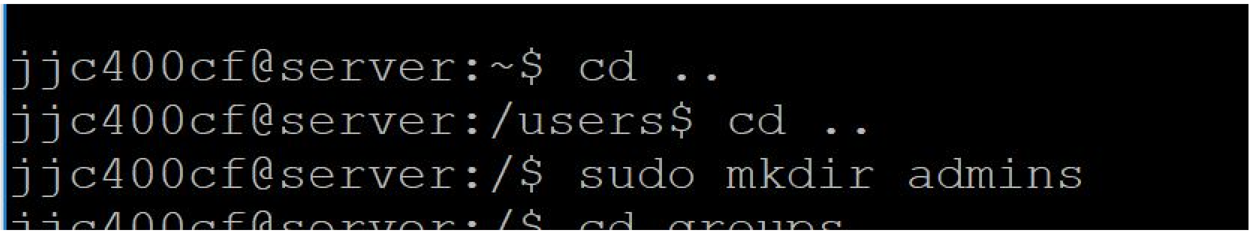
Lab4

4.1 The FrobozzCo Permissions Test

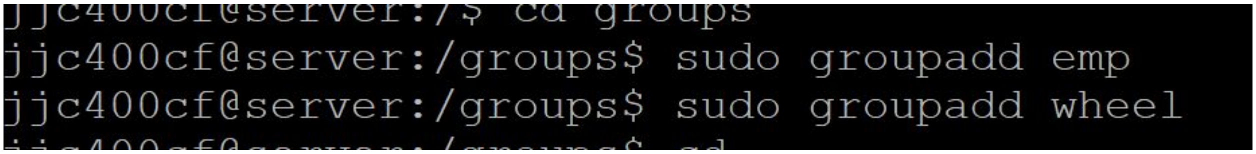
4.1.2 Home Directory Security

1.



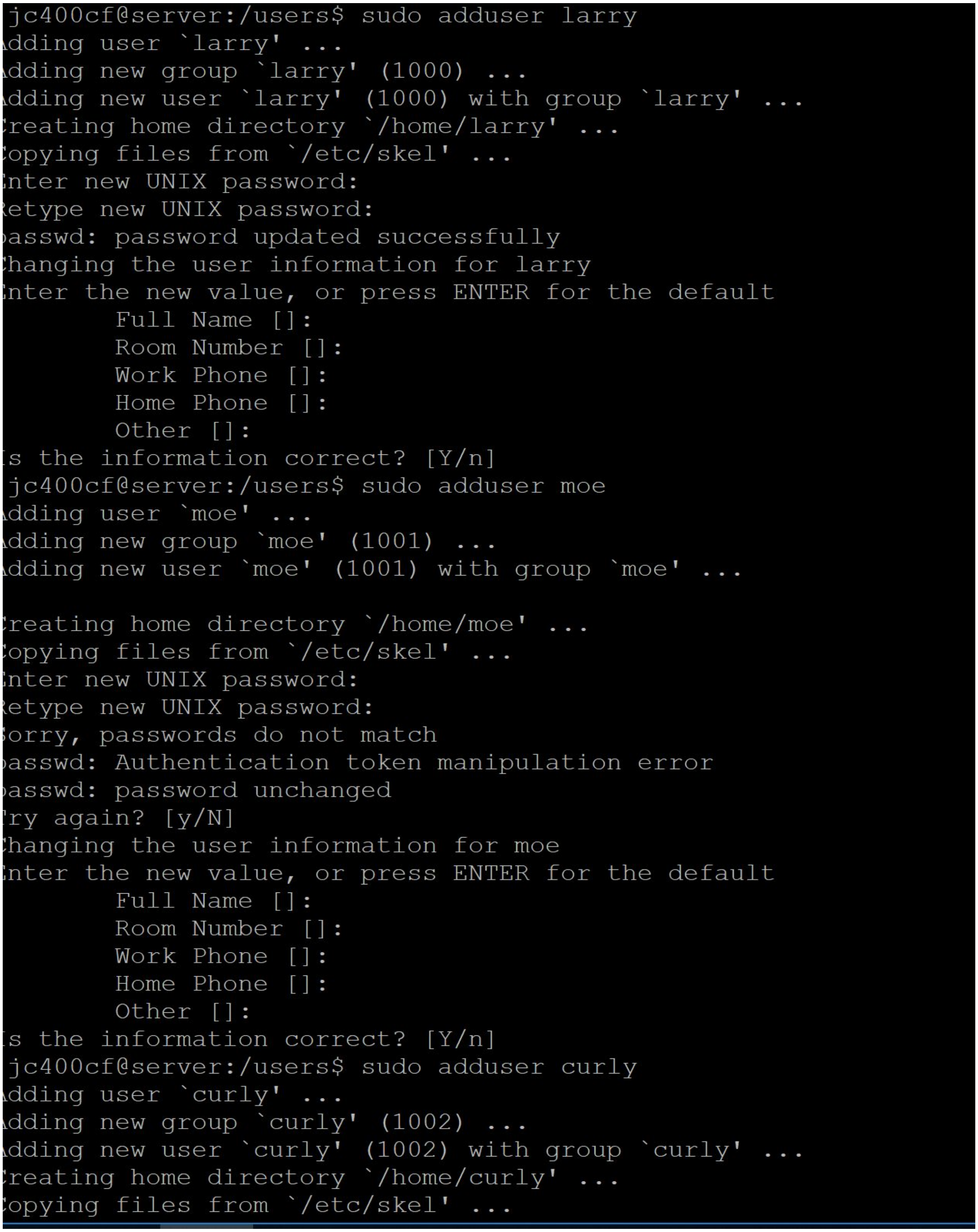
```
jjc400cf@server:~$ cd ..
jjc400cf@server:/users$ cd ..
jjc400cf@server:/$ sudo mkdir admins
jjc400cf@server:/$ cd groups
```

2.



