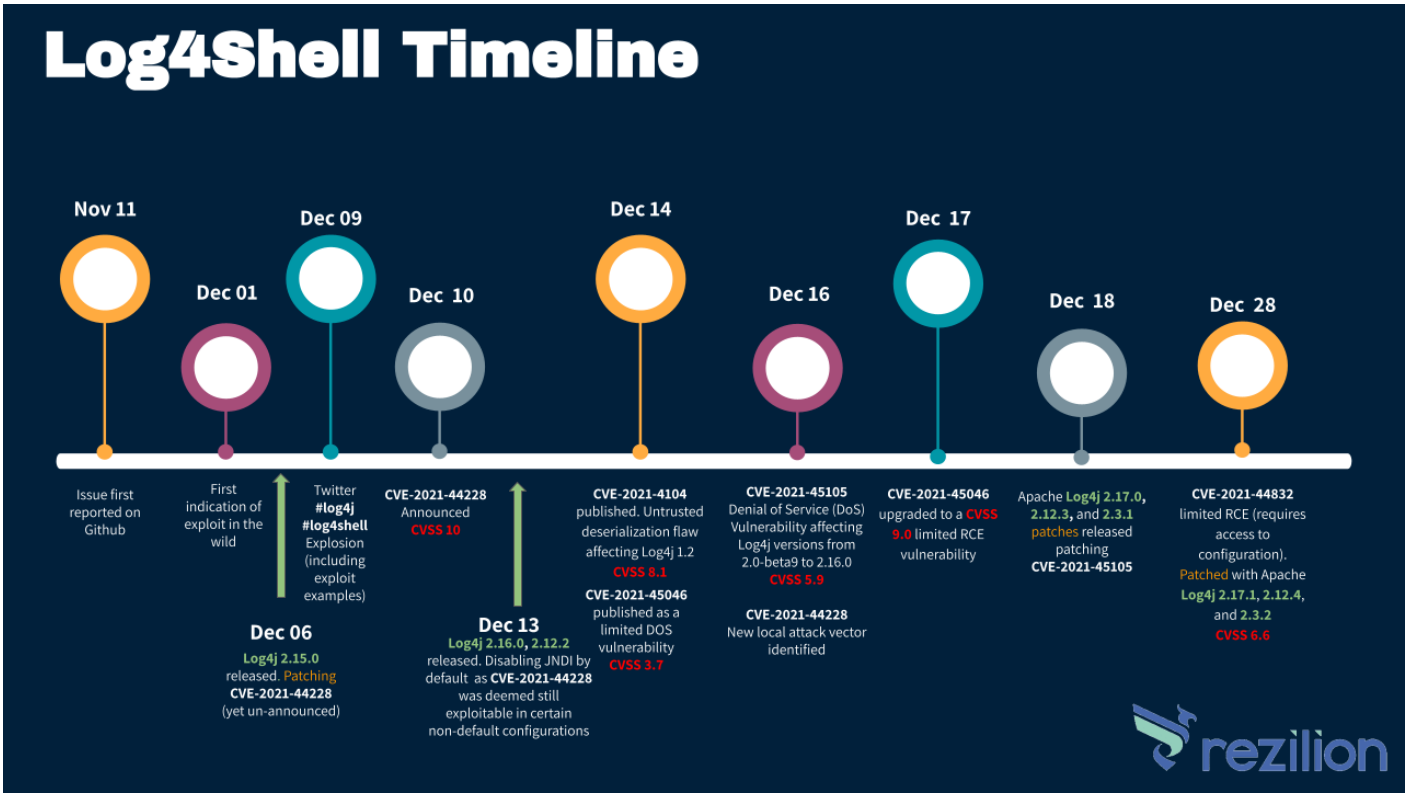
Log4shell - exploit log4j

Historique



Description

What Happened?

Apache Log4j2 is a ubiquitous logging framework (library) used in many open-source Java applications. On December 10th, a critical third-party zero-day exploit in Log4j2 called Log4Shell was discovered. Threat actors can send malicious code that eventually gets logged by Log4j2. From there, they can remotely seize control of the server.

