TANIUM

Security and risk management in the wake of the Log4j vulnerability

A new risk landscape for security leaders.

Security and risk management in the wake of the Log4j vulnerability

The Log4j security vulnerability announced in December 2021 was a wake-up call for organizations of all kinds, large and small. An open-source software component used in countless software products and in-house tools can be exploited to give attackers control of an application, executing commands of their choice to install malware, exfiltrate data, or perform other malicious acts. Cybercriminals and nation-states are already crafting new forms of malware to take advantage of old versions of Log4j before they can be patched. And many of these attacks have already succeeded.

Because of the severity of this threat — which scored 10 out of a possible 10 in the industry-standard CVSS ranking — IT teams are now busy scanning their networks for any sign of vulnerable Log4j software. This search won't end anytime soon. Even if all local copies of the software were immediately found, vulnerable code could be reintroduced at any time by new application downloads, restoration of backups, network connections with partners, infected devices connecting to networks, and so on.

For security and IT leaders, this crisis creates both short-term and long-term challenges. In the short term, it requires them to ensure that their company's IT and security teams are working diligently, following best practices to detect and mitigate all instances of Log4j, especially where the software plays a role in business-critical operations.

Longer term, Log4j is a warning that the scope of risk management and threat detection is expanding. Going forward, IT organizations need tools and processes for discovering and tracking all the components in the software applications they build, buy or subscribe to. They need tools for scanning and accurately inventorying vast amounts of software without crashing mission-critical systems with repetitive search algorithms that max out the central processing unit (CPU) and memory of every system being searched.

They also need up-to-date, accurate, verifiable software "bills of materials" (SBOM) for all their software assets, so that when a new vulnerability is announced, they can immediately determine if the software on any of their endpoints (laptops, servers and other devices) are affected. That way, they can take immediate action to protect the ongoing business operations, data security and compliance of the organization.

In this eBook, we look at both the short-term and long-term issues related to software vulnerabilities like Log4j. We begin with a quick refresher on the Log4j vulnerability and its threat. Then we look at the longer-term issues of software management, compliance risks, and threat hunting, and how security leaders should rethink their roles and risk management processes as a result.

Let's begin.

The Log4j vulnerability and why it matters

On Friday, December 10, 2021, the Apache Foundation announced a serious security vulnerability (CVE 2021-44228[1]) in most versions of its open-source Log4j utility, a popular logging utility that has been embedded in thousands — if not hundreds of thousands — of software applications and services.

The vulnerability involves the Java Naming and Directory Interface (JNDI) API. This interface allows Java applications to look up data and resources from external directory services, including LDAP (an industry standard directory service used for managing user accounts) or DNS (the standard service used for mapping network addresses like www.google.com to network addresses on the internet.).

Because of the way Log4j software is written, an attacker can leverage JNDI functionality and trigger a victim system to retrieve software from a server the attacker controls. Commands in that software can be used to install malware, erase data, exfiltrate data, copy credentials, or perform other malicious actions.

The implications of this vulnerability are enormous. Thousands of software applications, including well-known commercial applications from some of the biggest names in enterprise software, are susceptible to being hacked and used for malicious purposes.

Log4j software is even found in everyday devices like TVs and store kiosks. Google estimates about 4% of the Maven Central Java repository[2], the most significant Java package repository in the IT industry, is compromised by this vulnerability. Attacks can be launched through web browsers, command lines and other types of inputs. One security researcher even suggested that presenting a carefully crafted QR code to a store price-checking kiosk[3] could give attackers control of the store's IT systems.

Repercussions for IT and security leaders

That's the vulnerability. Here are four repercussions that your organization needs to be concerned about.

- Ensuring that threat hunting is being conducted effectively.
- Ongoing threat hunting to find vulnerable Log4j code and prevent attacks.
- Managing new compliance risks.
- New compliance risks from regulatory agencies such as the U.S. Federal Trade Commission (FTC).
- Adopting Software Bills of Material (SBOM).
- The movement, now picking up momentum thanks to the White House and the U.S. Department of Defense (DoD), to create and maintain software bill of materials.
- Broader, real-time visibility into software and the risks it poses.
- The need for visibility into all software components and the risk they pose to mission-critical operations.

Let's examine each of these in turn.