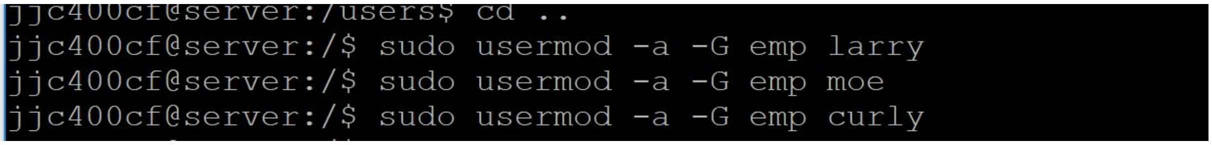
```
jjc400cf@server:/$ cd groups
jjc400cf@server:/groups$ sudo groupadd emp
jjc400cf@server:/groups$ sudo groupadd wheel
jjc400cf@server:/groups$ cd
```

3.



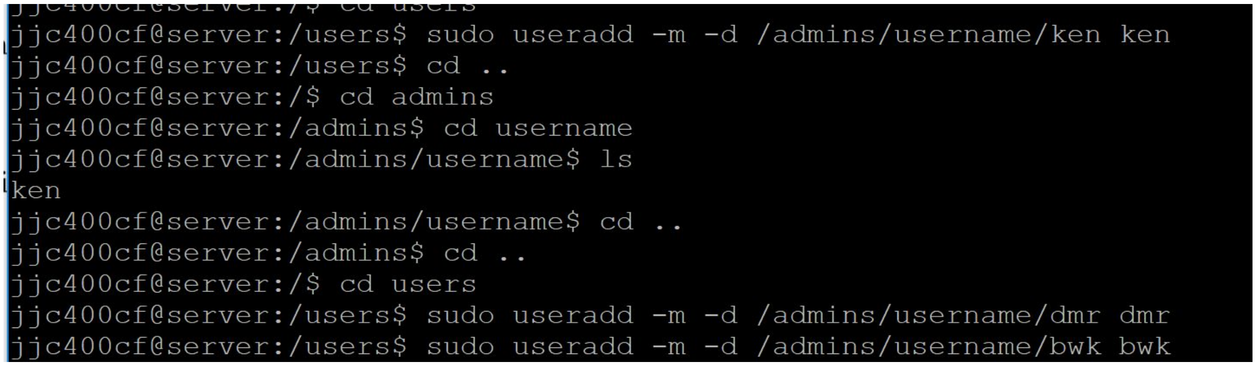
```
 jc400cf@server:/users$ sudo adduser larry
dding user `larry' ...
dding new group `larry' (1000) ...
dding new user `larry' (1000) with group `larry' ...
reating home directory `/home/larry' ...
opying files from `/etc/skel' ...
nter new UNIX password:
etype new UNIX password:
asswd: password updated successfully
hanging the user information for larry
nter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
 jc400cf@server:/users$ sudo adduser moe
dding user `moe' ...
dding new group `moe' (1001) ...
dding new user `moe' (1001) with group `moe' ...

reating home directory `/home/moe' ...
opying files from `/etc/skel' ...
nter new UNIX password:
etype new UNIX password:
orry, passwords do not match
asswd: Authentication token manipulation error
asswd: password unchanged
ry again? [y/N]
hanging the user information for moe
nter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
 jc400cf@server:/users$ sudo adduser curly
dding user `curly' ...
dding new group `curly' (1002) ...
dding new user `curly' (1002) with group `curly' ...
reating home directory `/home/curly' ...
opying files from `/etc/skel' ...
```

4.



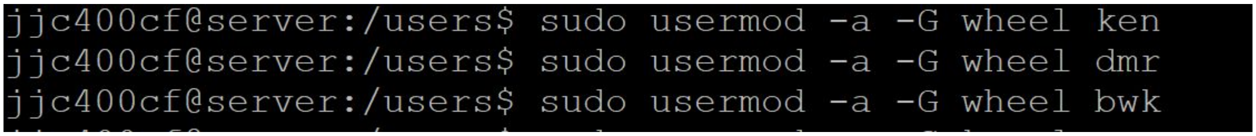
```
jjc400cf@server:/users$ cd ..
jjc400cf@server:/$ sudo usermod -a -G emp larry
jjc400cf@server:/$ sudo usermod -a -G emp moe
jjc400cf@server:/$ sudo usermod -a -G emp curly
```

5.



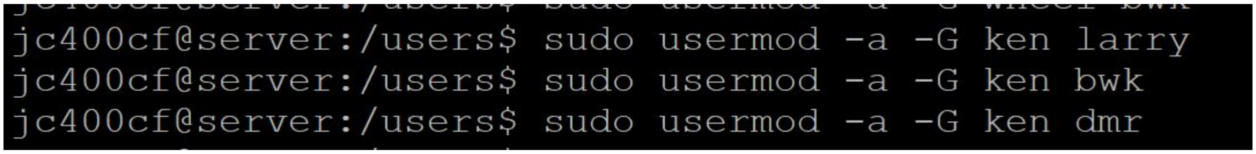
```
jjc400cf@server:/$ cd users
jjc400cf@server:/users$ sudo useradd -m -d /admins/username/ken ken
jjc400cf@server:/users$ cd ..
jjc400cf@server:/$ cd admins
jjc400cf@server:/admins$ cd username
jjc400cf@server:/admins/username$ ls
ken
jjc400cf@server:/admins/username$ cd ..
jjc400cf@server:/admins$ cd ..
jjc400cf@server:/$ cd users
jjc400cf@server:/users$ sudo useradd -m -d /admins/username/dmr dmr
jjc400cf@server:/users$ sudo useradd -m -d /admins/username/bwk bwk
```

6.



