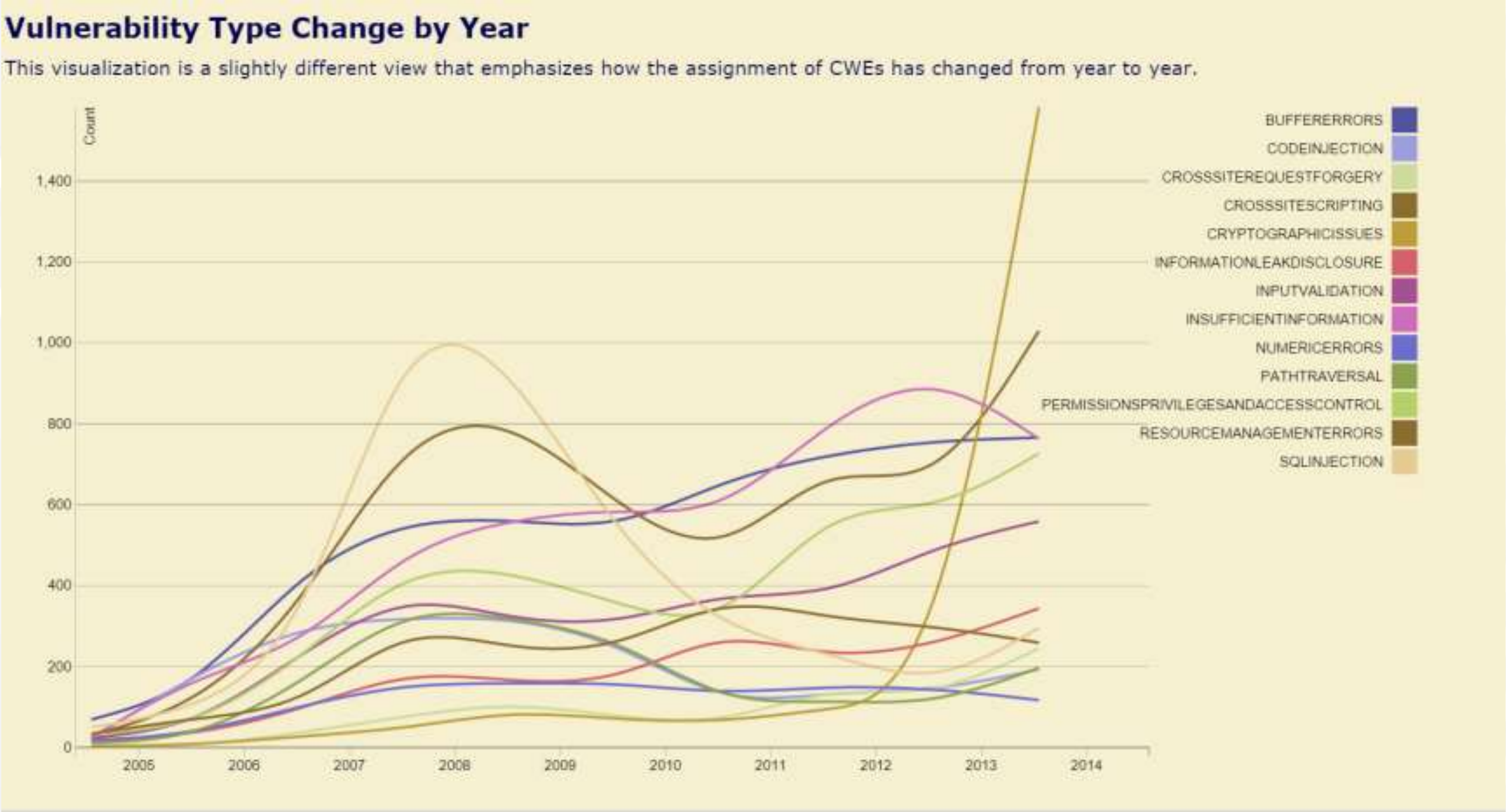
source: https://www.enisa.europa.eu/activities/cert/support/incident-management/ browsable/incident-handling-process/incident-taxonomy/existing-taxonomies

Types of vulnerabilities

Common Weakness Enumeration:

http://cwe.mitre.org/data/index.html

Vulnerability Type Change by Year



Vulnerability Type Change by Year

This visualization is a slightly different view that emphasizes how the assignment of CWEs has changed from year to year.

Buffer overflow: code

```
void foo(char *s) {
  char buf[10];
  strcpy(buf,s);
  printf(“buf is %s\n”,s);
}
…
foo(“thisstringistoolongforfoo”);
```

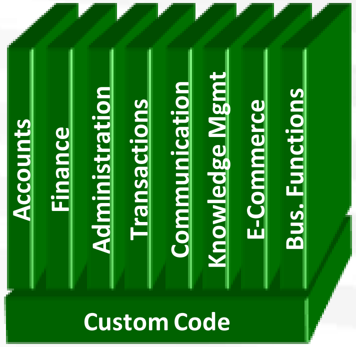
Buffer overflow: exploitation

- The general idea is to give to the program very large strings that will overflow a buffer.
- Result: crash or running of our code.

Session hijacking

1 User sends credentials

www.boi.com?JSESSIONID=9FA1DB9EA...



