Apache Log4j Critical Vulnerability (CVE-2021-44228)

https://campus.barracuda.com/doc/96024338/

Description:

Log4j is a Java based logging audit framework within Apache. Apache Log4j <=2.14.1 JNDI features used in configuration, log messages, and parameters do not protect against attacker-controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. The vulnerability impacts default configurations of several Apache frameworks, including Apache Struts2, Apache Solr, Apache Druid, and Apache Flink, which are utilized by numerous organizations from Apple, Amazon, Cloudflare, Twitter, Steam, and others.

The vulnerability is triggered by sending a specific string to the log4j software which means it is simple to exploit and the broad utilization of this software means there are multiple attack vectors. Over the course of the last few days, we have seen attackers increasingly obfuscate their reconnaissance and exploit attempts for this vulnerability.

CVSS: 10 - Critical

CVE: CVE-2021-44228

Barracuda Networks Product Status:

Barracuda Load Balancer ADC Hardware & Virtual appliances and instances running in public cloud platforms such as AWS and Azure, Barracuda Load Balancer ADC does not use log4j, and hence it is not affected by this vulnerability.

Attack Detection and Protection:

Barracuda Load Balancer ADC

For Barracuda Load Balancer ADC Hardware and virtual appliances model 540 and higher, the latest signatures for this vulnerability are being rolled out to units in the field. These signatures and settings will block both GET and POST requests that are attempting this exploit. While these signatures detect variations that have been seen so far, we continue to update them as newer variants pop up. As a best practice, we recommend patching your log4j installations to the latest versions that have this issue fixed.

If you have automatic updates turned on and an active Energize Updates subscription:

1. Ensure that your unit(s) have updated the latest attackdef which has the remedy (1.208).

2. Navigate to **SECURITY > View Internal Patterns** and ensure that the new patterns are in **Active** mode
3. Navigate to **SECURITY > Security Policy > URL Protection** and **SECURITY > Security Policies > Parameter Protection** and ensure that OS-Command-Injection is turned **ON** for these policies. This setting is normally turned **ON**.
4. Under **SECURITY > Allow/Deny/Redirect > Header: Allow/Deny Rules**, create a new Header ACL with the **Header Name : Cookie**. Save this ADR, then select edit. In the edit screen, ensure that the ACL is turned **ON** and in **Active** mode, with OS-Command-Injection turned **ON**.
5. If you have created any profiles or Custom parameter classes under **SECURITY > Website Profiles**, ensure that OS-Command-Injection is turned **ON** for these settings.

Since this vulnerability is being exploited by using any available HTTP header, a header validation rule also needs to be added with the following details:

Header ACL Name: star-acl
Header Name: *
Status: On
Mode: Active
Max Header Value Length: Default to 512 -- but check if this requires a larger value for your application
Denied Metacharacters: < default-set >
Blocked Attack Types: Ensure that the OS Command Injection is selected