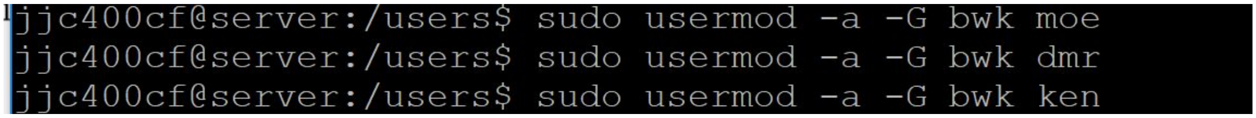
```
jjc400cf@server:/users$ sudo usermod -a -G wheel ken
jjc400cf@server:/users$ sudo usermod -a -G wheel dmr
jjc400cf@server:/users$ sudo usermod -a -G wheel bwk
```

7.



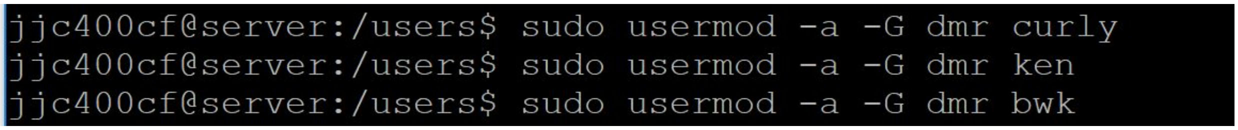
```
jc400cf@server:/users$ sudo usermod -a -G wheel bwk
jc400cf@server:/users$ sudo usermod -a -G ken larry
jc400cf@server:/users$ sudo usermod -a -G ken bwk
jc400cf@server:/users$ sudo usermod -a -G ken dmr
```

8.



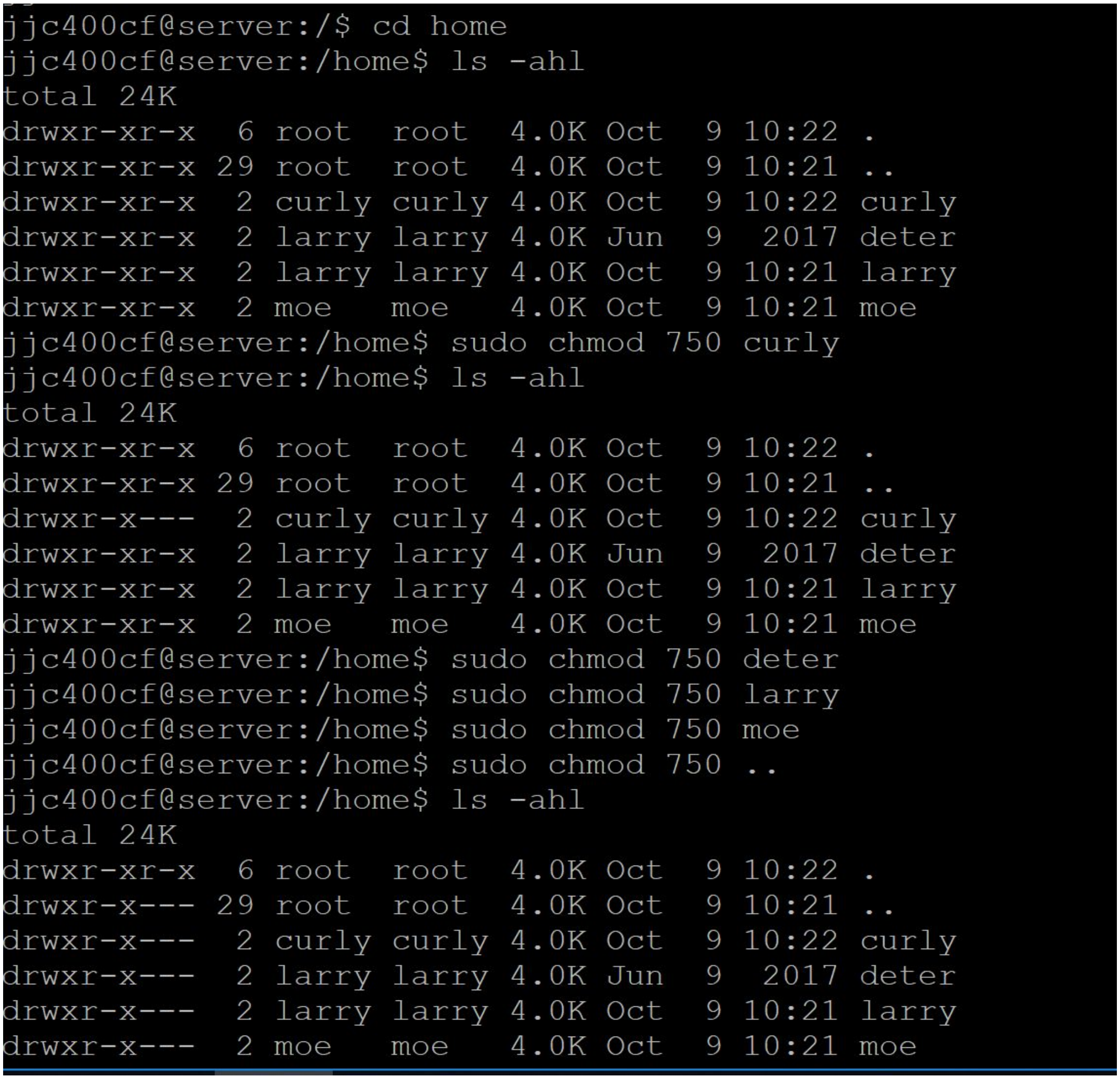
```
jjc400cf@server:/users$ sudo usermod -a -G bwk moe
jjc400cf@server:/users$ sudo usermod -a -G bwk dmr
jjc400cf@server:/users$ sudo usermod -a -G bwk ken
```

9.