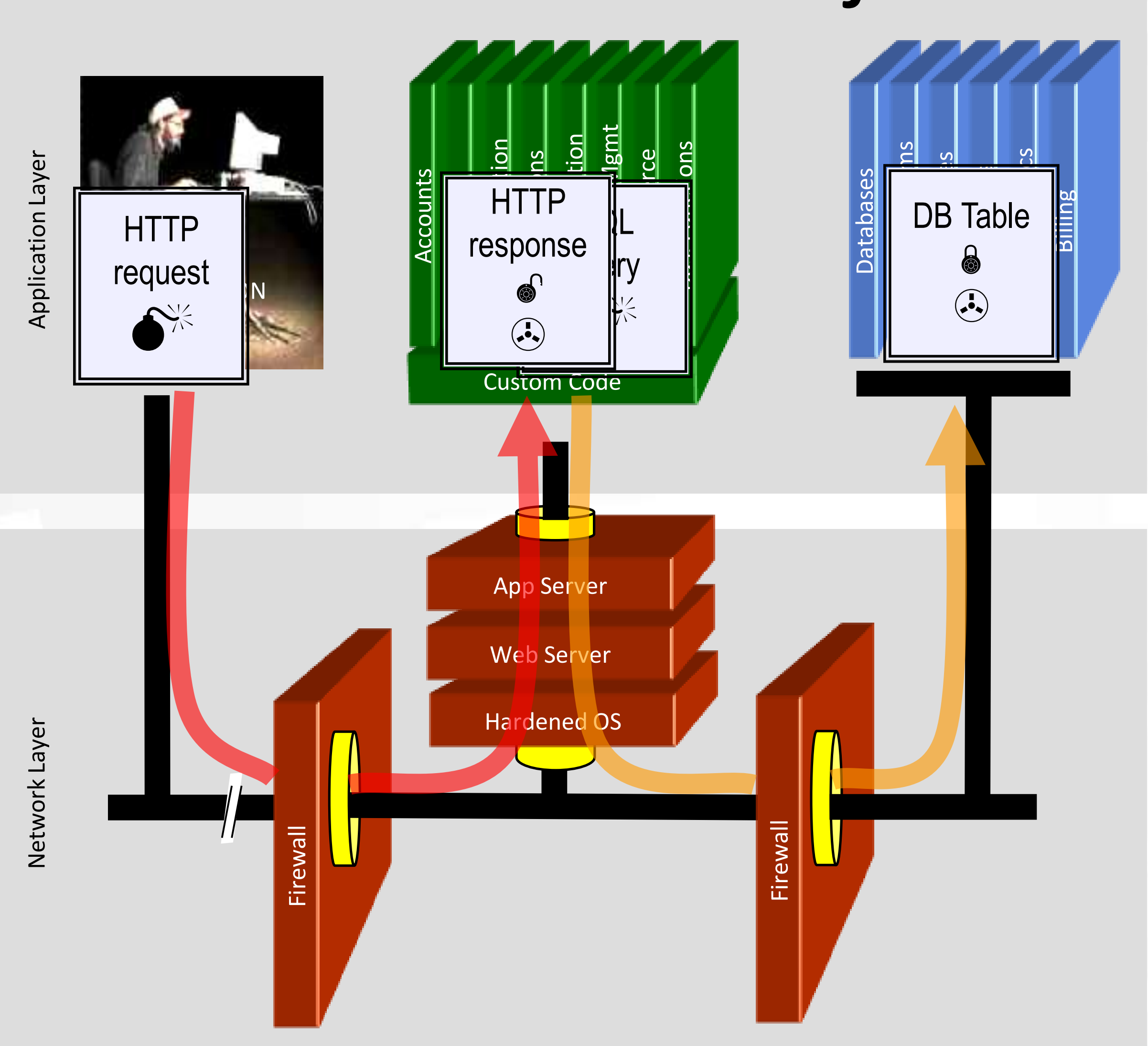
2 Site uses URL rewriting (i.e., put session in URL)

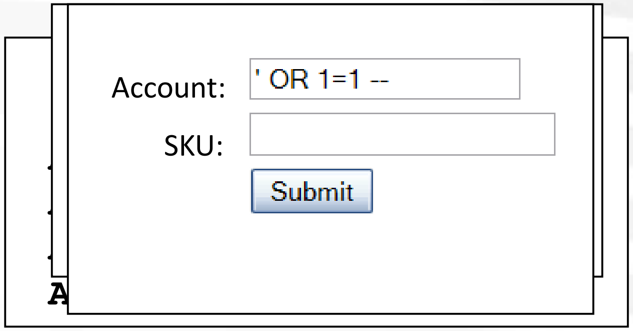
3 User clicks on a link to http://www.hacker.com in a forum

4 Hacker checks referrer logs on www.hacker.com and finds user's JSESSIONID

5 Hacker uses JSESSIONID and takes over victim's account

SQL-injection





Account: ' OR 1=1 --

SKU:

