DC-2 VULNHUB WALKTHROUGH: A STEP BY STEP GUIDE

By: Alina Prem

This walkthrough will give an idea of each step of the DC-2 machine exploitation hosted on VulnHub. The steps include everything from the initial network scanning , enumeration, exploitation ,privilege escalation and collection of the flags.

Step 1: Discovering the target machines IP

Objective:

To identify the ip of the target machine on the local machine.

Action:

I run the command **sudo arp-scan --localnet** to scan my local network.

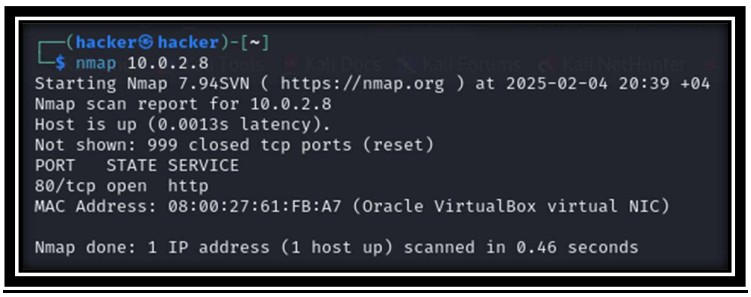
Step 2: Performing basic nmap scan

Objective:

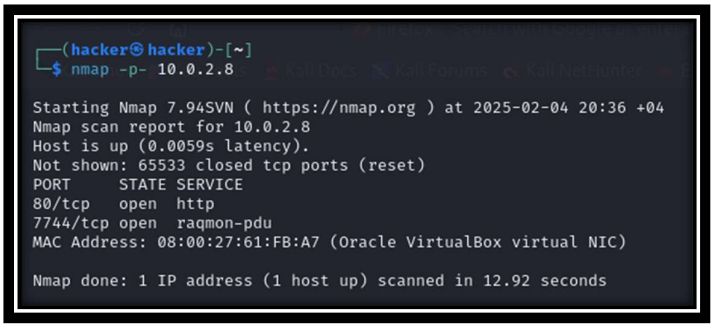
To find out the open ports on the target machine

Action :

Before jumping in it is always good to do a basic scan inorder to know what we are dealing with and for that I did a simple scan using nmap : **nmap 10.0.2.8** however it shows only port.



To gather all the information that I need I run another scan using **nmap -p- 10.0.2.8** by doing so it scanned through all the ports. There were 2 open ports available.



Step 3:Extensive nmap scan

Objective:

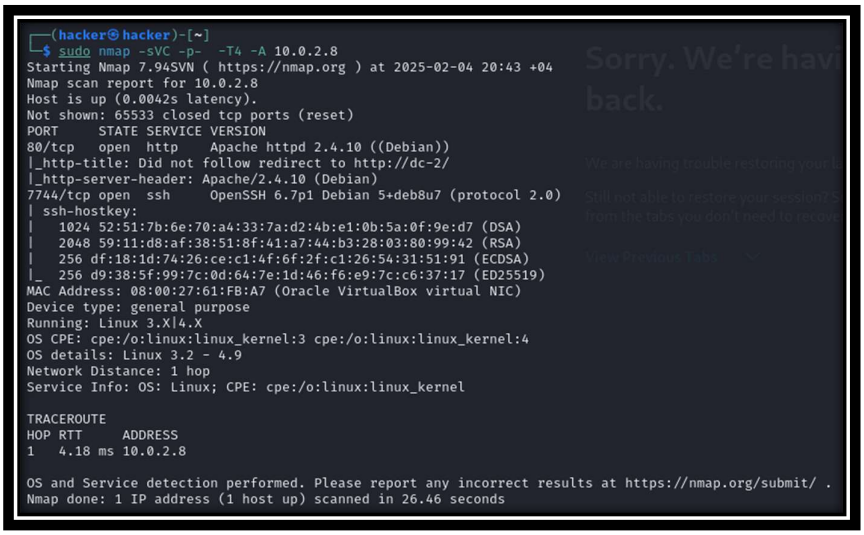
To gather a detailed analysis of the target machines open ports, the services , their versions and any potential vulnerabilities .

Action:

After conducting the basic scan I run a more extensive scan using the command **sudo nmap -svc -p- -T4 -A 10.0.2.8**

Result:

- The target was running WordPress on port 80.
- The scan also provided details about the version of WordPress and the server software running.