How Bad Is It?

Bad. What makes Log4Shell (now known as CVE-2021-44228) so dangerous is how easy it is to use. In fact, it’s so easy to use that the targeted application just has to write one string in the log for the attacker to be able to upload their own code and assume control of the server.

To illustrate its simplicity, even changing an iPhone's name can trigger the vulnerability. This vulnerability is exploitable with or without authentication, thus increasing the overall severity, scale and impact potential. The Apache Software Foundation assigned the maximum CVSS severity rating of 10 to Log4Shell, as millions of servers are potentially affected.

Who Might Be Impacted?

Pretty much everyone. The Log4j2 is an open-source Java-based logging framework commonly incorporated into Apache web servers, the library is used in numerous Apache frameworks services, and application frameworks in the Java ecosystem use this logging framework by default. For instance, Apache Struts 2, Apache Solr and Apache Druid all may be affected. Aside from those, Apache Log4j2 is also used in many Spring and Spring Boot applications.

Cloud services like Steam, Apple iCloud and applications like Minecraft have already been found to be vulnerable. The affected Apache Log4j2 Versions are 2.0 <= Apache Log4j <= 2.14.1.

What Should You Do?

Large organizations like Amazon Web Services can easily update their web servers; however, the same Apache software is also often embedded in third-party programs, which can typically only be patched by their owners. Even if you do not use Log4j2 directly in your application, your vendors might.

Here are some steps that you can take:

1. Identify

Identify the vulnerable technologies that your organization might be using. Begin with an architecture review, and then identify the technologies that might be leveraging Java as a first lead.

You'll also need to identify which of your critical third parties might be vulnerable, and also if they are taking action to mitigate their vulnerability as quickly as possible. Recognizing Log4j2 as a technology is not possible from external assessment processes because it is an internal technology used by vendors. Therefore, inquiring with vendors directly is the optimal solution. Another valuable resource created by Authomize is a list of Affected Apps and Affected Components with vendors.

2. Protect

Permanent mitigation: You can remediate this vulnerability by updating to version 2.15.0 or later, or follow the proposed mitigation on Apache Log4j Security Vulnerabilities. The log4j-core.jar is available on the Apache Log4j page, where you can download and update it in your system.

Temporary mitigations (In case it's hard to upgrade the Log4j2 version immediately):

In previous releases (>2.10) this behavior can be mitigated by setting the system property **log4j2.formatMsgNoLookups** to true by adding the following Java parameter: **-Dlog4j2.formatMsgNoLookups=true.**

Alternatively, you can mitigate this vulnerability by removing the JndiLookup class from the classpath. Simply remove JndiLookup class from the classpath: “zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class”

You can also mitigate some of the exposure via your WAF, by adding rules that prevent the possible injection text. For example, see Cloudflare, Fastly, etc.

You can also track all of the statuses of the major cloud providers and their updates with their IaaS, PaaS and SaaS services that might be vulnerable: Google Cloud, AWS, Azure.

