TN Department of Finance & Administration | Strategic Technology Solutions

Apache Log4j Vulnerability and the Log4shell exploit(s)

The Issue

There is a vulnerability (CVE-2021-44228) in the Apache Log4j logging library that allows for remote code execution (RCE), ransomware, crypto miners, and data exfiltration. Log4shell is the name given to the exploits broadly. The Log4shell attacks can be delivered through a variety of protocols including IMAP, DNS, SMTP, HTTP, and LDAP. Lateral movement within a compromised environment is possible using RCE

What to do First

The Cybersecurity and Infrastructure Security Agency (CISA) produced a valuable article with guidance on the vulnerability here: https://www.cisa.gov/uscert/apache-Log4j-vulnerability-guidance

From the callout "Immediate Actions to Protect Against Log4j Exploitation" at the beginning of the above article:

Immediate Actions to Protect Against Log4j Exploitation

- Discover all internet-facing assets that allow data inputs and use Log4j Java library anywhere in the stack.
- Discover all assets that use the Log4j library.
- Update or isolate affected assets. Assume compromise, identify common post-exploit sources and activity, and hunt for signs of malicious activity.
- Monitor for odd traffic patterns (e.g., JNDI LDAP/RMI outbound traffic, DMZ systems initiating outbound connections).

Patching

See: https://logging.apache.org/log4j/2.x/security.html

The version of Log4j that needs to be attained is different for different versions of Java.

Patch to Log4j 2.17.0 (Java 8), 2.12.3 (Java 7) and 2.3.1 (Java 6).

Log4j version 1.x is not vulnerable, but it is end of life should be updated or removed.

Apache Log4j Vulnerability and the Log4shell exploit(s)

Mitigation

If patching is not an option, in any release other than 2.16.0, you may remove the JndiLookup class from the classpath (environment variable). The JndiLookup.class contains the vulnerable code. It is found in log4j-core-*.jar files and should be deleted.

Linux: (command is one line)

```
zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
```

Windows:

- Copy log4j-core-*.jar to a temp folder and keep a secondary backup in another location.
- Right click on the file, choose properties then uncheck Read-only check box.
- Add the extension .zip to log4j-core-*.jar by renaming it. This will allow it to be opened with Windows File Explorer, 7zip or WinRAR.
- Open (do not extract) log4j-core-*.jar.zip with a zip utility, locate org/apache/logging/log4j/core/lookup/JndiLookup.class and delete it.
- Close the zip utility and reopen again to make sure the JndiLookup class is removed.
- Remove the .zip extension from the log4j-core-*.jar.zip

Old, deprecated guidance *(do not use)*.

Early mitigation advice included setting the system property "Log4j2.formatMsgNoLookups" or the environment variable "LOG4J_FORMAT_MSG_NO_LOOKUPS" to "true" as a mitigation measure. This is not effective and should be dismissed and if previously followed, reversed.

Affected Software

The Log4j library is a component in a large number of commercial and open-source products.

A comprehensive list of affected software can be found in this CISA GitHub repository: https://github.com/cisagov/Log4j-affected-db

Apache Log4j Vulnerability and the Log4shell exploit(s)

Detection of Vulnerable Log4j

CISA links to CERT/CC's CVE-2021-44338 scanner here: https://github.com/CERTCC/CVE-2021-44228_scanner

It provides bash, PowerShell and Python3 detection scripts. These require direct, administrative access to the target systems, however. Tenable Security's Nessus scanner can detect Log4j on a large scale.

Detection of Compromised Systems

This repository contains a wide variety of techniques to detect compromised systems:

https://gist.github.com/Neo23x0/e4c8b03ff8cdf1fa63b7d15db6e3860b

Be sure to log DNS queries and look for outbound LDAP attempts.

Workstations?

Workstations behind firewalls are not yet a concern. However, considering the large number of affected applications it is possible desktop exploits will be developed. It is likely the attack would involve phishing. By directing a user to a malicious site, an attacker would then have an inbound connection to the user's computer to attempt attacks against applications that use Log4j.

For questions, please reach out to the CISO's office

through this CyberSafeTN contact form.

Appendix A

CISA Tool Tips

There are three different scripts in the CISA github repository: https://github.com/CERTCC/CVE-2021-44228_scanner

Use the PowerShell script for Windows systems.

Copy checkjndi.ps1 to your Windows system.

Be sure to run PowerShell as Administrator.

The command is:

.\checkjndi.ps1 c:\ -Force

Sample output showing a "hit": (Note the scanner found the file in the Recycle Bin in a .jar in a .zip file!)

WARNING: C:\$Recycle.Bin\S-1-5-21-2149558826-3329038498-27948981-29912\$RH3WYH0.zip contains log4j-core-2.8.2.jar

Repeat for other drives.

See instructions "Removal of JndiLookup.class" below for removal.

Use the Bash or Python3 scripts on Linux.

Python version 3 is required. The Bash script is just as thorough and does not require any other software.

