blueoptima

Responding to the Log4j vulnerability

In early December 2021, organisations were alerted that a critical, zero-day exploit had been discovered in older versions of the widely used Apache Log4j framework, prompting a frantic scramble to identify which applications were exposed to the flaw and ensuring that they were secured.

In this report, we'll take a look at how organisations responded to the threat, and how effective their efforts were in ensuring that their applications security was not compromised. While mitigation approaches, other than upgrading, are not considered, evidence suggests that upgrading was the preferred approach, especially given that the mitigation recommendations given by Apache were deprecated[1]. Our analysis does not identify whether the code changes made it into production, but shows the earliest changes in the version control system that should precede any releases to production.

BlueOptima monitors hundreds of thousands of software developers across some of the world's largest enterprises, giving us unique insights into the reaction of these organisations to the vulnerability.

What is Log4j?

Log4j is a widely used open-source library commonly used by Java applications.

Developers use Log4j to track what happens in their software applications or internet services. It's essentially a massive log of a system's or

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

application's activities. This practice is known as logging, and it is utilised by developers to keep track of user issues.[2]

What were the vulnerabilities identified?

Between December 9 and December 17 2021, three vulnerabilities were discovered across different versions of the Log4j framework, summarised in the table below:

observed a sample of 1000 active repositories to demonstrate how organisations have responded to the publication of the vulnerability over the past three months.

Initial response (first 30 days)

The data shows that upgrade work started the following Monday from the initial publication of vulnerability on December 9. This delay can be partially attributed to the publication being published late on a Friday, not giving developers the opportunity to make the necessary changes until the following week.

Date	Vulnerability Description	Severity	Possible Impact	Recommended response (at time of identification)
Dec 9,2021	Apache Log4j zero-day exploit discovered for version pre 2.15.0	CVE-2021-44228 10/10	It could lead to remote code execution (RCE) on underlying servers that run vulnerable applications	Upgrade to 2.15.0[1]
Dec 14,2021	Vulnerability carrying denial-of-service threat detected for version 2.15.0	CVE-2021-45046 9/10	Prone to create denial-of-service (DoS) attacks	Upgrade to 2.16.0
Dec 17,2021	Vulnerability of infinite recursion flaw in 2.16.0	CVE-2021-45105 5.9/10	StackOverflowError that will terminate the process	Upgrade to 2.17.0

1 Other versions were available for Java 6 & 7

How wide was the impact?

According to Akamai, since publication of the vulnerability, they saw multiple variants of the exploit at a sustained rate of attack traffic of around 2M attempts per hour.[3] Akamai has also seen growing evidence to suggest that the vulnerability may have been exploited for months, prior to the publication of the vulnerability.

How have companies responded?

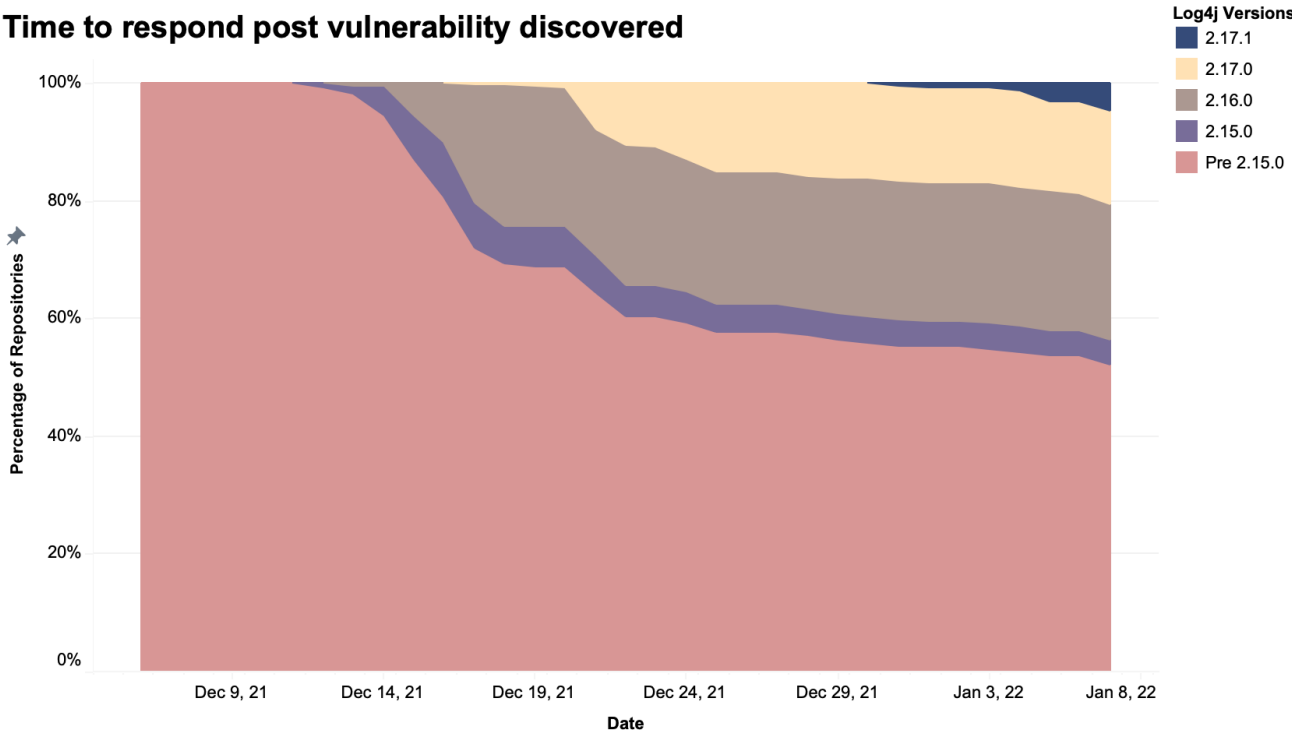
Using Code Insights from BlueOptima, a leading Software Composition Analysis (SCA) tool, we have