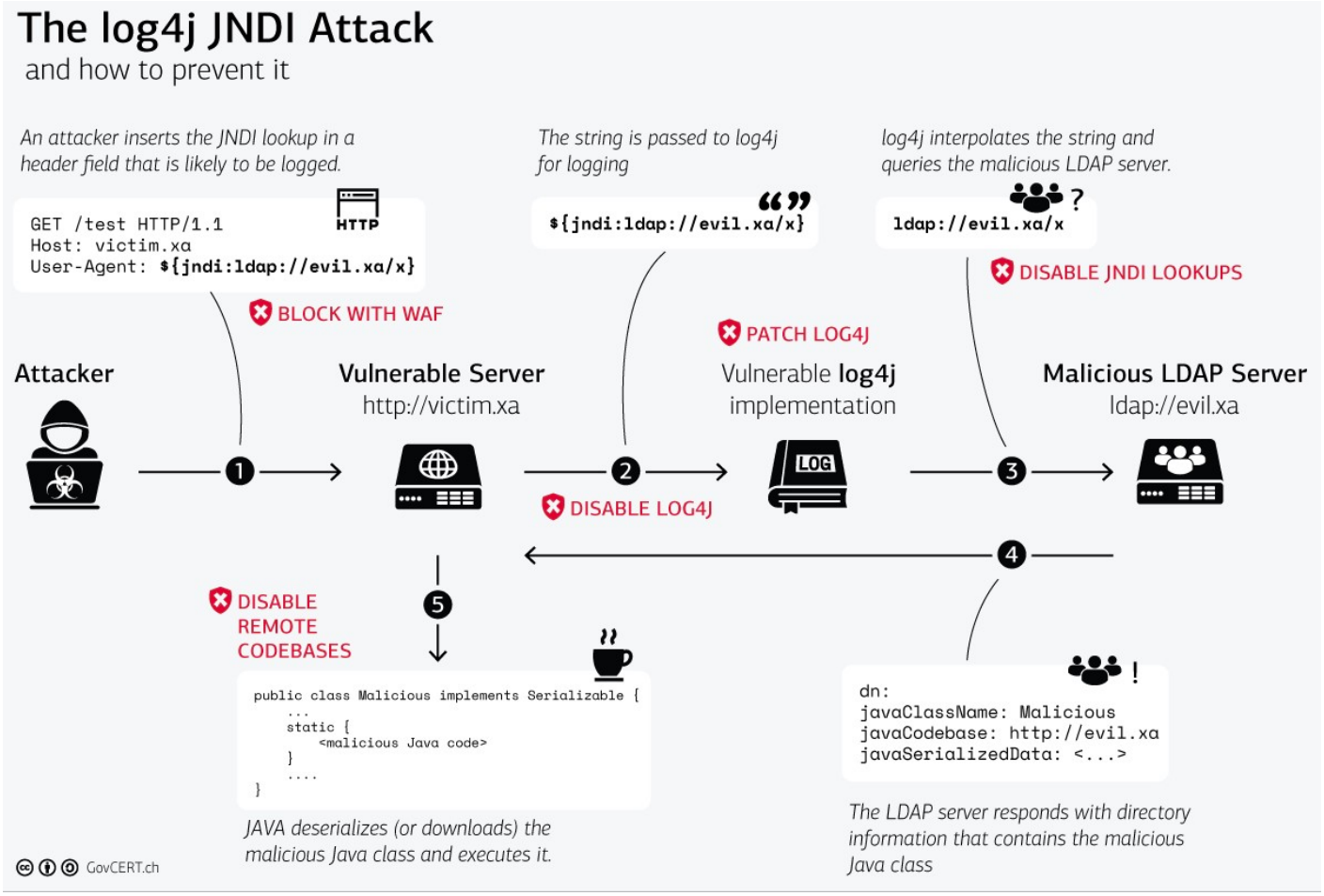
3. Respond

Panorays does not leverage or directly use a version of Log4j2 known to be affected by the vulnerability, and thus we do not currently believe the Panorays platform has been impacted. Nevertheless, for many, the challenge is patching the sheer number of programs using log4j2. These projects need updating or they remain vulnerable to exploitation.

Here are some things you can do:

1. Panorays’ The Third-Party Incident Response Playbook is available to help you respond to incidents like these with your third parties.
2. We’ve also created a Log4Shell exposure questionnaire, so you can gauge if the exploit affects you or your third parties. To receive the questionnaire, please contact support@panorays.com. We are happy to answer any questions you may have about this cyber vulnerability.

Schéma



Sources

https://panorays.com/blog/responding-to-the-log4shell-vulnerability/

https://www.rezilion.com/blog/making-sense-of-the-constantly-changing-log4shell-landscape/

Revision #1
Created 2026-07-17 15:09:51 UTC by Olivier
Updated 2026-07-17 15:09:53 UTC by Olivier