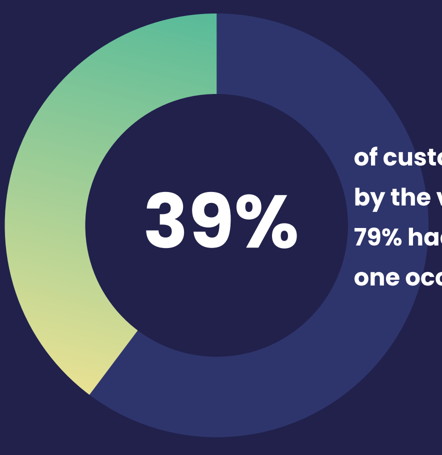
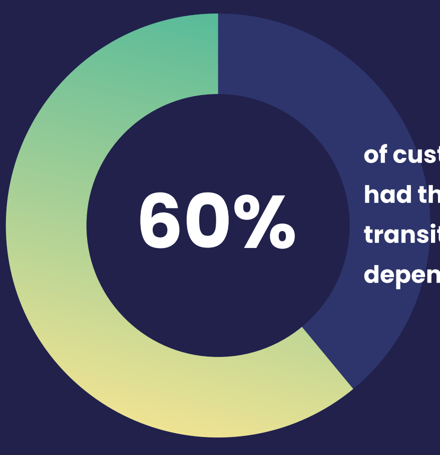
snyk

Log4Shell Remediation with Snyk

Snyk customers remediate 100x faster than industry average



of customers affected by the vuln. Of those 79% had more than one occurrence of it



of customers had the vuln in transitive dependencies

760% increase in projects created by customers within 2 days of Log4Shell surfacing

35 Log4Shell issues fixed on average by each Snyk customer

280 developer hours saved on average per customer by fixing with Snyk

$13,400 average ROI per customer from time savings having used Snyk

> Thanks for updating that database as quickly as you did. Snyk was the first to update... I felt very comfortable with understanding our posture, understanding who was impacted and being able to figure out next steps.

CVSHealth.

Amanda Alvarez, Technical Security Product Owner

> It was so easy to use Snyk to search issues by the CVE and quickly identify all the projects with the vulnerability. In addition to identifying all the application assets that needed to be fixed, its a simple way to watch the issues go down and ensure completeness.

David Matousek, Director, Lead Technical Product Owner, Cybersecurity Engineering

Manulife

> Props to @snyksec who made this a far less painful task than it's ever been in the past! I've no problem calling out good software when I see it.

sky betting & gaming

Glenn Pedgen, Security Vulnerability Manager, Sky Betting & Gaming

Time to Fix

	Log4j-2314720 vuln (CVE-2021-44228	**Log4j-2320014 vuln** (CVE-2021-45046)
Average Time to Fix for Container Issues	4.28 days	3.36 days
Average Time to Fix for Affected Customers	2.83 days	1.24 days
Affected Customers who Fixed within First 2 Days	91%	99%

Industry Average Timeto Fix	200 days
Snyk Customer Average Time to Fix	65 days

> The FTC intends to use its full legal authority to pursue companies that fail to take reasonable steps to protect consumer data from exposure as a result of Log4j, or similar known vulnerabilities in the future.

US FTC

Timeline

July 18th, 2013 - The code that introduces JNDI lookups and the vulnerability has been committed

November 24, 2021 - Alibaba Security Research team approaches Apache with a private disclosure

November 29 - Apache begins working on a new release (2.15.0) with a security fix

December 1 - The first rudimentary exploit attempts noted in the wild

December 5 - All fixes are merged into the master branch

December 9

- A GitHub user raises the suspicion that the fix relates to a security vulnerability
- A user creates a GitHub issue in the google/tsunami-security-scanner-plugins repository, identifying the RCE vulnerability in log4j
- Issue is leaked unofficially in a tweet by user p0rz9.
- PoC was published on Github

December 10

- Issue "officially" disclosed and given CVE (CVE not yet published to MITRE). Fixed version 2.15.0 was released.
- CVE Mitre updated, assigned (CVE-2021-44228)
- Critical severity added to Snyk SCA (CVE-2021-44228)

December 13 - Version 1.x medium severity vulnerability advisory (CVE-2021-4104)

December 14 - Version 2.15 medium severity DoS vulnerability discovered (CVE-2021-45046)

December 17 - CVE-2021-45046 upgraded to critical severity and recategorized as Arbitrary Code Execution

December 18 - CVE-2021-45105, a denial of service vulnerability was discovered in version 2.16 and fix version 2.17 issued.

December 28 - version 2.17 found to be vulnerable to very rare, nearly impossible to achieve scenario. 2.17.1 issued.

Jan 4th - FTC issues stern warning to US Companies about fixing Log4j

Contact us to learn more