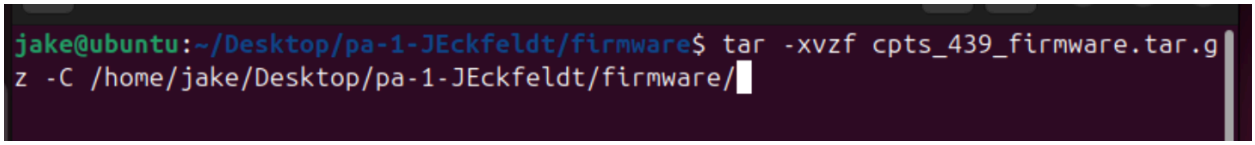
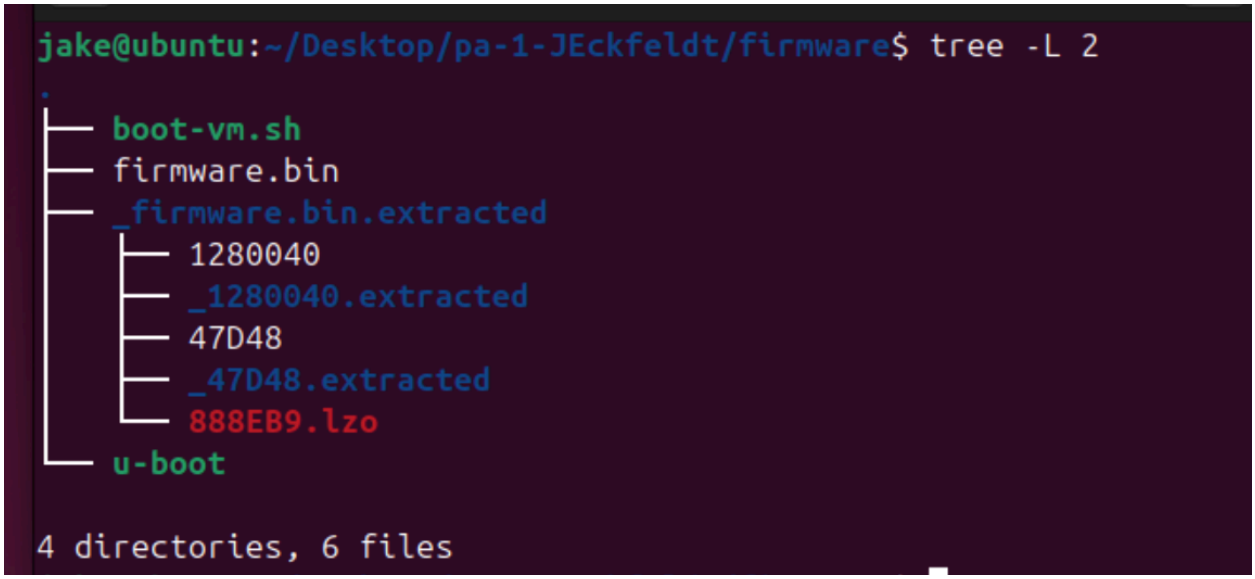
Jake Eckfeldt
PA 1

Task 1:

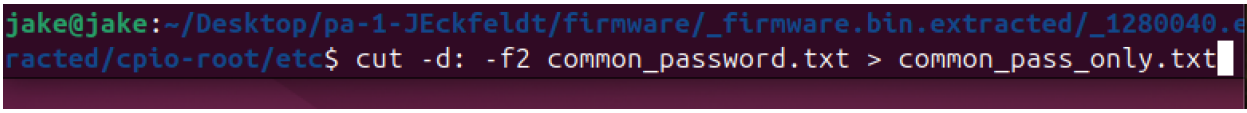
1) I used this command to unzip the firmware file and put its contents into the shown directory. It was easier to work with putting the contents in their own folder



2) Here is the tree command on the extracted firmware.tar.gz file. I ran this after extracting the binary file so you see the extracted version as well (I limited the tree depth so you can actually see what's going on)



