Written Testimony of:

Jen Miller-Osborn
Deputy Director of Threat Intelligence, Unit 42
Palo Alto Networks

Before the:

Homeland Security and Governmental Affairs Committee
United States Senate

Regarding:

“Responding to and Learning from the Log4Shell Vulnerability”

February 8, 2022
10:00am

Chairman Peters, Ranking Member Portman, and distinguished members of the committee, I am honored to appear before you today to discuss the impact and scope of the “Log4Shell” vulnerability and how policy makers and network defenders can better fortify defenses for future national-level vulnerabilities of this magnitude.

This committee’s continued commitment to advancing thoughtful, bipartisan cybersecurity policy and law is appreciated. On behalf of Palo Alto Networks, I offer my commitment to work in partnership with you and your staff as you continue addressing a range of critical cybersecurity issues.

For those not familiar with Palo Alto Networks, we were founded in 2005 and have since become the global cybersecurity leader. We serve more than 85,000 enterprise and government organizations - protecting billions of people - in more than 150 countries. We support 95 of the Fortune 100 and more than 71% of the Global 2000 companies, and are partnered with elite technology leaders.

Practically speaking, this means that we see a lot. This expansive telemetry, coupled with the contextual threat intelligence from our Unit 42 team, where I am privileged to be a senior leader, gives us broad visibility into the global attack surface of our digital infrastructure.

We are committed to using this visibility to be good cyber citizens and integrated homeland security partners with the United States Government.

As it relates to the focus of this hearing, it’s important to first take a step back and understand why Log4Shell matters. If it feels like Log4Shell is just the latest in a string of vulnerabilities that the cybersecurity community must rally in response to, you are right. Log4Shell is not the first vulnerability garnering significant public interest, and it almost certainly won’t be the last.

That’s why it’s important to look at Log4Shell both as a standalone vulnerability that demands discrete analysis and reflection, *and* as the latest in a string of national-level vulnerabilities that impact federal systems, critical infrastructure, and state and local networks alike.

Starting with the latter, I cannot stress enough the foundational importance of accurately understanding the size of your internet-exposed attack surface. If you do not understand the totality of your digital footprint through the eyes of the adversary, your security baseline is inherently incomplete. The ability to rapidly discover and remediate new vulnerabilities - Log4Shell or otherwise - is going to be predicated on production level mapping of your networks. This is true for organizations large and small, public and private.

Reinforcing this point is the Biden Administration’s well-crafted Zero Trust Strategy that was finalized two weeks ago. It highlights that to effectively implement a zero trust architecture, “an organization must have a complete understanding of its internet-accessible assets, so that it may apply security policies consistently and fully define and accommodate user workflows.” It goes on to acknowledge that “in practice, it can be very challenging for a large, decentralized organization to track every asset reliably.”

Bottom line: you can’t secure what you can’t see or aren’t actively monitoring.

Assessing Log4Shell

Now, zooming in to Log4Shell. Apache Log4j 2 is an open source Java-based logging framework that became the mainstream version of Log4j in 2015. Open source software generally refers to code that is managed in a decentralized manner that is publicly accessible. Think of it as the crowd sourced model of software development. Log4j 2 is leveraged within numerous software applications, and by many estimates is embedded within hundreds of millions of devices globally.

On December 9, 2021, a new Remote Code Execution (RCE) vulnerability in Apache Log4j 2 was identified and observed being actively exploited. It is this specific vulnerability which has become known as Log4Shell. RCE vulnerabilities are generally seen as having the potential for high consequence as they allow the attacker to remotely command devices in the victim's environment.

Shortly after the Log4j vulnerability was discovered, investigation revealed that exploitation was incredibly easy to perform. Due to the recency of this discovery and the complexity of remediating devices with embedded Log4j vulnerabilities, there are still many servers, both on-premises and within cloud environments, that have yet to be patched. As is typical for many high severity RCE vulnerabilities, adversaries have conducted massive scanning activity for Log4Shell with the intent of seeking out and exploiting unpatched systems.

During the Log4Shell exploit lifecycle, adversaries identify potentially vulnerable systems, trigger a file download as part of the remote code execution, and then deliver what's known as the "payload" - the part of the malware that is crafted for a specific malicious purpose. We have subsequently seen exploitation for coin mining (to commandeer computing horsepower from the victim to perform cryptocurrency mining), for hijacking victim networks as part of larger botnet systems (defined as network of computers controlled as a group without the owners' knowledge), and to infect systems with ransomware. A small minority of observed exploitation has been attributed by security researchers to nation state linked groups.

We highly recommend that organizations upgrade to the latest version (2.17.1) of Apache Log4j 2 for all systems. This version also patches three additional Log4j vulnerabilities discovered later in December 2021.

Discovering Log4Shell vulnerabilities across networks is more complicated, and the problem is more widespread than other notable recent vulnerabilities. The open source nature of Log4j software means it is used in potentially hundreds of millions of devices and services, often without the end user even knowing that it is present in the code. Additionally, due to the embedded nature of this particular vulnerability, network defenders must be able to mimic the exploit to accurately discover vulnerable instances.

This represents an additional layer of complexity beyond more elementary scanning and reinforces the importance of understanding your baseline attack surface. Otherwise, you will be caught flat-footed in your discovery and remediation efforts, without an accurate map of the playing field where you need to look.