Submit

1. Application presents a form to the attacker

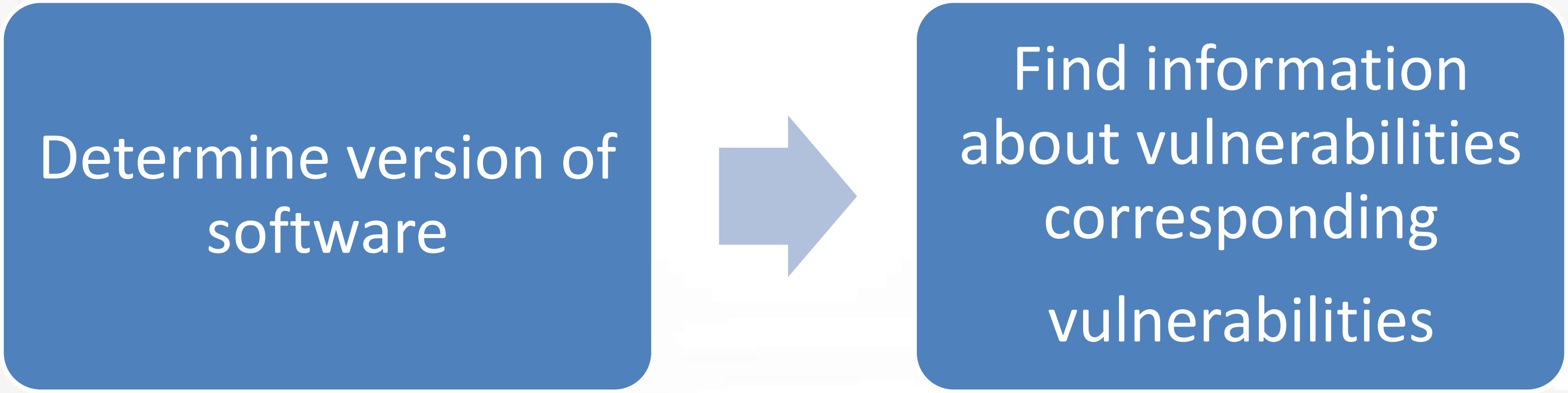
2. Attacker sends an attack in the form data

3. Application forwards attack to the database in a SQL query

4. Database runs query containing attack and sends encrypted results back to application

5. Application decrypts data as normal and sends results to the user

Manual search for known vulnerabilities

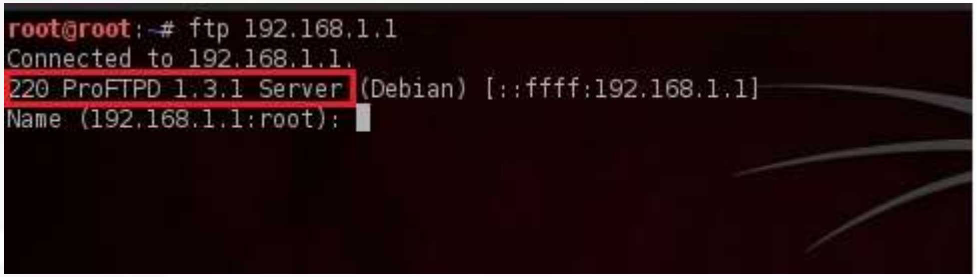


Determine version of software

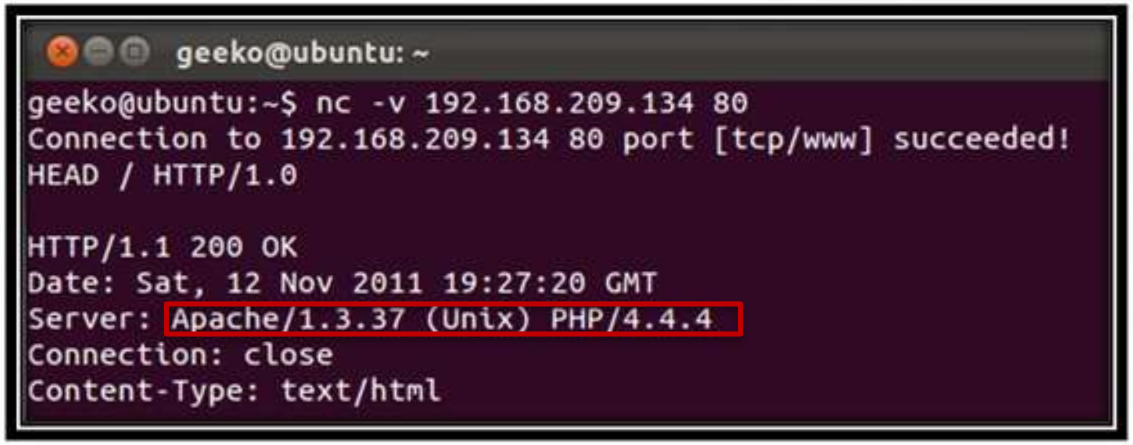
Find information about vulnerabilities corresponding

vulnerabilities

Banners: source of version info



```
root@root:~# ftp 192.168.1.1
Connected to 192.168.1.1.
220 ProFTPD 1.3.1 Server (Debian) [::ffff:192.168.1.1]
Name (192.168.1.1:root):
```