3) Some things I noticed analyzing the binary file:
    a) Uimage header, which implies that the firmware uses u-boot
    b) The modify dates on gzip data is null maybe to save space or privacy
    c) Gzip can be unzipped and seen in its raw form
    d) I see 2 CR32 Polynomial tables that could be used for partitions or maybe checksum
    e) 2 uimage headers implies a partition so probably what at least 1 CR32 table is for
4) Hashed pwd for root:
5tteRhk9Y$7u0nhaiBjVi9TNXvMrGk/Nh7lDSeZfV7bROSOJFVbN6
5) To crack the password:
    a) I grabbed the second field of the common_password.txt with the command



b) I had to figure out what type of hash was used on the root password hash with hashid using this command

```
jake@jake:~/Desktop/pa-1-JEckfeldt/firmware/_firmware.bin.extracted/_1280040.e
racted/cpio-root/etc$ hashid -m $(cat root.hash)
```

c) I got the code 7400 to use with hashcat and ran this command to crack the pwd

```
jake@jake:~/Desktop/pa-1-JEckfeldt/firmware/_firmware.bin.extracted/_1280040.
racted/cpio-root/etc$ hashcat -m 7400 root.hash common_pass_only.txt --force
```

d) I ran the same command with "--show" and got this result

```
jake@jake:~/Desktop/pa-1-JEckfeldt/firmware/_firmware.bin.extracted/_1280040.e
racted/cpio-root/etc$ hashcat -m 7400 root.hash common_pass_only.txt --show
$5$tteRhk9Y$7u0nhaiBjVi9TNXvMrGk/Nh7lDSeZfV7bROSOJFVbN6:realtek
```

6) Was able to login to root with password: realtek

```
# cat flag
cat: can't open 'flag': No such file or directory
# cat flag.txt
\f1"b
*D5#e-:]%Q# cd ..
# cd ..
# exit

IOTSEC101-CTF [made by CJHackerz]
target login: root
Password:
#
```

Task 2:

1) The file vault value

```
target login: root
Password:
# ls
flag.txt
# cat /proc/cmdline
root=/dev/ram console=ttyAMA0,38400n8 FILE_VAULT=NOeHAVYuo6KRbtWxxk4ixkUM9maqs670fV7g6p7
#
```

2) The values from the hex command on the file vault hash (saved to local machine)

```
jake@jake:~/Desktop/pa-1-JEckfeldt$ nano flag.hex
jake@jake:~/Desktop/pa-1-JEckfeldt$ cat flag.hex
1d 1a 5c 1d 14 66 0f 31 22 62 0a 2a 07 44 1e 0f
35 23 65 10 2d 01 14 3a 5d 25 51
```

3) After decrypting the hex value and saving it on the local machine

```
jake@jake:~/Desktop/pa-1-JEckfeldt$ xxd -r -p flag.hex flag.txt
jake@jake:~/Desktop/pa-1-JEckfeldt$ cat flag.txt
\f1"b
*D5#e-:]%Qjake@jake:~/Desktop/pa-1-JEckfeldt$
```

4) This is the result from the XOR output using the FILE_VAULT value as the key and the raw flag.txt as the data

```
jake@jake:~/Desktop/pa-1-JEckfeldt$ python3 xor.py "$(cat vault_key.txt)" < flag.txt
SU9UU0VDMTAxe0IwMHQyUjAwdH0jake@jake:~/Desktop/pa-1-JEckfeldt$
```

5) The Decoded XOR output result from step 6

```
jake@jake:~/Desktop/pa-1-JEckfeldt$ python3 xor.py "$(cat vault_key.txt)" < flag
.txt
SU9UU0VDMTAxe0IwMHQyUjAwdH0jake@jake:~/Desktop/pa-1-JEckfeldt$ python3 base64_de
code.py
Decoded string: IOTSEC101{B00t2R00t}
jake@jake:~/Desktop/pa-1-JEckfeldt$
```