Azul Byline for Tech Decisions Australia

azul

Best Practices for Applying the Essential Eight Framework to Java Security

Part 1. The Essential Eight Maturity Model

The Essential Eight is a maturity model for improving data security developed by the Australian Cyber Security Centre (ACSC) as the foundation of its Strategies to Mitigate Cyber Security Incidents.

The Essential Eight is designed to protect Microsoft Windows-based internet-connected networks, but may be applied to cloud services, enterprise mobility, and other operating systems as well.

When implementing the Essential Eight, organisations should identify a target maturity level from Zero to Three (summarised below) that is suitable for their environment.

Maturity Level	Summary
Zero	Weaknesses in an organisation's overall cyber security posture, compromising the confidentiality of their data, or the integrity or availability of systems and data.
One	Adversaries are content to leverage widely available commodity tradecraft to gain access to and control of systems, e.g., opportunistically using a publicly-available exploit for a security vulnerability in an unpatched internet-facing service.
Two	Adversaries are willing to invest more time in a target and, perhaps more importantly, in the effectiveness of their tools. They may make better attempts to bypass security controls implemented by a target and evade detection.
Three	Adversaries can exploit opportunities provided by weaknesses in their target's cyber security posture, such as the existence of older software or inadequate logging and monitoring. Adversaries make swift use of exploits when they become publicly available.

The first of ACSC's basic guidelines in the Essential Eight is Assessing Security Vulnerabilities and Applying Patches, which "is critical to ensuring the security of systems."

ACSC's Baseline Security Vulnerabilities Guidelines to Mitigate Cyber Threats:

1. Apply software patch within 48 hours of exposure.
2. Support tightly coupled and uncoupled software builds.
3. Scan for vulnerabilities.

Part 2. The Log4j Vulnerability

Threats abound, as evidenced by the Log4j vulnerability, a.k.a. Log4Shell, which exploded in the world's consciousness last December. Log4j is not part of Java itself. It is a software library to provide log functionality used as a building block in application development, thus simply dropped in as a block of code by developers in millions of products.

Because the Log4j vulnerability is so widespread, so serious (a rare score of 10.0, the highest possible), and so trivial to exploit (via a line of code copied-and-pasted remotely), it represents "the most serious vulnerability I have seen in my decades-long career," says Jen Easterly, security director of the U.S. Cybersecurity and Infrastructure Security Agency.
"If left unfixed," explains the ACSC, "malicious cyber actors can gain control of vulnerable systems; steal personal data, passwords and intellectual property; and install malware such as backdoors for future access, cryptocurrency mining tools and ransomware."

Part 3. The Critical Choice of Java Vendor

Java, given its broad (over 9 million developers worldwide) and mature (nearly 27 years since its inception) footprint in the IT landscape, will play an outsized role in any application of the Essential Eight framework. To ensure the Java platform is as secure as possible, it's critical to install updates quickly when necessary.

Which updates, from whom? There are two kinds of patch updates for Java, and it's important to understand the differences in what they are, where you can get them, and when they should be applied.

CPUs and PSUs

Each quarter, new Java vulnerabilities are discovered and published, together with their solutions, in both Critical Patch Updates (CPUs) and Patch Set Updates (PSUs).

- CPUs are security-only updates which, as defined by Oracle, "contain fixes to security vulnerabilities and critical bug fixes." The two key words here are *security* and *critical*—so *these updates should be deployed immediately*.

- PSUs contain all the fixes in the corresponding CPU *as well as* additional non-critical fixes. *Enterprises often defer deploying these updates until they are fully vetted by the community.*

Where can you get these updates? One option is a free build of OpenJDK. Free may sound appealing, but be forewarned:

Did you know that every free build of OpenJDK contains *only* PSUs?

And PSUs, as just noted, should *not* be deployed immediately to fix a vulnerability. So, a free OpenJDK will leave you exposed.

Did you know that security-only patches are available *only* from Oracle and Azul?

There are other commercial Java vendors, but, from a security standpoint, your choice is between Oracle and Azul.

Did you know that Azul has been doing Java longer than Oracle?

Azul, the only company 100% focused on Java and the JVM, has been supporting Java deployments for over 19 years.

Contact Azul

Grant Christian – Director ANZ

gchristian@azul.com

+61 424 661 795

Sydney

www.azul.com