Summary: SCS 9001 Potential Impacts to the Log4j Security Breach

Cyberattack: Apache Log4j Vulnerability (Log4Shell) (2021)

Breach Background

Java is a programming language supported by associated libraries and frameworks and used to develop applications estimated to be deployed in over 3 billion devices of all types, world-wide. Log4j is one such library, it is used pervasively, and provides logging services. It is an open-source library and offered under the Apache license.

There are two versions of the Log4j library, Version 1.x and Version 2.x. They are incompatible. The vulnerability was originally introduced in Version 2.0, which was a complete re-write of the library and originally released in 2014. This vulnerability has existed and gone undetected for over 6 years. Version 1.x does not have the vulnerability, but due to its age, it contains other known but less critical vulnerabilities. Version 1.x reached end of life in 2015.

The Log4j vulnerability, or Log4Shell as it is known, is a “zero-day” vulnerability. A zero-day vulnerability means it was previously unknown and as a result can be particularly damaging as it remains an exposure while the industry reacts in providing mitigation solutions. It is identified in the NIST Vulnerability Database as CVE-2021-44228 and has been assigned the highest vulnerability score of 10/10 Critical in the CVSS (version 3.x) rating system.

Many security experts have called this vulnerability the most serious they have ever seen. Prior to detection, bad actors are likely to have been exploiting the vulnerability for an unknown period of time and causing unknown levels of damage and exploitation in their targets. As an example of the extent of attack, Check Point monitored millions of attacks being initiated by hackers and observed a rate of over one hundred attacks per minute resulting in over 40% of business networks being attacked internationally.

Breach Description

Log4Shell is a remote code execution (RCE) vulnerability that enables hackers to execute arbitrary code and take full control of impacted devices.

In Log4j Version 2.x, new capabilities were added, and the vulnerability was introduced in one of those new capabilities. Specifically, Log4j Version 2.x enables users to provide formatting information to be applied to log messages. The capability includes the ability to perform a lookup in a directory server or database for needs like name resolution. This capability is built on a mechanism called Java Naming Directory Interface or JNDI which supports a variety of protocols such as LDAP (Lightweight Directory Access Protocol), DNS (Directory Name

Services), and others. In addition to formatting information, the user can also provide the name or address of the database in which to do the lookup.

The manner in which the attack works is as follows:

> An attacker finds a web application that allows the user to input data into fields accepting text and likely to be logged. The attacker inserts text and includes optional formatting information along with the address of an external directory server – operated by the hacker – to which Log4j will do a lookup. The directory server responds with malicious code, which is then unknowingly executed on the target system.
>
> For Log4Shell to be used successfully, there are actually several vulnerabilities that are exploited:
>
> 1) The use of a vulnerable version of an open-source software library,
>
> 2) Poor software design practices that do not validate user input,
>
> 3) Lack of effective network monitoring to detect anomalous behavior with attempts to access external systems, and
>
> 4) Lack of effective network protections to block malicious traffic from leaving the enterprise to unknown external systems.

SCS9001 Impacts

Certification to SCS 9001 may have provided benefit in potentially avoiding the vulnerability, or at least in more quickly addressing the problem and limiting exposure.

As an example, CISA recommended the following immediate actions to mitigate the risk of Log4Shell:

- CISA: discover all internet-facing assets that allow data inputs and use Log4j Java library anywhere in the stack

  SCS 9001 requires that a full asset inventory be compiled and kept current, including the interactions between all systems. Having an asset inventory would have provided a quicker determination on exposure.

- CISA: discover all assets that use the Log4j library.

  SCS 9001 requires that a Software Bill of Materials (SBOM) be produced and kept current for every IT asset. Having the SBOMs would have enabled immediate determination of the software composition of every device and accordingly, which ones had the vulnerability.

- CISA: update or isolate affected assets. Assume compromise, identify common post-exploit sources and activity, and hunt for signs of malicious activity.

  SCS 9001 requires full network monitoring for the detection of unauthorized access and anomalous activity.

Considering other recommendations from leading security authorities, an SCS 9001 certification may have provided additional benefits such as:

- SCS 9001 requires a network architecture be based on a Zero Trust Architecture (ZTA) using principles as described in NIST SP 800-207. A ZTA protects networks and resources by securing all communications regardless of physical location and ensuring access to resources is granted on a per-session basis and based on a dynamic policy including the observable state of client identity, application/service, and the requesting asset – and may include other behavioral and environmental attributes.
- SCS 9001 promotes the Least Privilege Policy which means network communications traffic should be denied by default unless explicitly enabled. There is no good reason to allow applications to be accessing unknown directory servers.
- SCS 9001 requires that organizations perform deep packet inspect and traffic throttling for timely response to network-based attacks associated with anomalous ingress or egress traffic patterns.
- SCS 9001 requires that organizations restrict, disable, or prevent the use of nonessential functions, ports, protocols, and services.
- SCS 9001 requires that network environments and traffic are monitored to detect and restrict traffic between trusted and untrusted connections.
- SCS 9001 requires that perimeter firewalls be implemented and configured to restrict unauthorized traffic.
- SCS 9001 requires a detailed Incident Management Process to quickly alert impacted customers and mitigate problems. Once the vulnerability is understood, such a process can be used to more quickly alert customers, communicate remediation approaches, and more quickly address the breach.
- SCS 9001 requires a Secure Development process, in which security is a design requirement including the address of common flaws such as failure to check input parameters and the testing of such flaws.

Measurements

The following measurements defined within SCS 9001 may be of value for this type of attack:

- Vulnerabilities: the detected vulnerabilities in the reporting period, by severity as defined by the NIST NVD CSS V3.0 scoring system.
- Update timeliness: how current systems are to current software and patch levels.
- Unauthorized Access: the time to discover instances of unauthorized access and the organization's responsiveness to those events upon discovery.

References

There is a large amount of information available on Log4Shell, some references are:

- https://cisomag.eccouncil.org/log4j-explained/
- https://thomsuninfocare.com/what-exactly-happened-with-log4j/
- https://nvd.nist.gov/vuln/detail/CVE-2021-44228
- Log4J (Log4Shell): Mitigating the impact on your organization | Synopsys
- Log4j – Apache Log4j 2.