Are Globalscape products affected by the Log4j v2 security vulnerabilities?

THE INFORMATION IN THIS ARTICLE APPLIES TO:

- All products, all versions

QUESTION

Are Globalscape products affected by the Log4j v2 security vulnerability?

ANSWER

At HelpSystems we are aware of CVE-2021-44228 in response to the open-source Apache "Log4j2" utility. HelpSystems is actively monitoring this issue, investigating the potential impact on our products, and assembling the appropriate mitigations. At this time, Globalscape is not impacted by this vulnerability. If you need additional details or assistance, please contact support.

- Globalscape EFT doesn't use any version of Java or log4j.
- Globalscape DMZ Gateway is the only Java-based product Globalscape has that uses **v1.X** of log4j, which is not affected by these Log4j v2 vulnerabilities. (Log4j was updated to v2.17 in DMZ Gateway v3.5.0.35.) The DMZ Gateway is shipped with an embedded Java Virtual Machine. We have a documented process for the customer to update the JVM to a newer version, as needed. Newer versions of Java have blocked the ability for remote code execution via a system property, which is disabled by default.
- Additionally, the EFT Arcus infrastructure has been analyzed by the lead Arcus engineer and found not to be susceptible to any Log4j exploits.

UPDATE: Apache Log4j2 versions 2.0-alpha1 through 2.16.0 (excluding 2.12.3) did not protect from uncontrolled recursion from self-referential lookups, which would allow an attacker to control 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 and 2.12.3.

- **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 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 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.
- **CVE-2021-45105** Apache Log4j2 versions 2.0-alpha1 through 2.16.0 (excluding 2.12.3) 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 and 2.12.3.
- **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.
- **CVE-2021-44228** 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 it can be mitigated in prior releases (<2.10) by removing the JndiLookup class from the classpath (example: zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class).
    - DMZ Gateway uses Log4j 1.2.16 which is a version of the library not reported as susceptible to the exploit in the CVE. Some websites have suggested the version DMZ uses could be susceptible as well. Regardless of whether this version is vulnerable, DMZ does not employ the JMS appender class and as such is not vulnerable to this exploit. Also, versions of Java deemed most susceptible to this attack are those older than 8u121; the version of Java deployed with the latest DMZ installer is 8u202.
- **CVE-2020-9488** Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender.

  - DMZ Gateway does not use the SMTP appender.
- **CVE-2019-17571** Included in Log4j 1.2 is a SocketServer class that is vulnerable to deserialization of untrusted data which can be exploited to remotely execute arbitrary code when combined with a deserialization gadget when listening to untrusted network traffic for log data. This affects Log4j versions 1.2 up to 1.2.17.
  - The Log4j SocketServer class is not used by DMZ code.

GlobalSCAPE Knowledge Base
https://kb.globalscape.com/Knowledgebase/11557/Are-Globalscape-products-aff...