NETTITUDE
AN LRQA COMPANY

Threat Intelligence Briefing: Log4Shell

Intelligence Summary (15 Dec Update)

Date: 15 DECEMBER 2021
Version: 1.0

Confidential Security Document [Nettitude + Clients]

Contents

1 OVERVIEW AND CONTEXT 2

1.1 LATEST INFORMATION: 2

1.2 WHAT DOES THIS VULNERABILITY LOOK LIKE? 2

2 MITIGATION ISSUES 4

2.1 INEFFECTIVE PREVIOUS MITIGATIONS: 4

2.2 SUMMARY 5

3 WIDER INTELLIGENCE SURROUNDING LOG4SHELL 5

3.1 NATION-STATE ATTACKS 5

3.2 CRIMINAL EXPLOITATION 6

3.3 HOW IS THIS LIKELY TO DEVELOP? 6

4 REVISION HISTORY 7

1 Overview and Context

1.1 Latest Information:

According to a new Apache security bulletin the initially suggested mitigation measures for CVE-2021-44228 (Log4j version 2.15.0) do not completely address the Log4Shell vulnerability in certain custom configurations. Further, the mitigation measures previously reported, such as setting the log4j2.formatMsgNoLookups variable to 'true', are not considered fully effective.

A new vulnerability CVE-2021-45046 has been released with further details and Apache have released another patch (version 2.16.0) to address this. This has been given a CVSS base score of 3.7.

Nettitude's threat intelligence teams have been assessing the impact of these disclosures and have issued this briefing as updated advice.

Nettitude encourage you to:

1 Invoke your incident crisis management teams, as required, to immediately review your organisation's exposure to this event;

2 Understand your public exposure to the Log4j vulnerabilities and take immediate steps to mitigate, patch and/or remove these;

3 Review all your internal assets for the use of Log4j (understanding that it is not always straightforward to detect this through automated scanning tools) and put in place the relevant mitigation;

4 Review the impact to your organisation's objectives, risks and clients;

5 Review your third party suppliers and ensure they are also taking the necessary and relevant steps to mitigate this issue.

1.2 What does this Vulnerability look like?

The key characteristics of this vulnerability are:

- **Ease of exploitation:** What makes CVE-2021-44228 especially dangerous is how easy it is to exploit. Someone who is an inexperienced hacker can successfully execute this attack simply by sending a single request that forces a log entry to be written. This will then enable the attacker to upload their own code into the application.
- **The prolific use of Apache Web Servers and the Log4j library**: This library is used in millions of websites, applications, and platforms, including a wide range of Apache products. It is also used in gaming platforms such as Minecraft.

- **Log4j can be hard to find:** Simply running scanning tools is not always effective in detecting where in networks and products Log4j is used. As it is open source it is very easily incorporated into products. It has been downloaded from the Apache website over 84 million times in the last 4 months alone.
- **Vendors and Developers are struggling to remediate quickly:** There is a key reliance of vendors and product owners to respond quickly with remediation, fixes and patching. Many large product vendors are struggling to respond within the timescales required meaning that clients' systems remain vulnerable long after the exploits are widely known and used.

Overnight (14-15 Dec) more intelligence emerged around how this vulnerability affects systems:

- **Previous mitigation measures are now thought to be ineffective:** In certain custom configurations, mitigation measures previously reported, such as setting the log4j2.formatMsgNoLookups variable to 'true', are not considered fully effective. A new vulnerability CVE-2021-45046[1] has been released with further details and Apache have released another patch (version 2.16.0) to address this. This has been given a CVSS base score of 3.7.
- **Proof of concept code is widespread, and exploitation is now ramping up**: The issue was discovered on 8 December and active scanning for this vulnerability started almost immediately. Vulnerability scanning tools have been updated to detect this issue and security products and tools themselves are being patched (where fixes are available).
- **Ransomware Initial Access Brokers (IABs) gaining access to systems and then selling this on:** A flurry of activity has now begun as those involved in the Ransomware-As-A-Service (RaaS) supply chain, such as Initial Access brokers, will look to establish persistence before then selling the access on to those skilled in the later stages of kill chain activity.
- **Sophisticated Nation-State threat actors are now conducting exploitation:** Planning and exploitation phases are likely due to be finished and they will now be conducting offensive actions using Log4Shell.
- **Use of this vulnerability in 'wormable' malware:** Of additional concern is that there is a high 'worm' potential from Log4Shell, and automated exploitation is likely.

[1] https://logging.apache.org/log4j/2.x/security.html

2 Mitigation issues

The following details have been observed since the distribution of the Nettitude Log4Shell Threat Update Report (14 Dec 2021).

2.1 Ineffective Previous Mitigations:

According to a new Apache Log4j security bulletin[2], version 2.15.0, the initially suggested mitigation update does not completely address the Log4Shell vulnerability in certain custom configurations.

It was discovered that version 2.15.0 would still be vulnerable when the configuration has a pattern layout containing a Context Lookup (for example, $${ctx:loginId}), or a Thread Context Map pattern %X, %mdc, or %MDC. In these cases, when the attacker manages to control the Thread Context values, JNDI lookup injections may be possible, resulting in JNDI connections.

Version 2.15.0 limited JNDI connections to 'localhost" but this possibility could result in a denial of service (DoS) or worse.

Therefore, an updated version (2.16.0) has been made available to address the original issues associated with CVE-2021–45046 along with more effective mitigation measures for versions to 2.x versions:

- Users of Java 8 (or later) should upgrade to release 2.16.0.
- Users requiring Java 7 should upgrade to release 2.12.2 when it becomes available (note currently available but expected soon).

