Log4j Cloud Remediation Checklist

To help you navigate the enormous amount of information that has been published on how to mitigate the Apache Log4j vulnerability, we have compiled this easily digestible checklist with practical remediation steps:

01

Find all vulnerable Log4j2 instances:

Run a scan of all your software and systems in the cloud to see if the **vulnerable Log4j versions** are present. You can either use a **vulnerable hash list** (however this will not detect all bundled jars) or a cloud security solution that has visibility at the workload level.

Note: Make sure that you not only scan for direct installations of the Log4j packages, but also jars that bundle Log4j within them.

02

Ensure 100% coverage:

Make sure that you scan your entire cloud estate. Note that if you are using an agent-based cloud security solution, on average you will only cover 50% of your cloud assets, due to cumbersome and incomplete agent deployment. An agentless solution deploys much faster and provides 100% workload coverage.

To help you cover your bases, Orca is providing a free Log4j Vulnerability Cloud Risk Assessment: go.orca.security/log4j.

03

Upgrade Log4j:

On the systems that allow upgrading, apply the latest Apache update for Log4j, which is currently version 2.17.1. While the 2.15.0 version removed the ability to resolve lookups and addressed issues to mitigate CVE-2021-44228, 2.16.0 patched CVE 2021-45046, 2.17.0 patched CVE-2021-45105, and 2.17.1 patches CVE-2021-44832. Since newer updates may still be released, it is best to check the Apache website for the latest updates.

04

If you cannot upgrade Log4j:

If for some reason you cannot upgrade Log4j, remove the JndiLookup class from the log4j-core jar.

05

Block outbound connections if necessary:

For externally facing systems that you are unable to patch, block outbound connections at the firewall, unless you are certain that this vulnerability is not exploitable or an updated version is released.

Note: On their own, Web application firewalls (WAF) do not provide adequate protection against attacks exploiting the Log4j vulnerability. Attackers may be able to bypass WAFs in numerous ways.

06

Monitor, monitor, monitor:

Assume there has already been a breach, and monitor vulnerable systems for unusual and malicious activities. We advise looking as far back as November since it is now apparent that this vulnerability may have been exploited by attackers on or before December 1.

07

Put your mitigations to the test:

After you have put in place all the mitigating controls, test your applications from outside your network to ensure that they block unauthenticated remote code execution.

08

Maintain visibility into applications & workloads:

To better prepare for future security incidents, create an asset inventory of all your software, applications, and systems in the cloud. You must have visibility into every workload, down to their installed applications and libraries.

09

Obtain the ability to search and query your assets:

It is also important to have a solution that allows you to easily query details on your cloud assets. For instance, in this case, to quickly identify which cloud applications are using Log4j. This enables you to triage critical CVEs even before a patch is released.

10

Get some rest:

You've probably had more than a few sleepless nights, so now get some well-deserved rest!