Apache Log4j Critical Vulnerabilities - December 2021

https://campus.barracuda.com/doc/96024300/

Update 23 December 2021

CVE-2021-4104 - Barracuda Networks has recently reviewed CVE-2021-4104. The *attackdef version 1.211* provides protection against Log4j vulnerability.

Update 20 December 2021 - Protections against Log4j Vulnerabilities

Since the original release of this article, Barracuda Networks has released updates to the attack patterns that can handle the newer vulnerabilities that have been released. The three attackdefs are described in the table below with details on the actual pattern names.

Vulnerabilities	Pattern	AttackDef Version	Release Date	Notes
CVE-2021-44228	log4j-rce-vulnerability	1.208	13 December 2021	First release
Various evasions and new attacks	log4j-rce-vulnerability	1.210	17 December 2021	Updated to attacks that do not use the closing "}"
CVE-2021-45105, CVE-2021-45046	log4j-rce-colon-vuln-strict log4j-rce-substition-strict	1.211	18 December 2021	Recursion & {ctx:) and other evasions. For enabling OS-Command-Injection-Strict, see the section later in this article.

Protecting against Log4j with Barracuda WAF & CloudGen WAF

The steps described earlier in this article for automated and manual enforcement are still valid. The only difference is in the association of various pattern groups and patterns.

If you wish to use the newly created strict patterns, ensure that you associate the OS-Command-Injection-Strict pattern group to the settings noted below. This pattern group contains several patterns that can cause false positives. If you have further questions on enabling the strict patterns, or need help enabling specific patterns from the group, contact Barracuda Networks Technical Support .

As always, to receive these updates, units must have an active Energize Updates subscription.

Templated Configuration of Log4j Protections across Services

If you have multiple services that require this configuration, consider using the template functionality to replicate configurations for ease of use. For more information about templates, see Templates Version 2 .

Protecting against Log4j with Barracuda WAF-as-a-Service

Protective configurations against Log4j for Barracuda WAF-as-a-Service are rolled out automatically to all applications. For CVE-2021-45105 and CVE-2021-45056, you are required to manually enable OS-Command-Injection-Strict if their applications require this setting. If you need help with enabling these patterns, contact Barracuda Networks Technical Support .

In general, if you find false positives due to these settings, and need help tuning the configurations for WAF, CloudGen WAF or WAF-as-a-Service, contact Barracuda Networks Technical Support .

Update: 16 December 2021 - Pattern to detect new evasion that does not use the closing “}”

Barracuda Networks recently observed that attacks are possible without using the closing curly bracket in the requests. The existing patterns will not detect such attempts. Detecting such attempts requires an aggressive pattern that could cause false positives. This pattern will be published as an "*attackdef*" in the coming days, along with the updates for any new evasions.

The updated pattern is available with our support team and can be manually applied. If you want to apply the updated pattern, contact Barracuda Networks Technical Support for the next steps.

Also, you can see the Proof of Concept for exfiltrating data from Log4j 2.15.0 here: https://www.praetorian.com/blog/log4j-2-15-0-stills-allows-for-exfiltration-of-sensitive-data/

The existing patterns released earlier this week can detect and block this exfiltration attempt.

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, such as Apple, Amazon, Cloudflare, Twitter, and Steam.

The vulnerability is triggered by sending a specific string to the Log4j software, which means it is simple to exploit. Furthermore, 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

For details, refer to the CVE-2021-44228 update from the National Vulnerability Database at NIST: https://nvd.nist.gov/vuln/detail/CVE-2021-44228.

Barracuda Networks Product Status

Barracuda Web Application Firewall hardware and virtual appliances; Barracuda CloudGen WAF on AWS, Azure, and GCP; Barracuda WAF-as-a-Service; and Barracuda LoadBalancer ADC do not use log4j, and are therefore not affected by this vulnerability.

Attack Detection and Protection

Barracuda WAF-as-a-Service

We are rolling out new signatures to detect and block the Log4j exploit attempts. These signatures have been updated to handle the latest evasions seen in the field as of 13 December 2021. These signatures and settings will block both GET and POST requests that are attempting this exploit.

Barracuda Web Application Firewall & Barracuda CloudGen WAF

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. Although these signatures detect variations that have been seen so far, we continue to update them as newer variants pop up. As a best practice, Barracuda Networks recommend patching your Log4j installations to the latest versions that have this issue fixed.