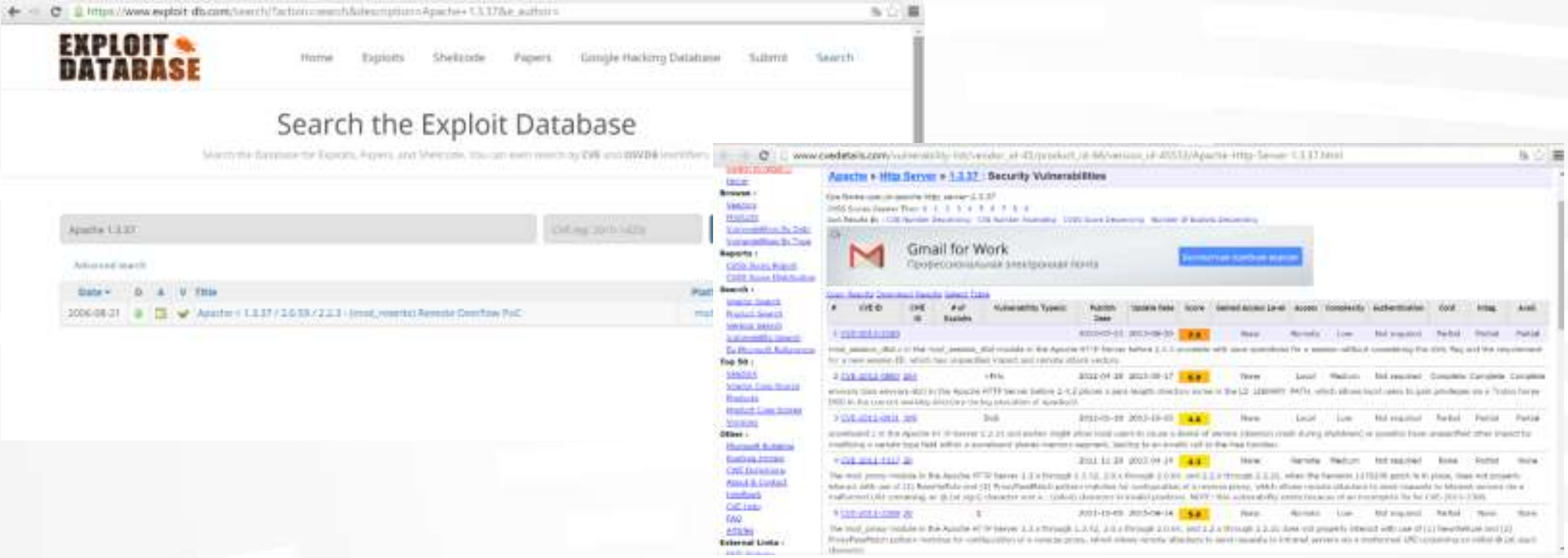
```
geeko@ubuntu:~$ nc -v 192.168.209.134 80
Connection to 192.168.209.134 80 port [tcp/www] succeeded!
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Sat, 12 Nov 2011 19:27:20 GMT
Server: Apache/1.3.37 (Unix) PHP/4.4.4
Connection: close
Content-Type: text/html
```

Additional sources of version info

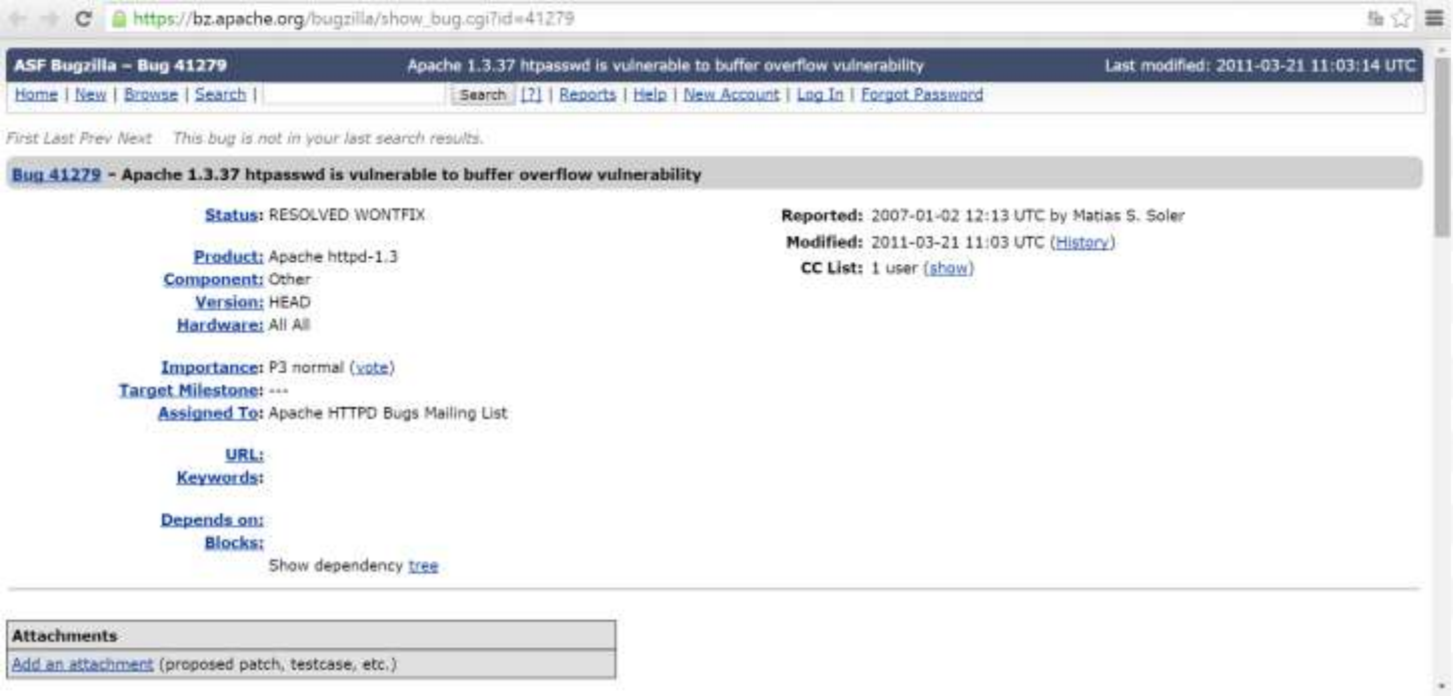
- HTTP-headers.
- Information on web-page: CMS version for example
- Technical pages for debugging: phpinfo.php
- Error messages
- Press-releases issued by vendors or suppliers
- Information containing in CV of IT-specialists (LinkedIn).