THREAT HUNTING

Security teams and attackers are trying to outrace each other

Apache has issued patches to Log4j that fix the JNDI security vulnerability. Now the race is on: Can IT security teams find all the instances of Log4j in all their software on all their endpoints before attackers take advantage of this vulnerability to launch attacks?

Attackers aren't standing idly by. Within 72 hours of the vulnerability being announced, more than 800,000 attacks[4] on applications were launched. And the threat landscape is only getting worse.

- Cloudflare is reporting more than 1,000 exploits per hour against Log4j.
- A cybercriminal gang in China is using Log4j to infect VMware Horizon software with NightSky, a new family of ransomware.
- Coin miners have already launched attacks[5] against systems running Log4j, installing mining software on vulnerable systems.
- Attackers have used Log4j to add devices to two Linux botnets, Muhstik and Mirai.
- Attackers have successfully penetrated high-value targets, such as the Belgium Defense Ministry.
- Some popular red team toolkits have added features for exploiting Log4j vulnerabilities, putting this exploit within reach of an even broader number of people.

Vulnerable Log4j software might be installed on any endpoint's desk. Worse, the software might also be present in backups and gold images, meaning that Log4j instances removed this month might reappear in the future when a system needs to be restored or new systems need to be provisioned.

An ongoing problem requires an ongoing solution

Even aside from the inevitability of other vulnerable software components, the work of cleaning up Log4j is going to require a long-term effort from IT teams. Here's why.

Vulnerable software is still being downloaded

More than a month after the vulnerability was announced, more than 40% of recent downloads[6] of Log4J are versions still vulnerable to compromise. That's 40% of over 10 million downloads, creating a lot of opportunities for attackers.

Even older, vulnerable software may be lurking on the network

Across the industry, scans of enterprise networks are finding not only recent, vulnerable versions of Log4j but also 1.x versions of the utility, which Apache officially declared end of life in August 2015. It's also a sign that there are probably copies of Log4j lurking in unsuspected places across an enterprise.

Vulnerable software might be reinstalled

Even if an IT team manages to find and remove all vulnerable instances of the JndiLookup. class in Log4j, they still need to be on the watch for vulnerable instances being restored from backups or golden images[7]. They also need to worry about employees downloading vulnerable copies from the internet or third parties using vulnerable versions that end up creating opportunities for attackers to penetrate the network.

Ongoing reporting is critical. These reports should examine how tools and processes are being put in place for ongoing threat hunting for Log4j, recognizing that new copies of the vulnerable code will almost certainly be reintroduced to the network.

The FTC expects companies to patch vulnerabilities or face steep penalties

Because the impact of the Log4j vulnerability is so sweeping, the U.S. Federal Trade Commission (FTC) stepped in. On January 4, 2022, it issued a statement, pointing out that Log4j jeopardized the protection of consumer data, and reminding companies that "the duty to take reasonable steps to mitigate known software vulnerabilities implicates laws including, among others, the Federal Trade Commission Act and the Gramm Leach Bliley Act."

To ensure that no one mistakes the seriousness of its warning, the FTC cites its $700 million fine against Equifax in 2017 for that company's failure to patch a known Apache Struts vulnerability. That lapse in patch management led to a data breach that eventually exposed the private information of 143 million consumers and tarnished the Equifax brand.

The FTC doesn't see Log4j as a unique occasion for regulatory scrutiny. Rather, it expects companies to discover and promptly mitigate similar risks in the future. The FTC couldn't be more clear: "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."

With the FTC taking this action, it's possible that other regulatory bodies will follow suit. Data breach regulations have been multiplying in recent years. Will U.S. states like California or European nations follow the FTC's example by viewing companies that fall behind with patching to be derelict in their duties? Will penalties be enacted, and companies be made an example of?

It seems imprudent to assume that the FTC is the only regulatory body anywhere that will view patch management as a matter of the first importance.

The bottom line for security teams, IT organizations, and the executive leadership of every organization: The tools and processes that IT security teams put in place now to address the Log4j vulnerability should be used continuously in the future to detect and mitigate similar risks.

Develop a comprehensive plan for hunting for other vulnerable code in the future.