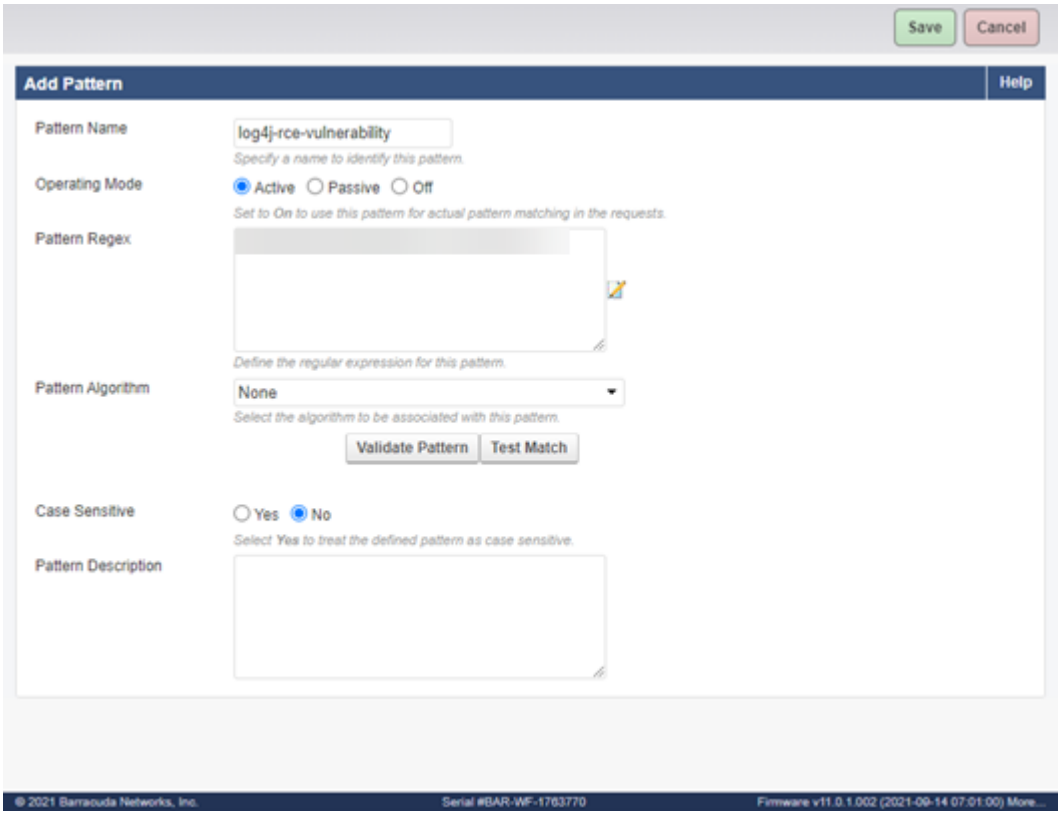
If you have Automatic Updates turned On and an Active Energize Updates Subscription

1. Ensure that your unit(s) have updated the latest attackdef that has the remedy (1.208).
2. Go to **ADVANCED > View Internal Patterns** and ensure that the new patterns are in the **Active** mode.
3. Go to **SECURITY POLICIES > URL Protection** and **SECURITY POLICIES > Parameter Protection,** and ensure that *Only* **OS-Command-Injection** is turned **ON** for these policies. This setting is normally turned **ON**.
4. Under **WEBSITES > Allow/Deny > Redirect > Header-ADRs**, edit the star-acl and turn it **ON** and ensure that **OS-Command-Injection** is **ON**. Ensure that the **ACL** is turned **ON** and in **Active Mode**.
5. Under **WEBSITES > Allow/Deny/Redirect > Header: Allow/Deny Rules**, create a new Header ACL with the **Header Name: Cookie**. Save the ADR, and then select **Edit**. In the edit screen, ensure that the **ACL** is turned **ON** and in **Active mode** with **OS-Command-Injection** turned **ON**.
6. If you have created any profiles or custom parameter classes under **WEBSITES > Website Profiles**, ensure that *Only* **OS-Command-Injection** is turned **ON** for these settings.

Since this vulnerability is being exploited by using any available HTTP header, turning on the star-acl is critical to ensure that all headers are parsed and validated for the attacks.

If you are in Offline Mode and cannot update to the latest Attack Definitions

1. Go to **ADVANCED > Libraries > Attack Types** and create a new pattern group (Example: log4j-rce-vulnerability). Create a new pattern with the required regex match.
   For security purpose, the list of patterns is not publicly published. Contact Barracuda Networks Technical Support for the list of patterns to add.
   Make sure that **Case Sensitive** is set to **No**.



2. Go to **SECURITY POLICIES > URL Protection** and **SECURITY POLICIES > Parameter Protection** and turn **ON** this new group for the required security policies.
3. Under **WEBSITES > Allow/Deny > Redirect > Header-ADRs**, edit the star-acl and turn it **ON** and ensure the new pattern group is **ON**. Ensure that the **ACL** is turned **ON** and in **Active Mode**.
4. Under **WEBSITES > Allow/Deny/Redirect > Header: Allow/Deny Rules**, create a new header ACL with the **Header Name: Cookie**. Save the ADR, and then select **Edit**. In the edit screen, ensure that the **ACL** is turned **ON** and in **Active mode**, the new pattern group is turned **ON**.
5. If you have created any profiles or custom parameter classes under **WEBSITES > Website Profiles**, ensure that the new pattern group is turned **ON** for these settings.

For any assistance with these settings or questions regarding the attack patterns, contact Barracuda Networks Technical Support.

Figures