Information about vulnerabilities (1)



Online databases with vulnerabilities/exploits

Information about vulnerabilities (2)



https://bz.apache.org/bugzilla/show_bug.cgi?id=41279

ASF Bugzilla – Bug 41279 Apache 1.3.37 htpasswd is vulnerable to buffer overflow vulnerability Last modified: 2011-03-21 11:03:14 UTC

Home | New | Browse | Search | Search [?] | Reports | Help | New Account | Log In | Forgot Password
First Last Prev Next

This bug is not in your last search results.

Bug 41279 - Apache 1.3.37 htpasswd is vulnerable to buffer overflow vulnerability

Status: RESOLVED WONTFIX

Product: Apache httpd-1.3
Component: Other
Version: HEAD
Hardware: All All

Importance: P3 normal (vote)
Target Milestone: ---
Assigned To: Apache HTTPD Bugs Mailing List

URL:
Keywords:

Depends on:
Blocks:
Show dependency tree

Reported: 2007-01-02 12:13 UTC by Matias S. Soler
Modified: 2011-03-21 11:03 UTC (History)
CC List: 1 user (show)

Attachments
Add an attachment (proposed patch, testcase, etc.)

Vendor’s site

Port scanning

Port scanning

- Attackers wish to discover services they can break into.
- sending a packet to each port, one at a time.
  - Based on the type of response, an attacker knows if the port is used.
  - The used ports can be probed further for weakness.

Port numbers

- Port number is an address of service on particular host
- Part of UDP and TCP packets
  - UDP and TCP port numbers are disjoint
  - Typical to use the same port number for both UDP and TCP service
  - E.g., 80/TCP and 80/UDP for www
- 16-bit unsigned integer
- Well Known Ports (0 .. 1023)
- Registered Ports (1024 .. 49151)
- Dynamic and/or Private Ports (49152 .. 65535).
- http://www.iana.org/assignments/ port-numbers

Well Known: 0 - 1023

- Only root-privileged programs are allowed to open these ports.
- Examples
    - ftp-data 20/udp
    - ftp 21/tcp
    - ssh 22/tcp
    - telnet 23/tcp
    - Time 37/tcp
    - Time 37/udp
    - Whois 43/tcp
    - Imap 143/tcp

Registered: 1024 ..49151

- Ordinary programs/users can use these
- shockwave2 1257/tcp Shockwave 2 shockwave2 1257/udp Shockwave 2
- x11 6000-6063/tcp X Window System x11 6000-6063/udp X Window System

Dynamic/Private: 49152 .. 65535

- Ordinary programs can use these

State of a Port

- Open
  - A service process is listening at the port. The OS receives packets arriving at this port and gives the messages to the service process. If the OS receives a SYN at an open port, this is the first packet of the three way handshake.
- Closed
  - No process is listening at the port. If the OS receives a SYN at a closed port, an RST is sent.
- Filtered
  - A packet filter is listening at the port and blocks the communication.

TCP connect(0) scanning

- Try connect()-ing to every port
  - If the port is listening, connect() will succeed.
  - Otherwise, the port isn’t reachable.
- No need for any special privileges. Any user can use it.
- Speed - slow.
- Scanner can be identified.

TCP SYN scanning

- Often referred to as half-open scanning.
  - Send a SYN packet
  - Wait for a response.
- A SYN/ACK indicates the port is listening.
- If a SYN/ACK is received, send an RST to tear down the connection immediately.
- Most sites do not log these.
- Need root privileges to build SYN packets.

UDP Scans

- UDP is simpler, but the scanning is more difficult
- Open ports do not have to send an ACK.
- Closed ports are not *required* to send an error packet.
  - Most hosts send an ICMP_PORT_UNREACH error when you send a packet to a closed UDP port.
  - Can find out if a port is NOT open.

