The information concerning the Log4j and Log4j2 vulnerabilities has recently expanded since the beginning of the holidays, and it seems like every day that passes another vulnerability is discovered concerning the log4j and log4j2 software from Apache.org.

VSI is currently working to compile a list of applications provided by VSI that may be affected by this new information and will look to make all relevant information available as soon as possible.

Here is what we at VSI have recently learned:

The CVE List:

* CVE-2021-45046
* CVE-2021-44228
* CVE-2021-4104
* CVE-2021-45105
* CVE-2021-44832

CVE-2021-4104:

JMSAppender in Log4j 1.2 is vulnerable to deserialization of untrusted data when the attacker has write access to the Log4j configuration. The attacker can provide TopicBindingName and TopicConnectionFactoryBindingName configurations causing JMSAppender to perform JNDI requests that result in remote code execution in a similar fashion to CVE-2021-44228. Note this issue only affects Log4j 1.2 when specifically configured to use JMSAppender, which is not the default. **Apache Log4j 1.2 reached end of life in August 2015**. Users should upgrade to Log4j 2 as it addresses numerous other issues from the previous versions.

Solution:

- In Log4j 2.12.2 (for Java 7) and 2.16.0 (for Java 8 or later) the message lookups feature has been completely removed. In addition, JNDI is disabled by default and other default configuration settings are modified to mitigate CVE-2021-44228 and CVE-2021-45046.

- For Log4j 1, remove the JMSAppender class or do not configure it. Log4j 1 is not supported and likely contains unfixed bugs and vulnerabilities (such as CVE-2019-17571).

- For applications, services, and systems that use Log4j, consult the appropriate vendor or provider.

Workarounds:

- Remove the JndiLookup class from the classpath, for example:

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

- As analysis has progressed, certain mitigations have been found to be less effective or incomplete. See "Older (discredited) mitigation measures" on the Apache Log4j Security Vulnerabilities page.

- SLF4J also recommends write-protecting Log4j configuration files.

CVE-2021-44228:

The CVE that started it all…
Apache Log4j2 2.0-beta9 through 2.15.0 (excluding security releases 2.12.2, 2.12.3, and 2.3.1) 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. From version 2.16.0 (along with 2.12.2, 2.12.3, and 2.3.1), this functionality has been completely removed. Note that this vulnerability is specific to log4j-core and does not affect log4net, log4cxx, or other Apache Logging Services projects.

Log4j 1.x mitigation

- Log4j 1.x does not have Lookups so the risk is lower. Applications using Log4j 1.x are only vulnerable to this attack when they use JNDI in their configuration. A separate CVE (CVE-2021-4104) has been filed for this vulnerability. To mitigate: Audit your logging configuration to ensure it has no JMSAppender configured. Log4j 1.x configurations without JMSAppender are not impacted by this vulnerability.

Log4j 2.x mitigation:

Implement one of the following mitigation techniques:

- Upgrade to Log4j 2.3.1 (for Java 6), 2.12.3 (for Java 7), or 2.17.0 (for Java 8 and later).

- Otherwise, in any release other than 2.16.0, you may remove the JndiLookup class from the classpath:

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

- Note that only the log4j-core JAR file is impacted by this vulnerability. Applications using only the log4j-api JAR file without the log4j-core JAR file are not impacted by this vulnerability.

- Also note that Apache Log4j is the only Logging Services subproject affected by this vulnerability. Other projects like Log4net and Log4cxx are not impacted by this.

CVE-2021-44832:
Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack where an attacker with permission to modify the logging configuration file can construct a malicious configuration using a JDBC Appender with a data source referencing a JNDI URI which can execute remote code. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2.

Solution:

- **Log4j 1.x is not impacted by this specific vulnerability.**

- Upgrade minimally to Log4j V2.3.2 (for Java 6),
  V2.12.4 (for Java 7),
  -or-
  V2.17.1 (for Java 8 and later)

CVE-2021-45046:
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 allow 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 an information leak and remote code execution in some environments and local code execution in all environments. Log4j 2.16.0 (Java 8) and 2.12.2 (Java 7) fix this issue by removing support for message lookup patterns and disabling JNDI functionality by default.

Solution:

- **Log4j 1.x is not impacted by this specific vulnerability.**

- In Log4j 2.12.2 (for Java 7) and 2.16.0 (for Java 8 or later) the message lookups feature has been completely removed. In addition, JNDI is disabled by default and other default configuration settings are modified to mitigate CVE-2021-44228 and CVE-2021-45046.

- For Log4j 1, remove the JMSAppender class or do not configure it. Log4j 1 is not supported and likely contains unfixed bugs and vulnerabilities (such as CVE-2019-17571).

- For applications, services, and systems that use Log4j, consult the appropriate vendor or provider.

Workarounds:

- Remove the JndiLookup class from the classpath, for example:

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

- As analysis has progressed, certain mitigations have been found to be less effective or incomplete. See "Older (discredited) mitigation measures" on the Apache Log4j Security Vulnerabilities page.

- SLF4J also recommends write-protecting Log4j configuration files.

CVE-2021-45105:
Apache Log4j2 versions 2.0-alpha1 through 2.16.0 (excluding 2.12.3 and 2.3.1) did not protect from uncontrolled recursion from self-referential lookups. This allows an attacker with control over Thread Context Map data to cause a denial of service when a crafted string is interpreted. This issue was fixed in Log4j 2.17.0, 2.12.3, and 2.3.1.

Solution:

- In Log4j 2.12.2 (for Java 7) and 2.16.0 (for Java 8 or later) the message lookups feature has been completely removed. In addition, JNDI is disabled by default and other default configuration settings are modified to mitigate CVE-2021-44228 and CVE-2021-45046.

- For Log4j 1, remove the JMSAppender class or do not configure it. Log4j 1 is not supported and likely contains unfixed bugs and vulnerabilities (such as CVE-2019-17571).

- For applications, services, and systems that use Log4j, consult the appropriate vendor or provider.

Workarounds:

- Remove the JndiLookup class from the classpath, for example:

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

- As analysis has progressed, certain mitigations have been found to be less effective or incomplete. See "Older (discredited) mitigation measures" on the Apache Log4j Security Vulnerabilities page.

- SLF4J also recommends write-protecting Log4j configuration files.

That is the current Narrative and Optics on these CVE vulnerabilities for Log4j and Log4j2.

This document will be updated by VSI as more information comes to light concerning these vulnerabilities.

Ref:
https://www.mitre.org/ CVE Search List.
https://logging.apache.org Log4j Developers and Documentation site.
https://techrepublic.com
“Log4j vulnerability: Why your hot take on it is wrong Author: -Matt Asay 2021”,
https://www.kb.cert.org/vuls/id/930724 - Log4j and Log4j2 vulnerabilities.