Otherwise, advice is to remove the JndiLookup class from the classpath: zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class

The mitigation measures previously reported, such as setting the log4j2.formatMsgNoLookups variable to 'true', is not considered fully effective.

The advisory points out:

> "The reason these measures are insufficient is that, in addition to the Thread Context attack vector mentioned above, there are still code paths in Log4j where message lookups could occur: known examples are applications that use Logger.printf("%s", userInput), or applications that use a custom message factory, where the resulting messages do not implement StringBuilderFormattable. There may be other attack vectors."

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

2.2 Summary

In summary:

1. Log4j 2.x mitigation: Implement one of the mitigation techniques below.
    - Java 8 (or later) users should upgrade to release 2.16.0.
    - Users requiring Java 7 should upgrade to release 2.12.2 when it becomes available (work in progress, expected to be available soon).
    - Otherwise, remove the JndiLookup class from the classpath: zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
2. If none of these options are viable, you should disable/remove Log4j from your environment, or manage through additional security products (**Warning:** This latter option may be hard to implement and will not provide a guaranteed fix).

The reliance on vendors to update their products, especially where Log4j has been integrated into their overall products, is a continuing challenge. Even after a patch is released the update process for many organisations takes time and is not always straightforward. This is one of the key reasons why this issue will continue to pervade for many weeks/months yet to come.

3 Wider Intelligence Surrounding Log4Shell

There has now been confirmed exploitation of CVE-2021-44228 (Log4Shell) by multiple State-sponsored groups.

3.1 Nation-State Attacks

On 14 December 2021, Microsoft's Threat Intelligence Centre (MSTIC) published updated information regarding threat activity groups observed targeting the Log4Shell vulnerability in the Java-based logging utility Log4J. MSTIC confirmed that the CVE-2021-44228 vulnerability is being used by multiple state-sponsored activity groups originating from China, Iran, North Korea, and Turkey. This activity reportedly ranges from experimentation during development, integration of the vulnerability to in-the-wild payload deployment, and exploitation against targets in order to gather information against state collection objectives.

For example, MSTIC has observed PHOSPHORUS (also known as Charming Kitten), an Iranian threat actor that has been seen deploying ransomware, acquiring and making modifications of the Log4j exploit. MSTIC also observed the suspected Chinese state-sponsored group HAFNIUM utilising the vulnerability to attack virtualisation infrastructure to extend their typical targeting.

In these attacks, HAFNIUM-associated systems were observed using a DNS service typically associated with testing activity to fingerprint systems. In early 2021, HAFNIUM was the first group observed exploiting the major ProxyLogon vulnerability in Microsoft Exchange.

3.2 Criminal Exploitation

In addition to state-sponsored activity, MSTIC observed financially motivated Initial Access brokers (IAB) begin to use the vulnerability to gain initial access to target networks on both Windows and Linux systems. These access brokers then sell access to these networks to ransomware-as-a-service affiliates. Insikt Group expects continued adoption of Log4Shell usage by state-sponsored and financially motivated actors over the coming weeks.

Some companies have offered their services to monitor and publish tagged IP addresses that appear to be scanning for the Log4j vulnerability. Other organisations, such as Cloudflare, have openly stated that the damage potential is so great that they are introducing proactive mitigation measures, even for non-paying organisations.

Microsoft has stated that the vast majority of activity observed is mass scanning to fingerprint servers, though they have also witnessed ‘exploitation and post-exploitation’ activity.

Researchers on Twitter have also reported malicious exploitation. Blackberry researcher Greg Linares reports witnessing attacks specifically targeting SIEM installations, specifically Splunk. Splunk has stated Data Fabric Search (DFS) and Splunk Analytics for Hadoop (Hunk) product features are affected and provided guidance on how to remove Log4j from Splunk.

3.3 How Is This Likely to Develop?

Dark web threat actors on XSS and Exploit forums have discussed the Log4Shell exploit and shared links to Proof-of-Concept code. Some actors mentioned that GitHub appeared to be taking down these POC repositories, but many have been forked and several can be openly found and are repeatedly linked. Some of the traffic may be lower given the extreme ease of exploiting Log4Shell.

The Log4Shell exploit is useful both for initial access and lateral movement, as both externally-facing and internal services can use the Log4j logging library. As a result, it is no surprise that criminals have already begun exploiting this vulnerability, and active exploitation continues.

Blackberry researcher Greg Linares also predicts the likelihood of a ‘wormable’ version within the next few days, with some groups reportedly actively working on such versions. In addition, researchers believe that because the combination of web shells and Cobalt Strike beacons are typically the first tools deployed by nation-state groups and ransomware gangs in attacks, we will shortly begin to see exploitation from these types of groups.

4 Revision History

Version	Issue Date	Issued by	Comments
1.0	15 Dec 2021	Nettitude Threat Intelligence	Initial Release

NETTITUDE
AN LRQA COMPANY

UK Head Office
Jephson Court, Tancred
Close, Leamington Spa,
CV31 3RZ

Americas
50 Broad Street,
Suite 403, New York,
NY 10004

Asia Pacific
1 Fusionopolis Place,
#09-01, Singapore,
138522

Europe
Leof. Siggrou 348
Kallithea, Athens, 176 74
+30 210 300 4935

Follow Us

solutions@nettitude.com
www.nettitude.com