"Responding to and Learning from the Log4Shell Vulnerability"

Opening Statement by David Nalley
President, Apache Software Foundation
Senate Committee on Homeland Security and Government Affairs
February 8, 2022

Chairman Peters, Ranking Member Portman, and distinguished members of the Committee: thank you for the invitation to appear this morning.

My name is David Nalley, and I am the President of the Apache Software Foundation (ASF). The ASF is a non-profit public-benefit charity established in 1999 to facilitate the development of open source software. Thanks to the ingenuity and collaboration of our community of programmers, the ASF has grown into one of the largest open source organizations in the world. Today, more than 650,000 contributors around the world contribute to more than 350 ongoing projects, comprising more than 237 million lines of code.

Open source is not simply a large component of the software industry -- it is one of the foundations of the modern global economy. Whether they realize it or not, most businesses, individuals, non-profits, or government agencies depend on open source; it is an indispensable part of America's digital infrastructure.

Projects developed from open source, like Log4j, tend to resolve problems that many people have, essentially serving as reusable building blocks for solving those problems. This enables faster innovation because it eliminates the need for every company or developer to reimplement software for already solved problems. This efficiency allows programmers to stand on the shoulders of giants. The ASF provides a vendor-neutral environment to enable interested programmers – oftentimes direct competitors of one another – to do this common work together in transparent, open-handed cooperation.

This is the essence of open-source software: brilliant individuals contributing their time and expertise to do unglamorous work solving problems – many with the intent of incorporating the results into their employer's products. And it's why I've dedicated my professional life to it.

Log4j – first released by Apache in 2001 – is the product of just this kind of collaboration. It performs a particular set of functions, like recording a computer's operating events, so well that it has been used in products as diverse as storage management software, software development tools, virtualization software and (most famously) the Minecraft video game. As Log4j's footprint grew over the years, so did its feature list. It was a 2013 addition to Log4j, along with a part of the Java programming environment, that combined in such a way that exposed this security flaw.

The vulnerability was reported to Apache's Log4j team late November 2021, after having been latent for many years. The Apache Logging project, and Apache's Security team immediately got to work addressing the vulnerability in the code. The full solution was released approximately two weeks later. Given the near ubiquity of Log4j's use, it may be months or even years before all deployed instances of this vulnerability are eliminated. As a software professional myself, I am proud of how the Logging project and the ASF’s security team (and many others across the ASF’s projects) responded and remediated last fall. We acted quickly and in accordance with practices we have adopted over many years of supporting a diverse set of open source projects. We will continue to develop our projects in responding to and preventing security vulnerabilities.

Moreover, every stakeholder in the software industry – including its largest customers, like the federal government – should be investing in software supply chain security. While ideas like the Software Bills of Materials won’t prevent vulnerabilities, they can mitigate the impact by accelerating the identification of potentially vulnerable software. However, the ability to quickly update to the most secure and up-to-date versions remains a significant hurdle for the software industry.

The reality is that humans write software, and as a result there will continue to be bugs, and despite best efforts some of those will include security vulnerabilities. As we continue to become ever more connected and digital, the number of vulnerabilities and potential consequences are likely to grow. There is no easy software security solution - it requires defense in depth – incorporating upstream development in open source projects, vendors that incorporate these projects, developers that make use of the software in custom applications, and even down to the organizations that deploy these applications to provide services important to their users.

Rather than shying away from this risk, I submit that software developers, open-source communities, and federal policymakers should face it head-on together – with the determination and the vigilance it demands.

Thank you again, and I look forward to answering any questions you might have.