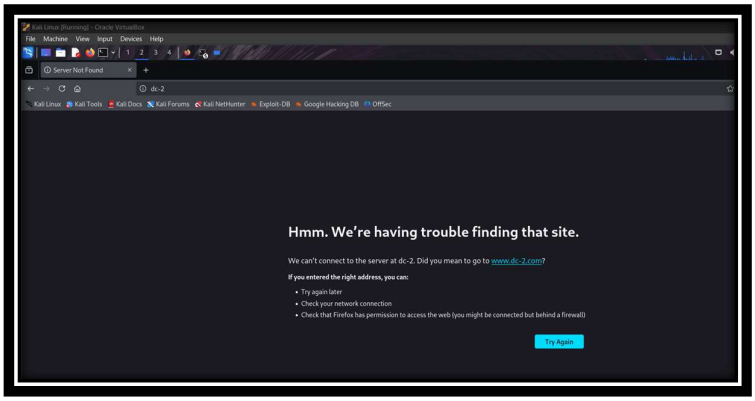
Step 4: interacting with the web service

Objective:

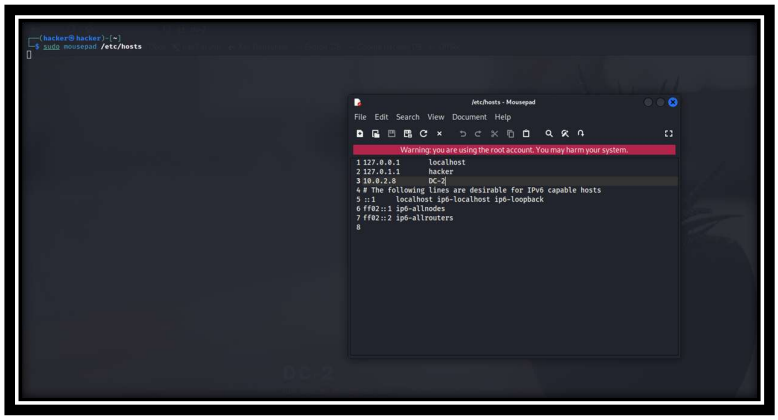
To assess the web service running on port 80

Action:

The next thing I do is head to port 80. I open the browser and type in the ip to interact with web service provided but it couldn't connect.



To troubleshoot the issue I modified the /etc/hosts file using **sudo mousepad /etc/hosts**. Inside the file I add the ip address and the hostname of DC-2 to make sure the system can resolve it correctly.



Then I again tried the Ip and this type it showed me a WordPress site with some basic informations . I inspect around a little to see if I can find any clues .

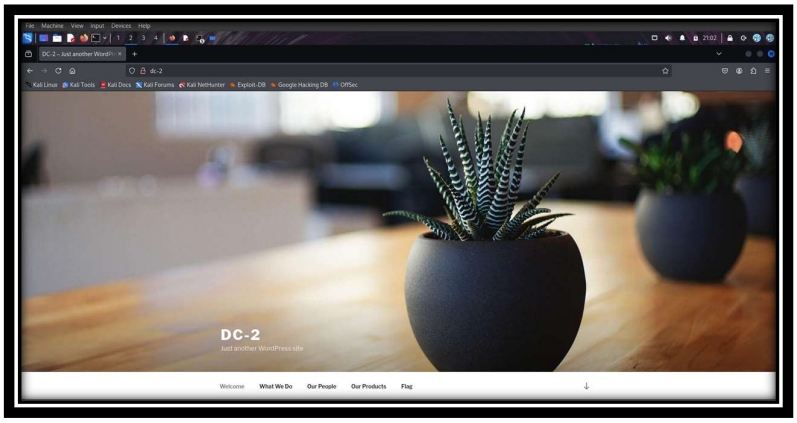
Step 5: inspecting the wordpress site

Objective:

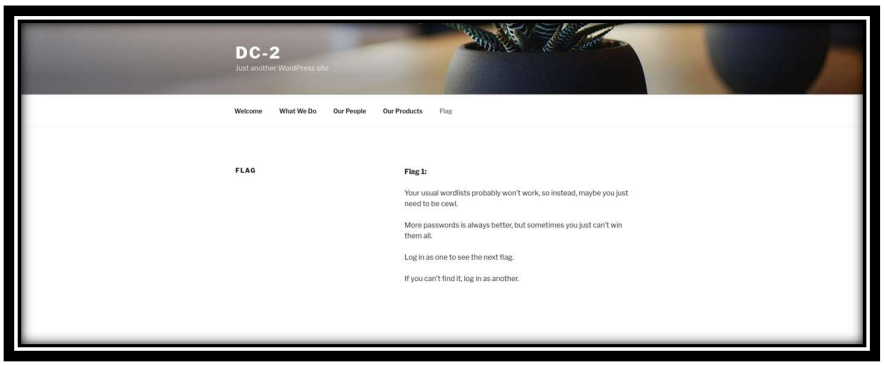
To look for flags , clues or any other information's available in the website.