Organizations need a systematic way of tracking the contents of all software

Log4j won't be the last serious vulnerability IT teams go hunting for. In fact, within several weeks of Apache's announcement about Log4j, a serious vulnerability in a Linux permission tool was announced. Because of this vulnerability, dubbed **PwnKit,** attackers can run commands as privileged users in Linux environments. The vulnerability is found in all major Linux distributions.

These vulnerabilities point to the importance of knowing the contents — the "bill of materials," as it were — of every piece of software and every operating system configuration running in an organization.

The White House Executive Order on Improving the Nation's Cybersecurity[8], issued on May 14, 2021, gives this idea an official push. The order notes:

> *The development of commercial software often lacks transparency, sufficient focus on the ability of the software to resist attack, and adequate controls to prevent tampering by malicious actors. There is a pressing need to implement more rigorous and predictable mechanisms for ensuring that products function securely, and as intended. . . . Accordingly, the Federal Government must take action to rapidly improve the security and integrity of the software supply chain, with a priority on addressing critical software.*

The order calls for the Director of Commerce to work with the National Institute of Standards and Technologies (NIST) to recommend practices and guidelines for creating software bills of materials. Specifically, it calls for:

- *Maintaining accurate and up-to-date data, provenance (i.e., origin) of software code or components, and controls on internal and third-party software components, tools, and services present in software development processes, and performing audits and enforcement of these controls on a recurring basis.*
- *Providing a purchaser a Software Bill of Materials (SBOM) for each product directly or by publishing it on a public website.*

Any software application is only as secure as the components it includes. This Executive Order acknowledges that no organization, public or private, can place confidence in its software without knowing the components that go into its software.

Create a strategy for adopting a software bill of materials. Internal development organizations should create them for their own software. As the SBOM movement gains steam, IT security teams and purchasing departments should ask for SBOMs from third parties or partners before installing or connecting to their software.

Beyond Log4j to continuous threat hunting

The Log4j vulnerability highlights the need for organizations of all kinds to improve their capacity for discovering vulnerabilities, patching them at scale, and remediating threats on an ongoing basis. It also highlights the need for organizations to adopt SBOMs so that any organization using an application can quickly understand its contents.

With the FTC committed to penalizing companies that leak data because of any vulnerabilities similar to Log4j, a major regulatory agency is signaling that longstanding lapses in patch management will no longer be tolerated. Failure could result in fines up to hundreds of millions of dollars.

IT teams need to act now, not just to find and fix Log4j vulnerabilities, but also to implement tools and processes for finding software component vulnerabilities generally. If six months from now, another open-source software component is found to contain a security flaw, security teams will need to respond quickly to detect and fix all instances of that component.

Ideally, security teams would have real-time visibility into threats. If a new vulnerability is announced, they could consult SBOMs and quickly understand their exposure. Fast, automated tools would scan endpoints on premises and in the cloud, providing additional coverage. When vulnerabilities are pinpointed, they can be managed and patched. And the automation and efficiency of this process make it repeatable so that even if a vulnerability is reintroduced to a network, it can be detected and remediated quickly.

Conclusion

IT and security leaders should understand both the short-term and long-term implications of the Log4j vulnerability. Business as usual has changed. IT organizations need new tools and processes, and departments, as varied as security and purchasing, need to change their expectations.

IT organizations of all kinds need to improve their capacity for finding specific software components at scale. At the same time, organizations need to be ready to produce and receive SBOMs as part of software sales, purchases and deployments.

Assess your organization's risk posture

Request a five-day, no-cost risk assessment to get a comprehensive view of risk posture across your organization.

Get risk assessment →

Sources:

1. https://nvd.nist.gov/vuln/detail/CVE-2021-44228
2. https://security.googleblog.com/2021/12/understanding-impact-of-apache-log4j.html
3. https://searchitoperations.techtarget.com/news/252510999/Log4j-vulnerability-nightmare-A-DevSecOps-wake-up-call
4. https://www.scmagazine.com/news/cloud-security/log4j-reaching-pandemic-level-exploit-numbers
5. https://isc.sans.edu/diary/Log4Shell+exploited+to+implant+coin+miners/28124
6. https://www.computerweekly.com/news/252511846/Almost-half-of-Log4j-downloads-still-dangerously-exposed
7. https://www.techopedia.com/definition/29456/golden-image
8. https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/