After 10 days from first vulnerability detection,~30% repositories upgraded Log4j to a secure version

After a month from first vulnerability detection,~50% repositories upgraded Log4j to a secure version

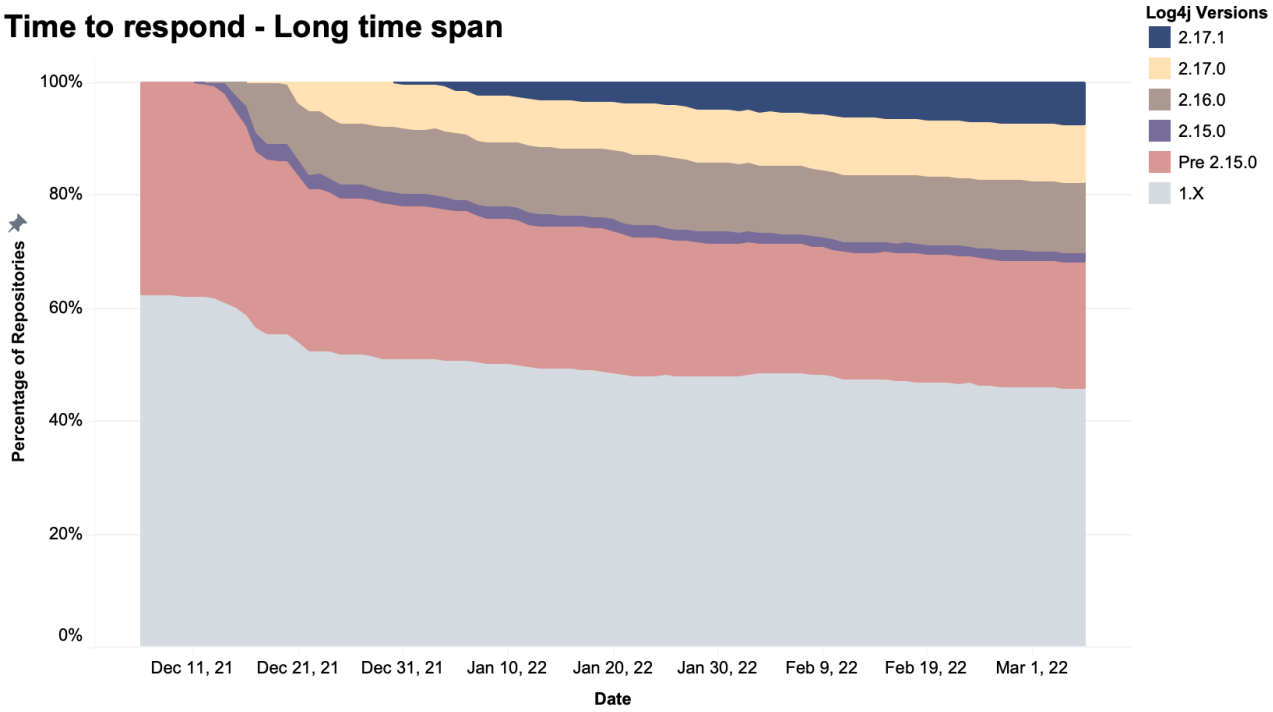
Perhaps the most shocking finding from this analysis is the number of active repositories running 1.X (End of Life 05/082015) and developers took this opportunity to upgrade to a 2.X version.

2 NSCV.gov.uk
3 https://www.akamai.com/blog/security/akamai-recommendations-for-log4j-mitigation



3 month review

- Log4j version pre 2.15.0 was widely being used across industries till Dec 2021
- From Dec 2021 to Jan 2022,~45% repositories had action taken
- In the most recent review, ~14% of repos were upgraded to the most recent 2.17.1
- However there are still over 40% of active repositories running vulnerabilities 1.X versions



Conclusions

Whilst over 50% of repositories have taken action to ensure that they are mitigating the vulnerabilities that have been identified, we are clearly able to see that there are still a high volume active of repositories who are highly likely to have taken limited to no action three months on from the initial identification of the vulnerability, leaving their applications and customers at risk from malicious parties.

One of the more concerning findings is that even whilst a significant proportion of repositories were upgraded in a relatively short period of time after the initial vulnerability was identified, many of them upgraded to 2.15 or 2.16, and then failed to upgrade further once additional vulnerabilities were identified in these versions.

This indicates that whilst many developers are highly active, they lack visibility into the overall composition of their software estates, leading to partially completed upgrades, but with no records of previous upgrades or outstanding vulnerabilities, meaning that they would need to start from scratch every time they needed to update.

Also concerning is data from Maven Central Administrator, Sonatype showing that even since December 10, 41% of the 31.4m Log4j downloads are vulnerable versions.

Code Insights from BlueOptima

Code Insights provides an objective insight into your estate's source code so that you can accurately and conveniently reduce application security risk while minimising the impact on technical debt incurred from shift-left initiatives.

Code Insights helps organisations to reduce risk in software development investments while minimising developer technical debt by scanning for Open Source & Internal dependencies to prioritise fixes on vulnerabilities and enable consistent Estate Management.

For organisations seeking to use data to drive collaboration to reduce technical debt, Code Insights is a SaaS tool that provides action-oriented strategies to reduce digital security risks in both the short and long term.

Contact us

p +44 207 100 8740
e enquiries@blueoptima.com
w www.blueoptima.com