Action:

Once I accessed the WordPress site my next aim was to look for any possible flags of hints that maybe available in the page source directories etc.



While browsing the site I found **flag 1** in the page contents with some clues that are required to move forward.



Step 6: Finding hidden directories using gobuster

Objective:

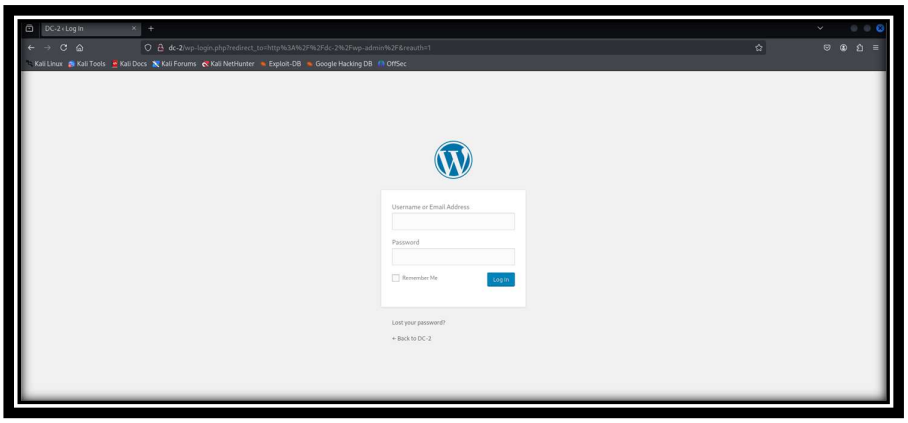
To use a tool to find the hidden directories that are available for the wordpress site.

Action:

I do gobuster search using command **gobuster dir -u http://10.0.2.8 -w /usr/share/wordlists/dirb/common.txt.**

```
┌──(hacker㉿hacker)-[~]
└─$ gobuster dir -u http://10.0.2.8 -w /usr/share/wordlists/dirb/common.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.0.2.8
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.hta                (Status: 403) [Size: 287]
/.htpasswd           (Status: 403) [Size: 292]
/.htaccess           (Status: 403) [Size: 292]
/index.php           (Status: 200) [Size: 53562]
/server-status       (Status: 403) [Size: 296]
/wp-admin            (Status: 301) [Size: 307] [--> http://10.0.2.8/wp-admin/]
/wp-content          (Status: 301) [Size: 309] [--> http://10.0.2.8/wp-content/]
/wp-includes         (Status: 301) [Size: 310] [--> http://10.0.2.8/wp-includes/]
Progress: 4614 / 4615 (99.98%)
/xmlrpc.php          (Status: 405) [Size: 42]
===============================================================
Finished
===============================================================
```

During the scan, Gobuster discovered a hidden directory that led to a **WordPress login page**.



Step 7: creating a customized wordlist using cewl

Objective:

To generate a customized wordlist based on the websites content.

Action:

Based on the hints that were given in the flag 1 I use cewl .I run the command **cewl -w DC-2.TXT http://dc-2/**

```
┌──(hacker㉿hacker)-[~]
└─$ cewl -w DC-2.TXT http://dc-2/
CeWL 6.2.1 (More Fixes) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
```

Step 8: Enumerating users using WPScan

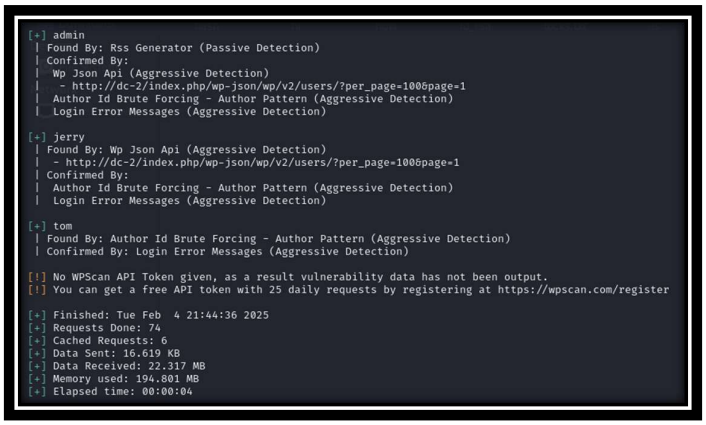
Objective:

Enumerate valid users of the wordpress site to gain access to the login page.