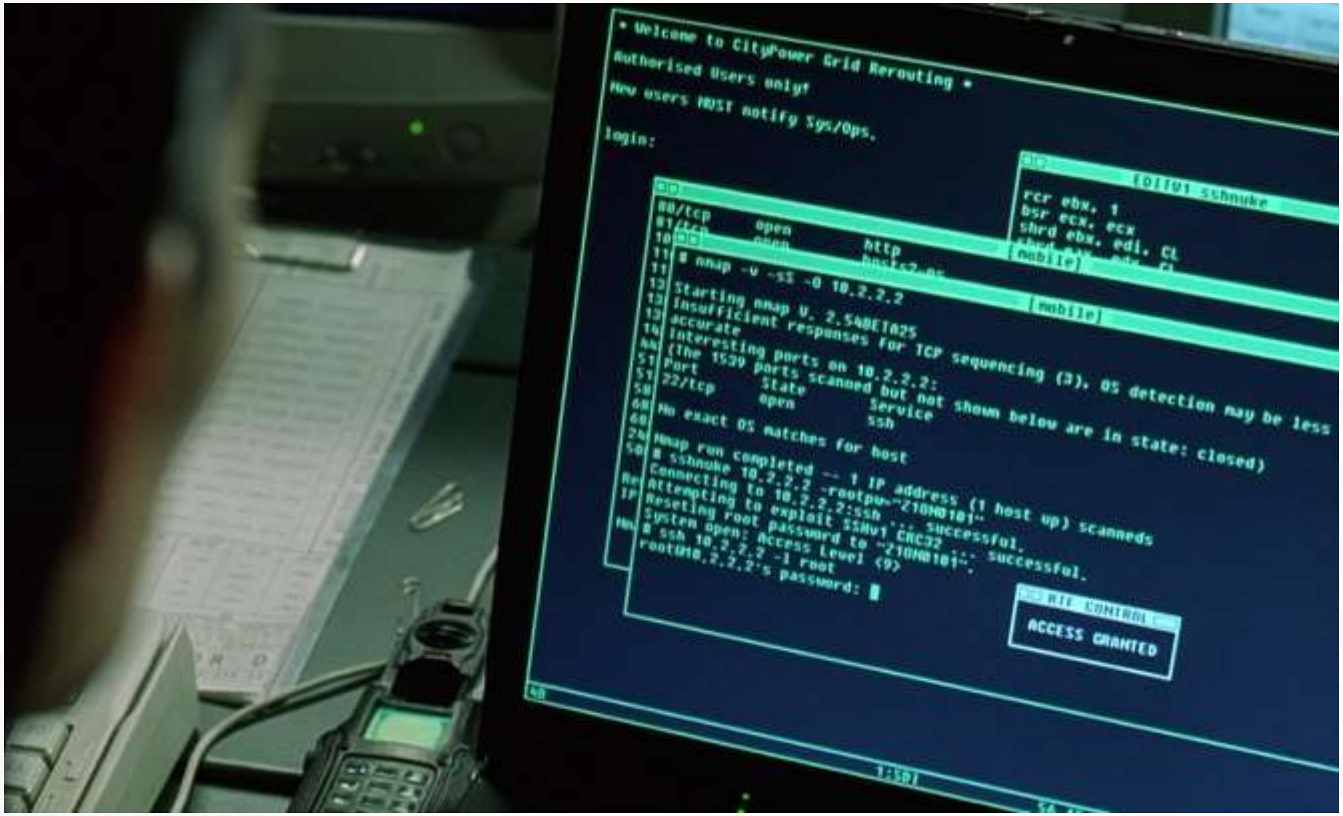
UDP Scans

- Neither UDP packets, nor the ICMP errors are guaranteed to arrive.
- Slow: the ICMP error message rate is limited.
- Need to be root for access to raw ICMP socket.
- Non-root users cannot read port unreachable errors directly.

UDP Scans

- But users can learn it indirectly.
- For example, a second write() call to a closed port will usually fail.
- recvfrom() on non-blocking UDP sockets usually return EAGAIN (try again), if the ICMP error hasn’t been received.
- It will return ECONNREFUSED (connection refuse), if ICMP error has been received.

NMAP



```
* Welcome to CityPower Grid Rerouting *
Authorised Users only!
New users MUST notify Sys/Ops.

login:
```

```
EDITV1 sshnuke
rcr ebx, 1
bsr ecx, ecx
shrd ebx, edi, CL
```

```
80/tcp      open
81/tcp      open         http
```

```
[mobile]
# nmap -v -sS -O 10.2.2.2
Starting nmap V. 2.54BETA25
Insufficient responses for TCP sequencing (3), OS detection may be less
accurate
Interesting ports on 10.2.2.2:
(The 1539 ports scanned but not shown below are in state: closed)
Port        State       Service
22/tcp      open        ssh

No exact OS matches for host

Nmap run completed -- 1 IP address (1 host up) scanneds
# sshnuke 10.2.2.2 -rootpw="Z10N0101"
Connecting to 10.2.2.2:ssh ... successful.
Attempting to exploit SSHv1 CRC32 ... successful.
Reseting root password to "Z10N0101".
System open: Access Level <9>
# ssh 10.2.2.2 -l root
root@10.2.2.2's password:
```