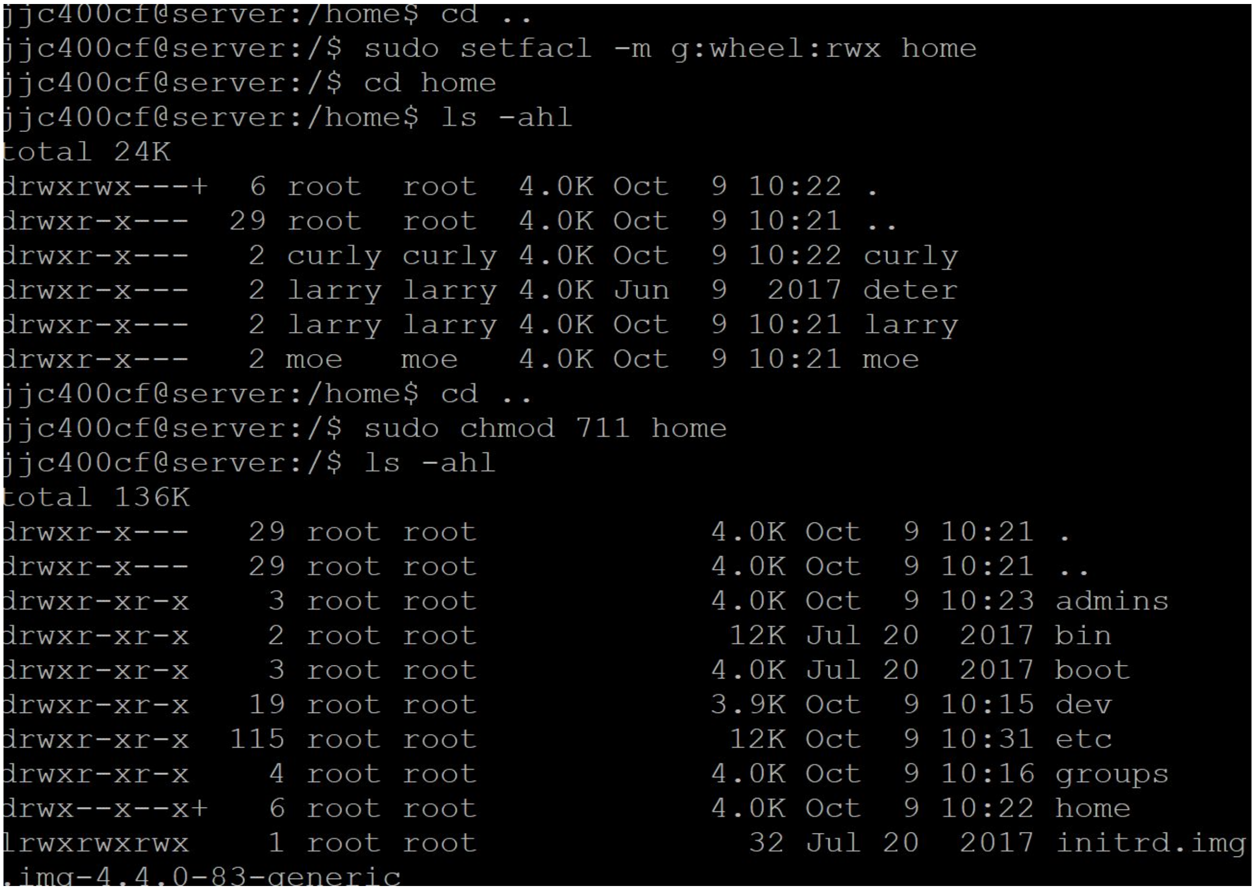
```
jjc400cf@server:/users$ sudo usermod -a -G dmr curly
jjc400cf@server:/users$ sudo usermod -a -G dmr ken
jjc400cf@server:/users$ sudo usermod -a -G dmr bwk
```

10.



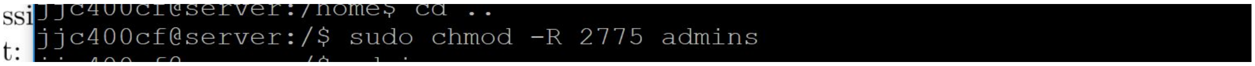
```
jjc400cf@server:/$ cd home
jjc400cf@server:/home$ ls -ahl
total 24K
drwxr-xr-x  6 root  root  4.0K Oct  9 10:22 .
drwxr-xr-x 29 root  root  4.0K Oct  9 10:21 ..
drwxr-xr-x  2 curly curly 4.0K Oct  9 10:22 curly
drwxr-xr-x  2 larry larry 4.0K Jun  9  2017 deter
drwxr-xr-x  2 larry larry 4.0K Oct  9 10:21 larry
drwxr-xr-x  2 moe   moe   4.0K Oct  9 10:21 moe
jjc400cf@server:/home$ sudo chmod 750 curly
jjc400cf@server:/home$ ls -ahl
total 24K
drwxr-xr-x  6 root  root  4.0K Oct  9 10:22 .
drwxr-xr-x 29 root  root  4.0K Oct  9 10:21 ..
drwxr-x---  2 curly curly 4.0K Oct  9 10:22 curly
drwxr-xr-x  2 larry larry 4.0K Jun  9  2017 deter
drwxr-xr-x  2 larry larry 4.0K Oct  9 10:21 larry
drwxr-xr-x  2 moe   moe   4.0K Oct  9 10:21 moe
jjc400cf@server:/home$ sudo chmod 750 deter
jjc400cf@server:/home$ sudo chmod 750 larry
jjc400cf@server:/home$ sudo chmod 750 moe
jjc400cf@server:/home$ sudo chmod 750 ..
jjc400cf@server:/home$ ls -ahl
total 24K
drwxr-xr-x  6 root  root  4.0K Oct  9 10:22 .
drwxr-x--- 29 root  root  4.0K Oct  9 10:21 ..
drwxr-x---  2 curly curly 4.0K Oct  9 10:22 curly
drwxr-x---  2 larry larry 4.0K Jun  9  2017 deter
drwxr-x---  2 larry larry 4.0K Oct  9 10:21 larry
drwxr-x---  2 moe   moe   4.0K Oct  9 10:21 moe
```