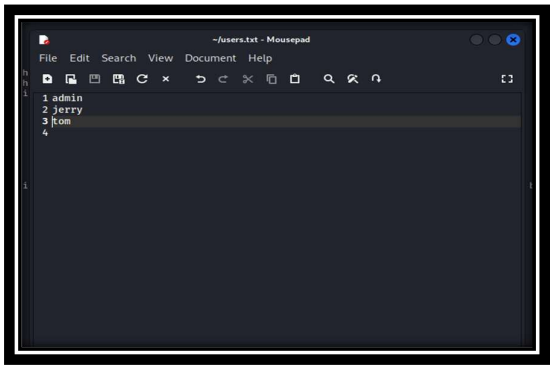
Action:

I enumerate the users available using command **wpscan –url http://dc-2 –enumerate u**

I got 2 usernames jerry and tom and admin is the default user in the wordpress site .



I save the users to a file users.txt



Step 9: brute forcing the wordpress login using WPScan

Objective:

Attempt a brute force attack to find the correct passwords for the login.

Action:

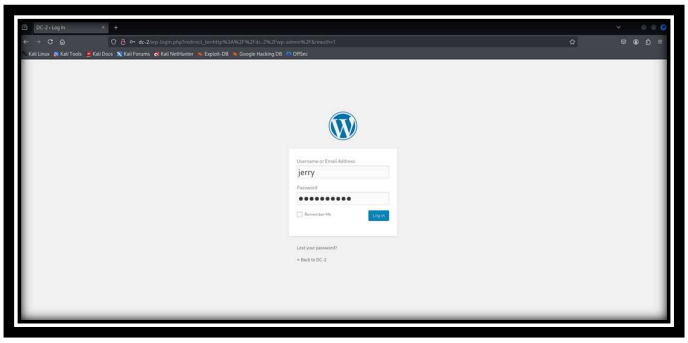
Now I have the list of usernames users.txt and passwords in wordlist DC-2.TXT . so next I try to bruteforce the target machine using the wpscan using command **wpscan –url http:dc-2 -U /home/hacker/users.txt -P /home/hacker/DC-2.TXT**

The wpscan is successful and I get the paswords of tom and jerry user.

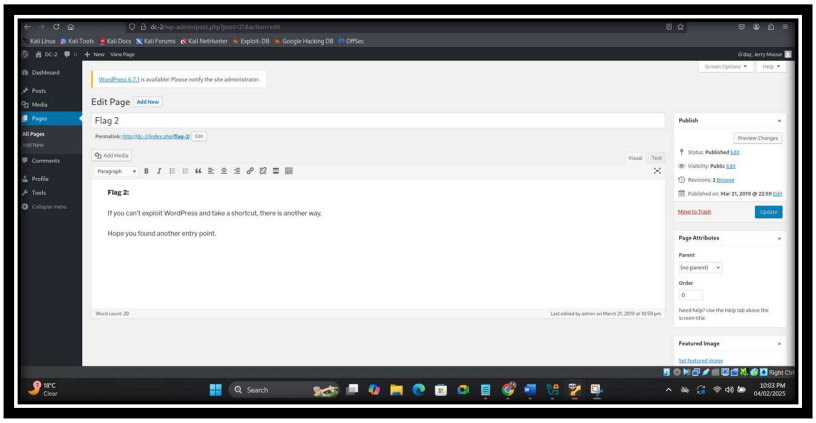
Step 10: logging in to jerry's WordPress account

Objective:

Using the credentials trying to gain access



As I inspect around a little I find flag 2 with further hints.



Step 11: gaining ssh access into the machine

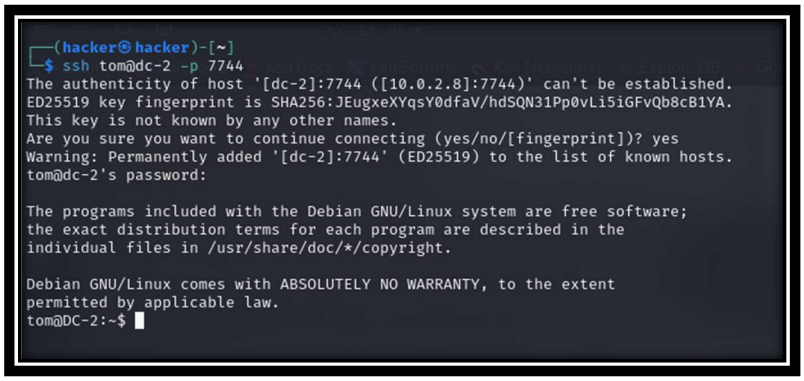
Objective:

To gain access into the system using the previously gained credentials.