Software Assurance Practices and Open Source Software Security

A number of best practices currently exist that promote software integrity. These approaches focus on integrating security tools into the engineering lifecycle early on - known as "shift left" security - to help detect any inadvertent vulnerabilities in code.

In particular, we recommend adopting:

- Static Application Security Testing (SAST), also known as "white box testing," a process of reviewing source code to identify security vulnerabilities.
- Open Source Software Vulnerability Analysis (OSSVA), which identifies vulnerabilities in third-party components and provides visibility into third-party code for control across the software supply chain.
- Container Vulnerability Analysis (CVA), a process of evaluating containers against common container misconfiguration and software package vulnerabilities.
- Secure Infrastructure-as-Code, which identifies, prevents and remediates security misconfigurations in infrastructure code before deployments in cloud such as: unauthorized privileges, network exposure, and public storage buckets.

Log4Shell has rightfully highlighted the urgent need for ongoing conversations about open source software security, as this panel will highlight in more depth. Best security practices for incorporating open source software into products include code scanning to identify any open source packages with vulnerabilities, and steps to check for security and integrity prior to approval for use in software products.

Our product security team was able to leverage these practices to quickly identify the vulnerable version of the library and then create a policy to block that version from being used in our product development. This safeguard was applied to engineering workstations, source code repositories, and continuous integration and development pipelines. Additionally, we leveraged a range of tools to identify and mitigate existing workloads that could be vulnerable. As we consider system-wide approaches, the Administration's work to promote Software Bills of Materials (SBOM) is a promising endeavor, though still in its early stages.

Operational Collaboration

Palo Alto Networks collaborates extensively with key stakeholders across the U.S. Government and with like-minded countries across the globe on both policy and operational matters.

We are proud to be a founding alliance member of the Joint Cyber Defense Collaborative (JCDC), a promising operational collaboration body that brings federal government and industry players together to move from information sharing to information *enabling*.

The most recent JCDC engagement, which occurred after Log4Shell was first discovered, presents an important use case of the long-term opportunity this collaboration vehicle presents. It can be an exemplar of successful public-private sector cooperation - specifically, the JCDC working as a venue for commercial competitors to act as peers, and share rapidly developing situational awareness to help secure our National Critical Functions. We appreciate the commitment from CISA Director Jen Easterly to continue maturing the JCDC and maximize the bidirectional value it brings.

We are also pleased that Wendi Whitmore, who runs our Unit 42 Threat Intelligence team, has recently been appointed to the Cyber Safety Review Board (CSRB) whose first tasking will be determining "key facts related to the root-cause of the Log4j vulnerabilities and exploitation and weaponization of the vulnerabilities."

In addition to our active participation in the JCDC and CSRB, Palo Alto Networks is a member of the President's National Security Telecommunications Advisory Committee (NSTAC), where industry can provide advice to the White House and other senior U.S. Government stakeholders on national security policy and technology issues; the Executive Committee of the Information Technology Sector Coordinating Council (IT-SCC), which serves as the principal coordinating body between the Department of Homeland Security and IT sector; and the Defense Industrial Base Sector Coordinating Council (DIB-SCC).

We are also an active participant in the DHS ICT Supply Chain Risk Management Task Force and were pleased to have been selected as a technology partner in NIST's National Cybersecurity Center of Excellence's 5G Cybersecurity Project.

Finally, we maintain robust threat intelligence sharing partnerships with DHS, the Intelligence Community and across the international community to share technical threat data and collaborate to support government and industry response to significant cyber incidents, like SolarWinds, Microsoft Exchange, and Log4Shell.

Recommendations

While there is no silver bullet, there are several tangible steps that will provide immediate risk reduction for the response to Log4Shell and future vulnerabilities:

1. Enumerate an accurate denominator of your digital infrastructure. This should be a foundational aspect of any national-level incident response and is applicable across federal and non-federal entities. You can't protect what you can't see.

2. Look for ways to automate compliance with vulnerability management policies. We applaud CISA for building and maintaining a catalog of Known Exploited Vulnerabilities, but manual reporting across 100+ federal civilian agencies is unlikely to stay ahead of the adversary.
3. Drive industry-wide commitment to Development Security Operations (DevSecOps). Impressive work is already being done in this arena, but the community would be well-served by increasing adoption of existing development tools to control access to open source components. These tools can scan all of the open source packages for both integrity and security before they are approved and allowed for engineering teams to use in products. Our recently released State of Cloud Security Report 2022, which surveyed over 3,000 IT professionals, found that organizations with tightly integrated DevSecOps principles were more than seven times likely to have strong or very strong security posture. They were also more than nine times more likely to have low levels of security friction.
4. Promote common visibility and automated security capabilities across your entire environment. Data from cloud, endpoint, and on-premise systems should be seamlessly integrated.
5. Lastly, cyber hygiene basics still remain as important as ever.

Closing

The cybersecurity threat landscape is only getting more complex. Whether it's vulnerabilities like Log4Shell, the ongoing ransomware threat, or our dynamic geopolitical environment (as our recently published research on a Russian-linked Advanced Persistent Threat Group actively targeting Ukraine reinforces) - cybersecurity will undoubtedly remain a core pillar of our national security posture. Now, more than ever, this demands a whole-of-society approach.

Thank you for the opportunity to testify, and I look forward to your questions.