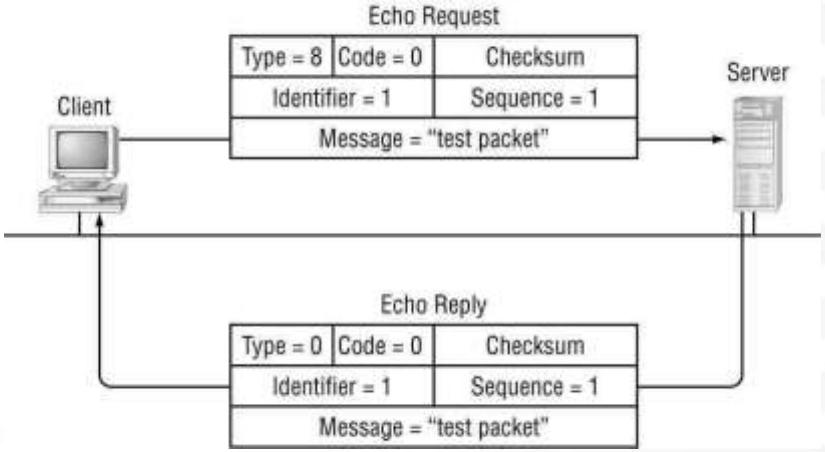
Matrix

Objectives of NMAP use

- Discovery of running services
- Discovery of versions of OS and services
- To Determin what firewall rules are applied
- Discovery information about vendor of the computer equipment

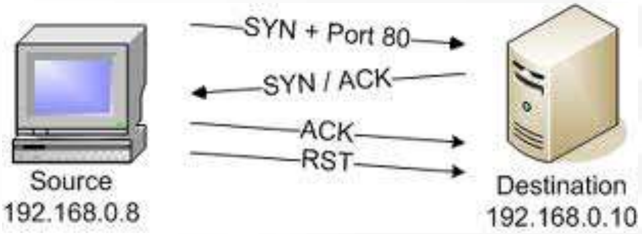
NMAP as a ping

nmap –sP –v 192.168.5.25



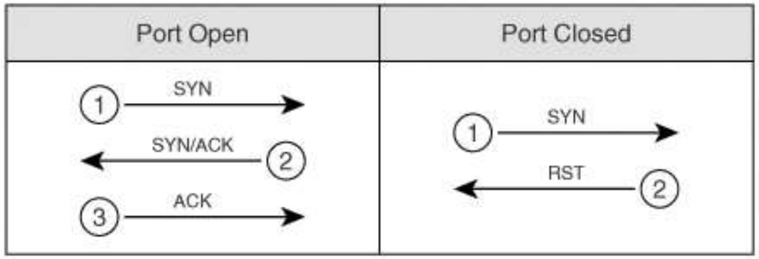
NMAP: tcp scan

- TCP connect/full scan – full TCP connection is established and interrupted by sending RST-packet
- NMAP key: -sT
- Usage: if NMAP cannot generate raw packets.



NMAP: stealth scan

- Stealth scan/half-open scan – scanning by sending packets with SYN flag.
- Allows to determine what ports are open, closed or filtered
- Good speed.
- NMAP key: -sS



Port Open

1 SYN →

← SYN/ACK 2

3 ACK →

Port Closed

1 SYN →

← RST 2

Exotic types of scans

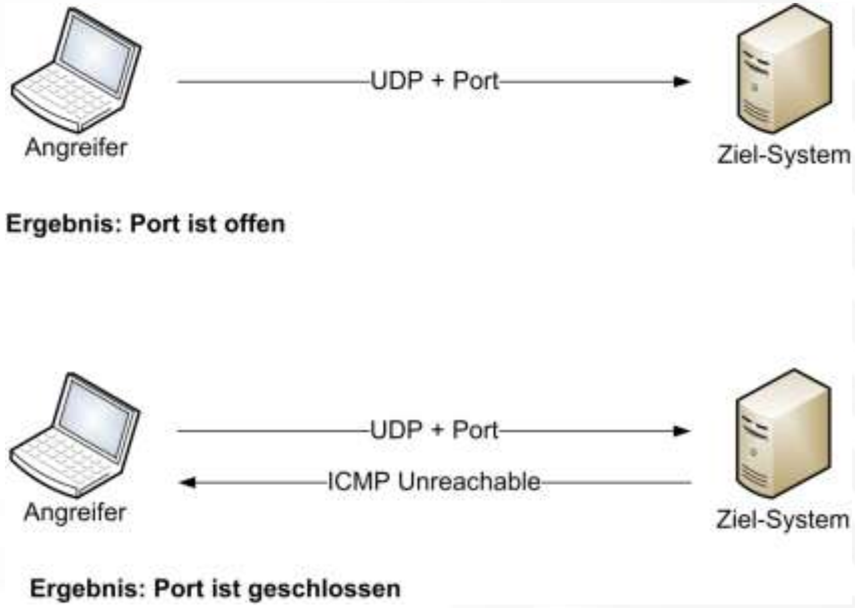
- Xmas Scan
- FIN Scan
- NULL Scan

It doesn't work with Windows!