Both are limited to one filesystem at a time.

Copy the Bash script to your Linux system and make it executable:

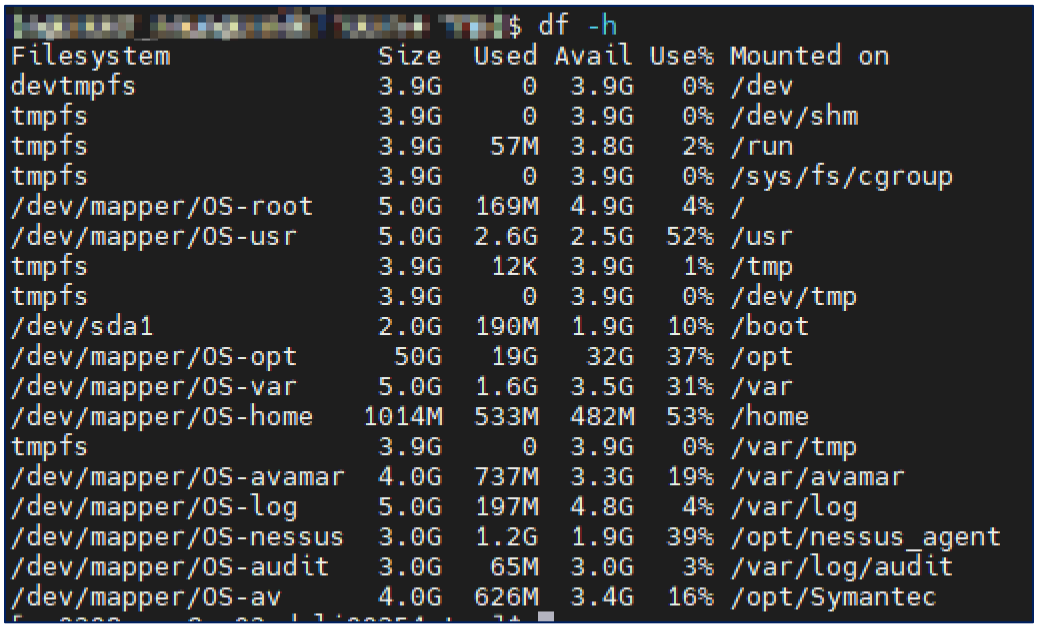
chmod 777 checkjndi.sh

The Bash command is:

sudo ./checkjndi.sh /

Apache Log4j Vulnerability and the Log4shell exploit(s)

Repeat for all filesystems such as /var and /opt. Use “df -h” (disk free, -human friendly) to find all the filesystems for a given Linux system. The “Mounted on” column contains the filesystem names.



```
[redacted]$ df -h
Filesystem               Size  Used Avail Use% Mounted on
devtmpfs                 3.9G     0  3.9G   0% /dev
tmpfs                    3.9G     0  3.9G   0% /dev/shm
tmpfs                    3.9G   57M  3.8G   2% /run
tmpfs                    3.9G     0  3.9G   0% /sys/fs/cgroup
/dev/mapper/OS-root      5.0G  169M  4.9G   4% /
/dev/mapper/OS-usr       5.0G  2.6G  2.5G  52% /usr
tmpfs                    3.9G   12K  3.9G   1% /tmp
tmpfs                    3.9G     0  3.9G   0% /dev/tmp
/dev/sda1                2.0G  190M  1.9G  10% /boot
/dev/mapper/OS-opt        50G   19G   32G  37% /opt
/dev/mapper/OS-var       5.0G  1.6G  3.5G  31% /var
/dev/mapper/OS-home     1014M  533M  482M  53% /home
tmpfs                    3.9G     0  3.9G   0% /var/tmp
/dev/mapper/OS-avamar    4.0G  737M  3.3G  19% /var/avamar
/dev/mapper/OS-log       5.0G  197M  4.8G   4% /var/log
/dev/mapper/OS-nessus    3.0G  1.2G  1.9G  39% /opt/nessus_agent
/dev/mapper/OS-audit     3.0G   65M  3.0G   3% /var/log/audit
/dev/mapper/OS-av        4.0G  626M  3.4G  16% /opt/Symantec
```

<u>Removal of JndiLookup.class</u>

If the scans locate any libraries containing the JndiLookup.class and the Log4j version is not the latest patched version use these commands to remove the file from the library:

Linux: (command is one line)

zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class

Windows:

Copy log4j-core-*.jar to a temp folder and keep a secondary backup in another location.
Right click on the file, choose properties then uncheck Read-only check box.
Add the extension .zip to log4j-core-*.jar by renaming it. This will allow it to be opened with Windows File Explorer, 7zip or WinRAR.
Open (do not extract) log4j-core-*.jar.zip with a zip utility, locate org/apache/logging/log4j/core/lookup/JndiLookup.class and delete it.
Close the zip utility and reopen again to make sure the JndiLookup class is removed.
Remove the .zip extension from the log4j-core-*.jar.zip