11.

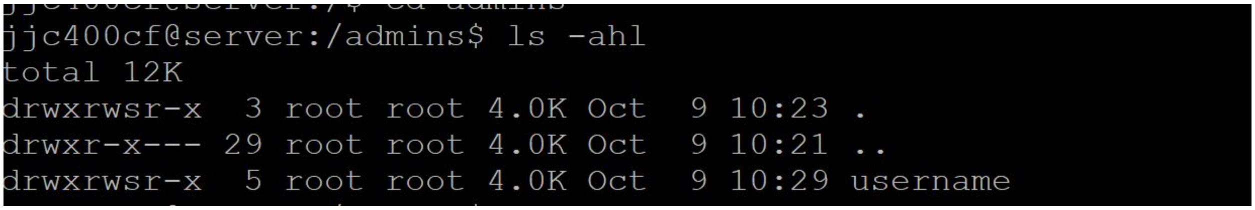


```
jjc400cf@server:/home$ cd ..
jjc400cf@server:/$ sudo setfacl -m g:wheel:rwx home
jjc400cf@server:/$ cd home
jjc400cf@server:/home$ ls -ahl
total 24K
drwxrwx---+  6 root  root  4.0K Oct  9 10:22 .
drwxr-x---  29 root  root  4.0K Oct  9 10:21 ..
drwxr-x---   2 curly curly 4.0K Oct  9 10:22 curly
drwxr-x---   2 larry larry 4.0K Jun  9  2017 deter
drwxr-x---   2 larry larry 4.0K Oct  9 10:21 larry
drwxr-x---   2 moe   moe   4.0K Oct  9 10:21 moe
jjc400cf@server:/home$ cd ..
jjc400cf@server:/$ sudo chmod 711 home
jjc400cf@server:/$ ls -ahl
total 136K
drwxr-x---   29 root root            4.0K Oct  9 10:21 .
drwxr-x---   29 root root            4.0K Oct  9 10:21 ..
drwxr-xr-x    3 root root            4.0K Oct  9 10:23 admins
drwxr-xr-x    2 root root             12K Jul 20  2017 bin
drwxr-xr-x    3 root root            4.0K Jul 20  2017 boot
drwxr-xr-x   19 root root            3.9K Oct  9 10:15 dev
drwxr-xr-x  115 root root             12K Oct  9 10:31 etc
drwxr-xr-x    4 root root            4.0K Oct  9 10:16 groups
drwx--x--x+   6 root root            4.0K Oct  9 10:22 home
lrwxrwxrwx    1 root root              32 Jul 20  2017 initrd.img
.img-4.4.0-83-generic
```

12.



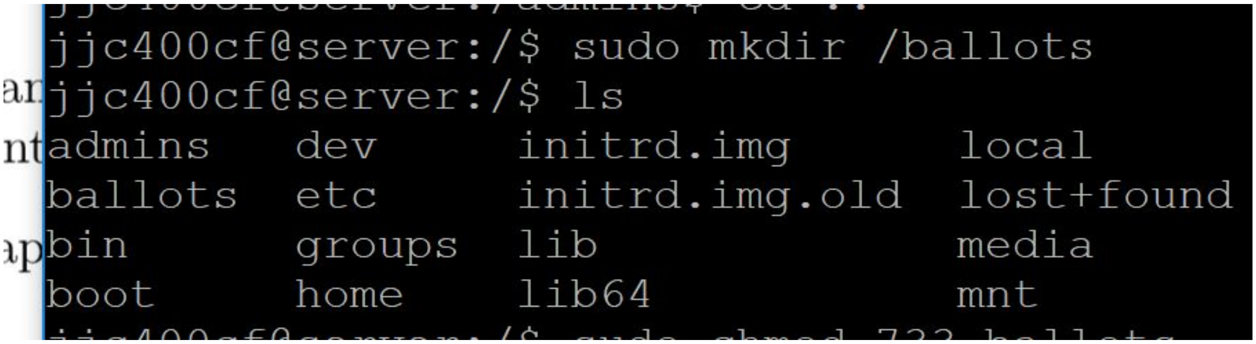
```
jjc400cf@server:/home$ cd ..
jjc400cf@server:/$ sudo chmod -R 2775 admins
```



```
jjc400cf@server:/$ cd admins
jjc400cf@server:/admins$ ls -ahl
total 12K
drwxrwsr-x  3 root root 4.0K Oct  9 10:23 .
drwxr-x--- 29 root root 4.0K Oct  9 10:21 ..
drwxrwsr-x  5 root root 4.0K Oct  9 10:29 username
```