Action:

Now as I have a username and password I try to connect to ssh using the command **ssh tom@dc-2 -p 7744**

I get access.



I try listing the contents but the shell was rbash as it was a restricted shell I couldn't gather any other information.

Step 12: escaping the rbash using Vi

Objective:

To escape from the restricted shell and to gain full access into the machine

Action:

After researching around a little I found that I could use vi editor to gain full access .

Inside the vi editor I typed **:set shell=/bin/sh** and then **:shell**

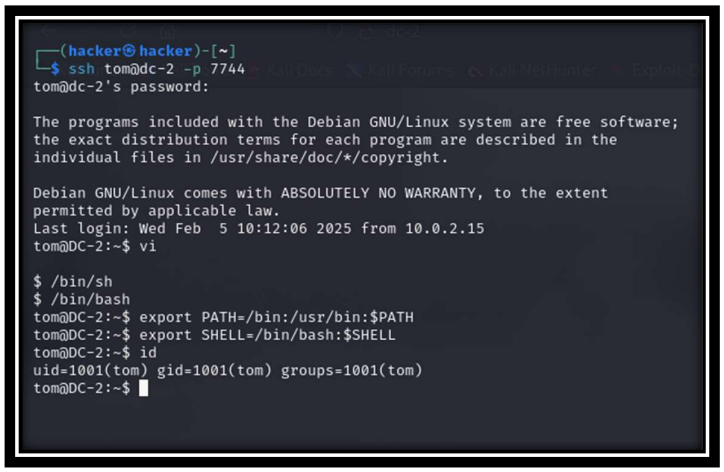
After that, I typed the /bin/bash command to switch to the Bash shell

Then I added the missing directories using the command

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

export SHELL=/bin/bash:$SHELL

id



Step 14: to display flag 3

Objective:

To find and display the third flag

Action:

I ,listed the contents and found flag3.txt and I opened it using cat command.

Step 15: switching to jerry's account

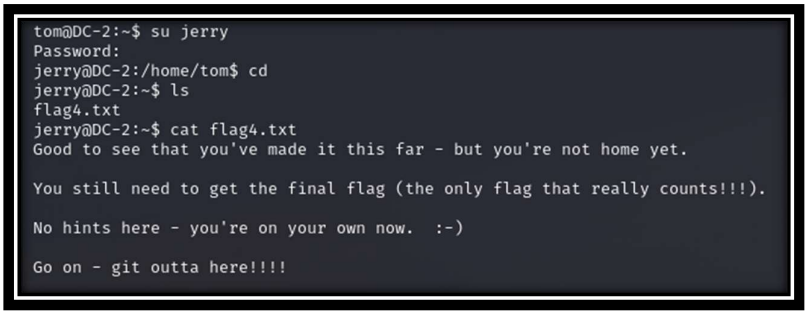
Objective:

To find if there are any flags present in jerrys account.

Action:

I switched to jerrys account using the same credentials that I got when I did WPScan.

Flag 4.txt was in the home directory.



Step 16: Finding sudo privileges and privilege escalation

I find there is a hint here and next to check the permissions of the user I try to find if there are any sudo privileges using **sudo -l** and it shows user can run the git command without the root password.

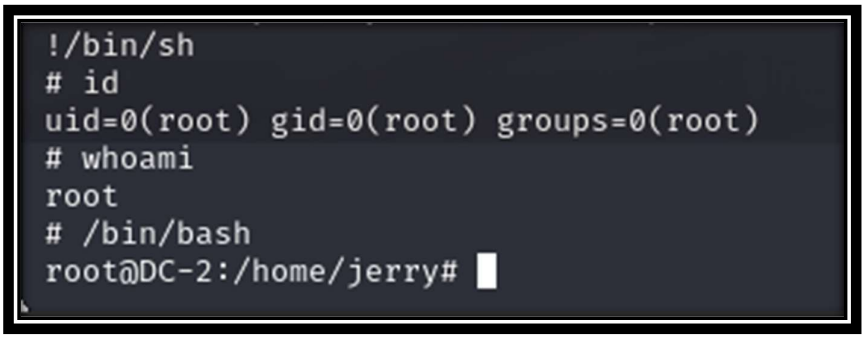
I search git in GTFOBins and find a code I execute it

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

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

I execute the first code **sudo git -p help config**

Then the next line **!/bin/sh**



Step 17 :Finally I printed out the final flag .

THE END !!!