Rezilion

Log4j

REZILION VULNERABILITY BULLETIN

LOG4J IS A JAVA PACKAGE THAT is located in the Java logging systems and is essentially a Java library for logging error messages in applications. Log4j makes it easier for Java applications to log data, making it immensely pervasive.

What is the Log4j Vulnerability? Tracked as CVE-2021-44228, it is a zero day vulnerability, a remote code execution flaw in Log4j that allows hackers to take control of a system and all the information on it and puts millions of devices and customers at risk.

What devices are affected? Any device that is exposed to the internet and if it's running versions 2.0-beta-9 to 2.14.1 of Apache Log4j are at risk affected.

What is the associated risk? Due to the pervasive nature of Log4j in millions of devices globally signifies that threat actors have the ability to exploit these devices and steal customer and company data at scale. This has earned it a CVSS ranking of 10, which is classified as critical.

How Do You Discover Log4j?

THE BIGGEST CHALLENGE in iscovering Log4j is Java itself, specifically the way it is packaged. In Java you have dependencies. These dependencies are in the form of JAR files. It is also possible for a Java Archive (JAR) file to have another JAR to address a dependency. The closest analogy is Russian nesting dolls, and similarly you have a JAR nested in a JAR, nested in a JAR, and so on. These dependencies and packaging means that the vulnerabilities can be hidden deep down within an application making it almost impossible for traditional scanners to detect them.

SOFTWARE COMPOSITION ANALYSIS (SCA) SCANNERS — These scanners are reactive tools that find both nested and un-nested instances of Log4j, however they can only do so within the Continuous Integration (CI) & Development environments, leaving the entire production environment potentially exploitable and making them ineffective.

INFRASTRUCTURE SCANNERS — Within the production environments, during scheduled scans, the infrastructure scanners can discover Log4j, but not the nested types making them quite ineffective as well.

SUGGESTED REMEDIATION — While a fix (version 2.15.0) has already been released by Apache, systems still remain vulnerable. CISA has issued advisories for companies to first identify all their internet-facing devices that are running Log4j and upgrade them to version 2.15.0. Alternatively they should immediately apply the mitigations provided by the respective device vendors. Even with patches, it will take a significant amount of time to patch the millions of devices, and to complicate things, threat actors will discover additional threat vectors in that time making the task very complex.

The Rezilion Approach: Holistic Software Attack Surface Management

→ **End-to-End Visibility** — Rezilion's platform is able to create a dynamic, live Software Bill of Materials (SBOM) of both your development and production environments, providing a holistic view of all the software components present within your environment within seconds. Consequently, Rezilion is able to discover all instances of Log4j — nested and un-nested — all the way down to the class level.

→ **Run Time Visibility** — Just because Log4j is present in your environment, it does not mean it's exploitable. Rezilion's run time visibility lets you know what is loaded into memory and what is not so that you can really determine your exposure to Log4j. This is very important as it empowers you to focus on vulnerabilities that are exploitable.

→ **Continuous** — Rezilion's solution is dynamic and continually scans your development and production environment and updates the SBOM to provide you with an accurate and real time view into your vulnerabilities.

→ **One Tool** — You don't need any other tools, Rezilion covers both your development and production environments.

	SCA Scanners (app level)	Infrastructure Scanners	Dynamic SBOM (Rezilion)
What can be detected	Detects both nested and un-nested instances of Log4j	Can't detect any nested instances of log4j	Detects all instances of log4j down to the class level
Environment	CI/Development	Production	Single platform that covers end-to-end from development to production
Accuracy	Potential false positives as it is not able to distinguish between exploitable instances of Log4j and unexploitable ones		Run time visibility clearly lets you know what is exploitable in your environment
Frequency	Once when code is scanned	Once every few months or scheduled scan window	Dynamic and Continuous with real time view into your environment via dynamic SBOM
Time	Can take hours to days		Dynamic SBOM within seconds

To Learn more about the Rezilion platform and get a free risk assessment please visit us at www.rezilion.com