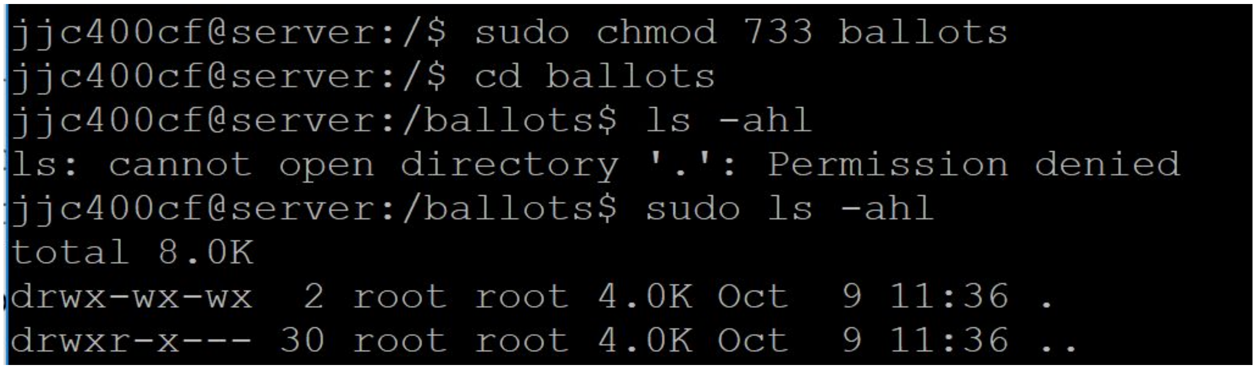
4.1.3 The Ballot Box

1.



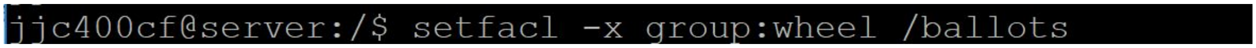
```
jjc400cf@server:/admins$ cd ..
jjc400cf@server:/$ sudo mkdir /ballots
jjc400cf@server:/$ ls
admins   dev      initrd.img      local
ballots  etc      initrd.img.old  lost+found
bin      groups   lib             media
boot     home     lib64           mnt
jjc400cf@server:/$ sudo chmod 733 ballots
```

2.



```
jjc400cf@server:/$ sudo chmod 733 ballots
jjc400cf@server:/$ cd ballots
jjc400cf@server:/ballots$ ls -ahl
ls: cannot open directory '.': Permission denied
jjc400cf@server:/ballots$ sudo ls -ahl
total 8.0K
drwx-wx-wx  2 root root 4.0K Oct  9 11:36 .
drwxr-x--- 30 root root 4.0K Oct  9 11:36 ..
```

3.



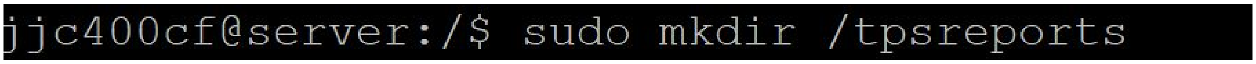
```
jjc400cf@server:/$ setfacl -x group:wheel /ballots
```

4. It is not possible for employees to see the ballots of other users for it is set that only the owner has the read privilege in the ballot directory. Therefore unless especially configured by the owner it is not possible.

5.the third ’x’ bit is the execute permission of the file or direction.

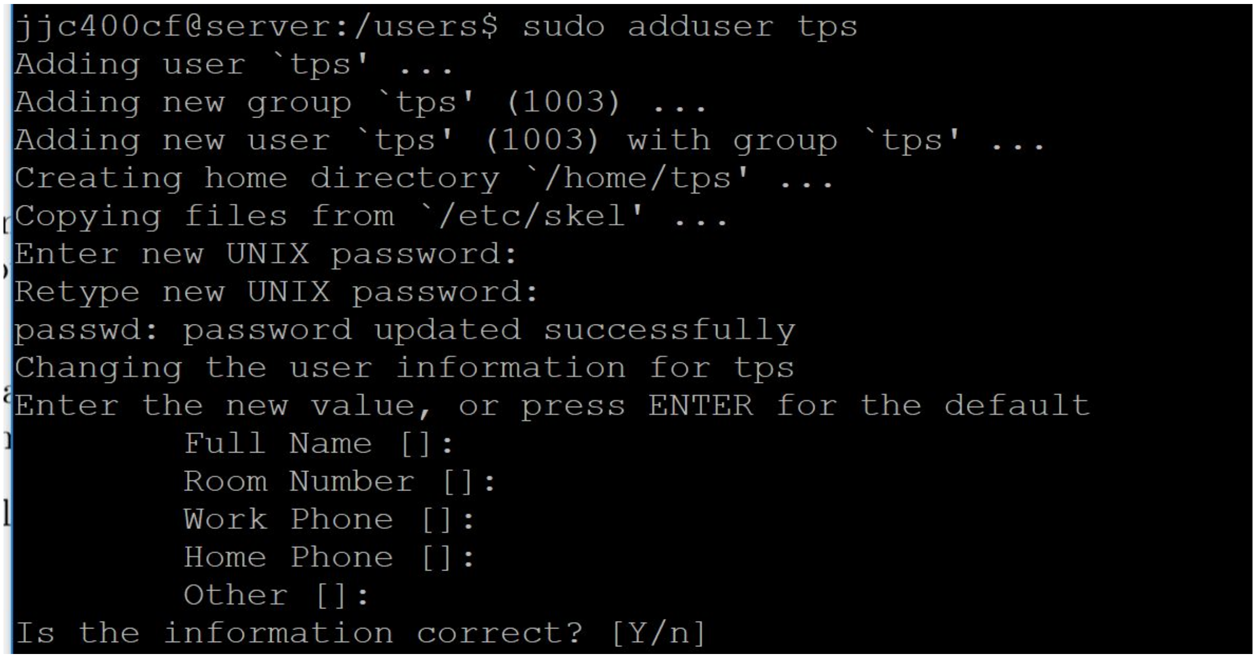
4.1.4 The TPS Reports Directory

1.



