Advisory – Log4Shell vulnerability

v1 12-Dec-2021 11:30 IST

Introduction

A vulnerability in the widely used Log4j utility was disclosed in public domain on 5-Dec-2021. The vulnerability – now called Log4Shell or LogJam - allows for the attacker – through a specifically crafted input string - to take full control of the targeted system remotely. Numerous active exploits are noticed for this vulnerability. The vulnerability is provided CVE id CVE-2021-44228, and is given the highest vulnerability rating of 10.

Impact

Log4j utility is so widely used that organisations are recommended to start with the assumption that it could be deployed on all IT systems. Also, the ease of exploit is simple and the number of threat actors using the exploit is estimated to be numerous.

So, this seems to be a perfect storm of an extremely critical, easy to exploit vulnerability in a software widely used.

Recommendations

1. If running on Log4j 2.10 or later, upgrade to Log4j 2.15.0.
2. Where upgrade is not possible:
    a. For Log4j 2.10 or later, modify the system property log4j2.formatMsgNoLookups or the environment variable LOG4J_FORMAT_MSG_NO_LOOKUPS to true.
    b. For Log4j 2.0-beta9 to 2.10.0, remove JndiLookup class from the classpath.
    zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class
3. Monitor for exploit patterns like “jndi:ldap” in input strings / logs.
4. Use security controls like WAF and firewalls to block malicious traffic. Consider disabling outbound LDAP and RMI traffic.
5. Reach out to all your OEMs and follow their advisory for patching the vulnerability on OEM systems. Refer to the latest advisories from OEM.

It is important that these changes are deployed following proper Change Management processes for Emergency category changes. It is also important that all components deployed are obtained from valid sources with

Additional recommendations:

1. An open-source Python programme is now made available to detect for exploits. See https://github.com/Neo23x0/log4shell-detector.
2. Tools like Nessus now have a plugin to detect the presence of this vulnerability, as well as detect any IOCs (Indicators of Compromise).
3. If you have a Software Composition Analysis tool, use that to identify software that would have this vulnerable version of Log4j.
4. Follow expert Kevin Beaumont https://twitter.com/GossiTheDog/ for regular updates.

References

1. https://logging.apache.org/log4j/2.x/security.html
2. https://www.rapid7.com/blog/post/2021/12/10/widespread-exploitation-of-critical-remote-code-execution-in-apache-log4j/
3. https://blog.cloudflare.com/inside-the-log4j2-vulnerability-cve-2021-44228/
4. https://blog.cloudflare.com/cve-2021-44228-log4j-rce-0-day-mitigation/
5. https://unit42.paloaltonetworks.com/apache-log4j-vulnerability-cve-2021-44228/