Automation of search for vulnerabilities: vulnerability scanners

NMAP: UDP-scan

- NMAP key: -sU



Ergebnis: Port ist offen

Ergebnis: Port ist geschlossen

Examples of NMAP usage

nmap 192.168.1.1	Default scanning of the host
nmap -v server1.cyberciti.biz	Scanning in verbose mode
nmap -F 192.168.1.1	Quick scan
nmap --reason 192.168.1.1	Show state of ports
nmap -p U:53 192.168.1.1	Scan UDP-port 53 only
nmap -v -O --osscan-guess 192.168.1.1	Determine what version of OS is used
nmap -sV 192.168.1.1	Determine versions of services

Vulnerability scanner

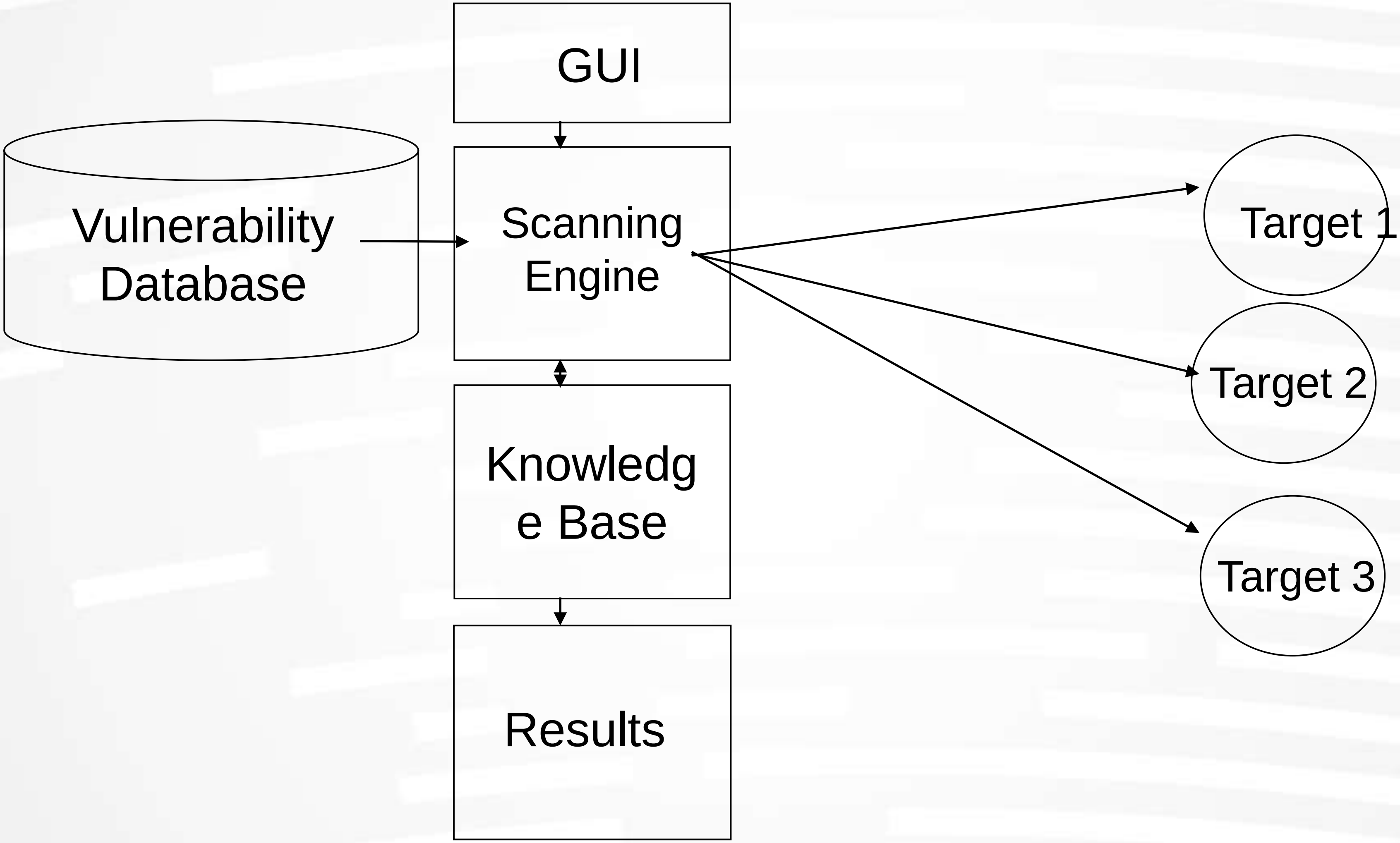
69.72.169.241 →

→

IP-addresses to scan

Report with discovered vulnerabilities

How vulnerability scanners work



Vulnerability scanners

- Similar to virus scanning software:
  - Contain a database of vulnerability signatures that the tool searches for on a target system
  - Cannot find vulnerabilities not in the database
    - New vulnerabilities are discovered often
    - Vulnerability database must be updated regularly

Typical Vulnerabilities Checked

- Network vulnerabilities
- Host-based (OS) vulnerabilities
  - Misconfigured file permissions
  - Open services
  - Missing patches
  - Vulnerabilities in commonly exploited applications (e.g. Web, DNS, and mail servers)

Vulnerability Scanners - Benefits

- Very good at checking for hundreds (or thousands) of potential problems quickly
  - Automated
  - Regularly
- May catch mistakes/oversights by the system or network administrator
- Defense in depth

Vulnerability Scanners - Drawbacks

- Report “potential” vulnerabilities
- Only as good as the vulnerability database
- Can cause complacency
- Cannot match the skill of a talented attacker
- Can cause self-inflicted wounds

Popular vulnerability scanners

- Nessus
- OpenVAS
- Qualys