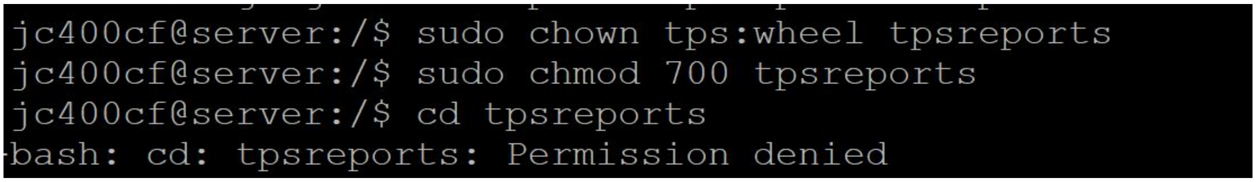
```
jjc400cf@server:/$ sudo mkdir /tpsreports
```

2.



```
jjc400cf@server:/users$ sudo adduser tps
Adding user `tps' ...
Adding new group `tps' (1003) ...
Adding new user `tps' (1003) with group `tps' ...
Creating home directory `/home/tps' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for tps
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
```

3.



```
jc400cf@server:/$ sudo chown tps:wheel tpsreports
jc400cf@server:/$ sudo chmod 700 tpsreports
jc400cf@server:/$ cd tpsreports
-bash: cd: tpsreports: Permission denied
```

4.

5.

4.1.5

sudo: Editing Configuration Files

-It is not safe to give larry sudo access to vim because this gives larry access to making changes with the file configurations. We can sudo configure it such a way that larry has access to edit only the apache2.conf file and nothing else, and this is a safer method for larry will not have complete sudo privileges.

4.1.6

sudo: Restarting System Processes

-it is not secure for larry to have sudo access to the command /etc/init.d/apache2 , for it might contain sensitive information that can be used to attack the system.

A better way would be to edit configuration in sudoer where larry could be configured to restart the apache2 file.

4.1.7
UNIX and sudo: Two Wrongs Make a Much Bigger Wrong
Only root has the permission to modify the files. And if you check the sudoers file it is a read only file, therefore normal users such as curly and moe do not have authority to make any changes in the files, unless it is configured to do so by the root by using sudo and reconfiguring the sudoers file .

4.2 Insiders, and that Man in the Middle

4.2.1 Eavesdropping

1. What data is being transmitted in cleartext?
    - The ports I can see being used are ports 23, 443, and 80. The protocols being used are Telnet and HTTP/s.
    - Some meaningful data I can extract from the 10.1.1.2 are applications/services running based off the telnet connection running on port 23. Along some of the port 80 connections I can see HTMl chunked text traffic passing through both Alice and Bob's connections.
    - I can also see a text file which is sent over port 80, there's also a seperate one containing text from the “Matrix”.

2. Authentication information sent over the wire.
    - Yes I can see telnet usernames and passwords being jimbo, jumbo, and jambo with the passwords visible in Telnet traffic is showing a connection between 10.1.1.3 and 10.1.1.2. I can see a bunch of login attempts going through port 23 passwords such as donald78, minnie77, goofy76.You can view the additional plain text authentication through inspecting the packets sent through the network using a packet sniffer for unencrypted protocols/ports.

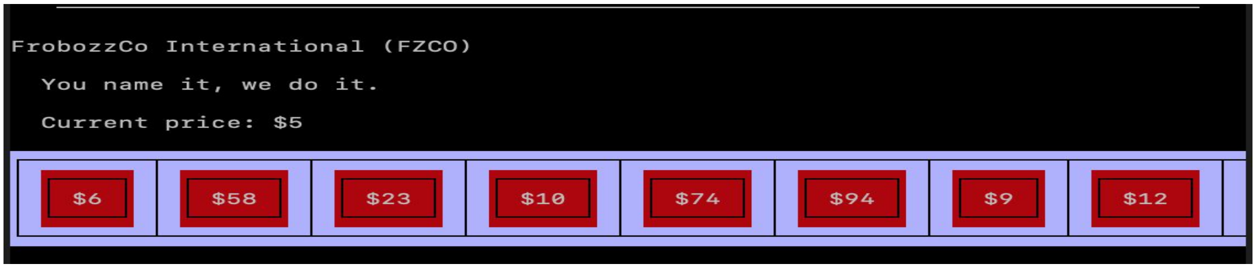
3. Encrypted Traffic
    - Yes there is encrypted traffic running through the network the port used is HTTPS 443.

4.2.2 Replay Attack against the Stock Ticker

- In order to perform a replay attack we must first initiate a new arp poisoning attack against the two targets 10.1.1.2 and 10.1.1.3.
- After this, I used the command to get the tcp packet data using tcpdump -i eth3 -s 1500 -X -w output.pcap. This gave me a bunch of http data.

- I focused on the get request for the stock ticker, and curled the link in order to parse an html file which contained the stock prices and graphs.
- Once this was complete I used the tool chaosreader using the output.pcap the command used is chaosreader on the output.pcap. This created a ton of useful files which I used later on in order to find the cryptographic hash values for the stock pages.

1. 10.1.1.3/cgi-bin/stock.cgi?symbol=DNGL&new=93&hash=4ddb36779f73a2f1eea628018271e3d and 10.1.1.3/cgi-bin/stock.cgi?symbol=FZCO&new=5&hash=2e44e0cae6f555a563387dc381334929
   By refreshing the page eventually we will attain the values associated for each new price point being $93 for DNGL and $5 for FZCO.
   2. Basically I continuously curled/elinked the two web addresses above until I received the values I wanted, unfortunately this took quite a bit of patience to occur.
   3. In the screenshot below it shows that I was able to replay attack the FZCO value to $5.



FrobozzCo International (FZCO)

You name it, we do it.

Current price: $5

4.2.3 Insertion Attack

To start, I needed to create the filter to be used in order to replace (insert) the stock name OWND into the field where FZCO was previously:

#ownd.filter

```
#if the data is TCP/IP traffic DESTINED for port 80 (HTTP)
# since we're concerned with data from server to browser
if (ip.proto == TCP && tcp.dst == 80) { # could also use ip.src or ip.dst
# if the packet contains "FZCO"...
if (search(DATA.data, "FZCO")) {
replace("FZCO", "OWND");
msg("Replaced stock names (symbols).");
        }
}
#end of filter
```

For the second filter I also created another filter as followed:

```
#replace.filter
if (ip.proto == TCP && tcp.dst == 80) {
if (search(DATA.data, "$")) {
replace("$", "$0");
msg("Replaced stock prices.");
}
}
# end of filter
```

Now when creating these files we must be aware that the .filter extension they have, needs to be converted into a file format which is usable by ettercap.

1. Given the power of etterfilter and the kinds of traffic on this network, you can actually make significant changes to a machine or machines that you're not even logged in to. How?

   Insertion attacks can be used to cause harm to network even remote machines. This can be done by an insertion which downgrade a protocol to a weaker version which leads to weaknesses in communication between alice and bob.

2. Of the cleartext protocols in use, can you perform any other dirty tricks using insertion attacks? The more nasty and clever they are, the better.

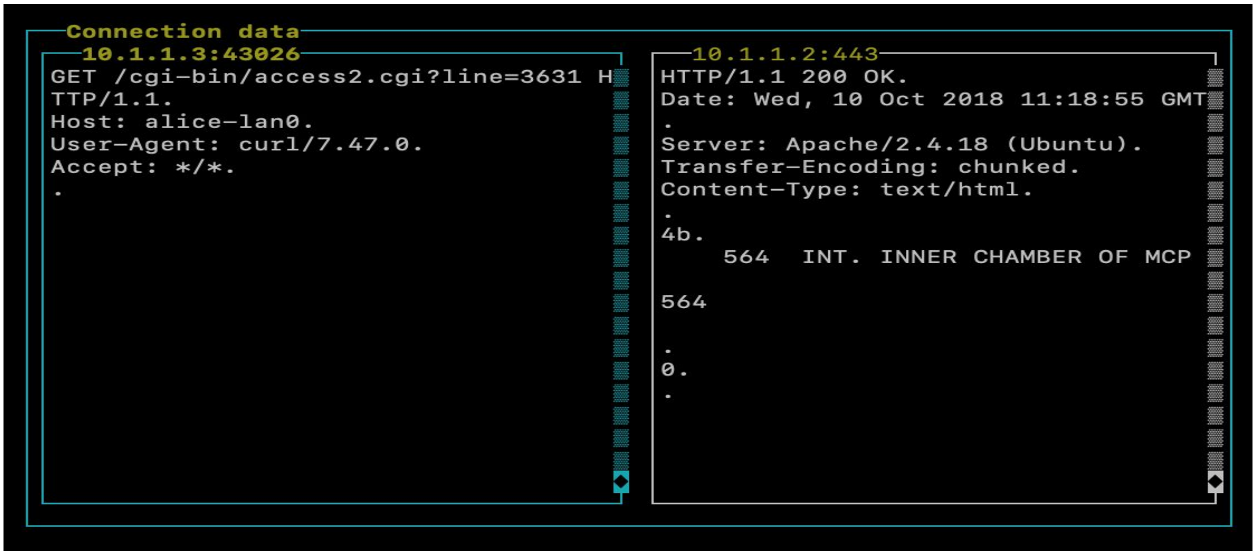
Some ways that insertion can be used are in the transmission of malicious files via two parties (or more), basically the communication can be intertercepted and the file can be completely replaced with another file without anyone knowing. This can also be done to modify http requests in order to execute code.

4.2.4 MITM vs. Encryption

1. What configuration elements did you have to change?

Inside the etter.conf file located at the root inside the etc folder, I had to change ec_uid= 0 and gc_uid=0 this enabled root privileges. I also needed to modify a portion of the iptables near the bottom of the conf file which were redirect commands to enable the decryption of the SSL communication.

2. Copy and paste some of this data into a text file and include it in your submission materials.



```
GET /cgi-bin/access2.cgi?line=3631 H
TTP/1.1.
Host: alice-lan0.
User-Agent: curl/7.47.0.
Accept: */*.
.
```

```
HTTP/1.1 200 OK.
Date: Wed, 10 Oct 2018 11:18:55 GMT
.
Server: Apache/2.4.18 (Ubuntu).
Transfer-Encoding: chunked.
Content-Type: text/html.
.
4b.
    564   INT. INNER CHAMBER OF MCP

564

.
0.
.
```

3. Why doesn’t it work to use tcpdump to capture this ”decrypted” data?
TCPDump does not work for this data because it does not have the required dependency to work with decryption on packets.
4. For this exploit to work, it is necessary for users to blindly ”click OK” without investigating the certificate issues. Why is this necessary?
This is necessary because if not the actual HTTPs certificates would be used and would be encrypted. Ettercap however, uses its own spoofed certificates which allows for the decryption of SSL traffic.
5. What is the encrypted data they’re hiding?
According to the screenshots provided earlier, the data hiding is the TRON script located here:
https://www.imsdb.com/scripts/TRON.html

Division of Labor:

Yonga Lhamo, Kristina Lagasca- 4.1

Michael Gonzalez, Ann Yassim - 4.2