Name : Laxmi Deepika Manchikanti
SBU Id : 113324192

Homework 4: Plugboard Proxy

Implementation of pbproxy.go:

Data flow :

ssh <--stdin/stdout--> pbproxy-c <--socket 1--> pbproxy-s <--socket 2--> sshd
_____/ _____/
client server

Main function :

- Takes the listening port in input.
- Reads the pwdFile for the symmetric key.

Client Side:

In client mode, I'm using 2 functions -

1. stdinreading -
    - Reads from the standard input
    - Then the data Buffer received from standard input has been encrypted
    - Encrypted data was written to the socket 1 thus, sending it to the server.

2. socketreading -
    - Reads from socket 1
    - Then the data buffer received from the socket 1 was decrypted
    - Decrypted data has been written to the standard output.

Server Side:

In the server side, I'm using 2 functions -

1. clientproxymode -
    - Reads from the socket 1
    - Then the data Buffer received from socket 1 was decrypted.
    - The decrypted buffer has been written to the socket 2.

2. sshdreading -
    - Reads from socket 2
    - Then the data buffer received from the socket 2 was encrypted

- Encrypted data was written to the socket 1.
- Code was also written such that - In any instance, If the connection was lost, the server would wait for a new connection to be established.

Cryptography:

I'm using 3 functions :

1. get_key -
   - For the client to use the same symmetric key used by the server to encrypt the traffic.
   - A crypto key was generated using 'PBKDF2' with SHA256 algorithm and 1000 iterations when a passphrase was passed and an array of [key,salt] was returned.

2. data_encryption -
   - When a passphrase was passed along with plaintext, a key was generated with new salt.
   - Then, the plaintext was encrypted with the derived key using AES-GCM.
   - Salt,iv and data(ciphertext) have been hex encoded and joined by '-', thus returning in the format 'salt-iv-data'.

3. data_decryption -
   - When a key and ciphertext were passed, the ciphertext was decrypted.
   - Original plaintext was returned.

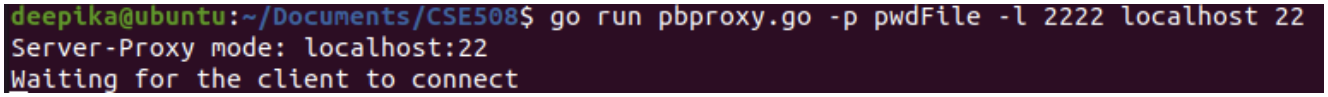
Test case :

SSH was running.
pwdFile had the key written in it.

➔ **Test case 1**

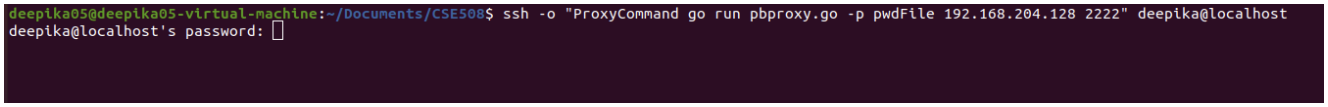
In the server machine,

Command: go run pbproxy.go -p pwdFile -l 2222 localhost 22
Output :
Server-Proxy mode: localhost:22
Waiting for the client to connect



In the client machine,

Command : ssh -o “ProxyCommand go run pbproxy.go -p pwdFile 192.168.204.128 2222” deepika@localhost
Output:
deepika@localhost’s password:



When password was entered,
Response from the client machine was :

Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.8.0-50-generic x86_64)

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

43 updates can be installed immediately.

0 of these updates are security updates.

To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.

Last login: Sat May 1 03:35:12 2021 from 127.0.0.1

deepika@ubuntu:~$

- Prompted for the password and once entered, able to ssh into it successfully.

Then, the response from server machine :

Yay! Client has just connected.
Waiting for the client to connect
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:57 1024
1024
2021/05/01 13:21:57 1024
1024
2021/05/01 13:21:58 1024
1024
2021/05/01 13:22:05 1024
1024
2021/05/01 13:22:05 1024
1024
2021/05/01 13:22:05 1024
1024
2021/05/01 13:22:06 1024
1024
2021/05/01 13:22:07 1024
1024

2021/05/01 13:22:07 1024
1024
2021/05/01 13:22:07 1024
1024
2021/05/01 13:22:07 1024
1024
2021/05/01 13:24:43 1024

```
deepika@ubuntu:~/Documents/CSE508$ go run pbproxy.go -p pwdFile -l 2222 localhost 22
Server-Proxy mode: localhost:22
Waiting for the client to connect
Yay! Client has just connected.
Waiting for the client to connect
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:43 1024
1024
2021/05/01 13:21:57 1024
1024
2021/05/01 13:21:57 1024
1024
2021/05/01 13:21:58 1024
1024
2021/05/01 13:22:05 1024
1024
2021/05/01 13:22:05 1024
1024
2021/05/01 13:22:05 1024
1024
2021/05/01 13:22:06 1024
1024
2021/05/01 13:22:07 1024
1024
2021/05/01 13:22:07 1024
1024
2021/05/01 13:22:07 1024
1024
2021/05/01 13:22:07 1024
1024
2021/05/01 13:24:43 1024
1024
```

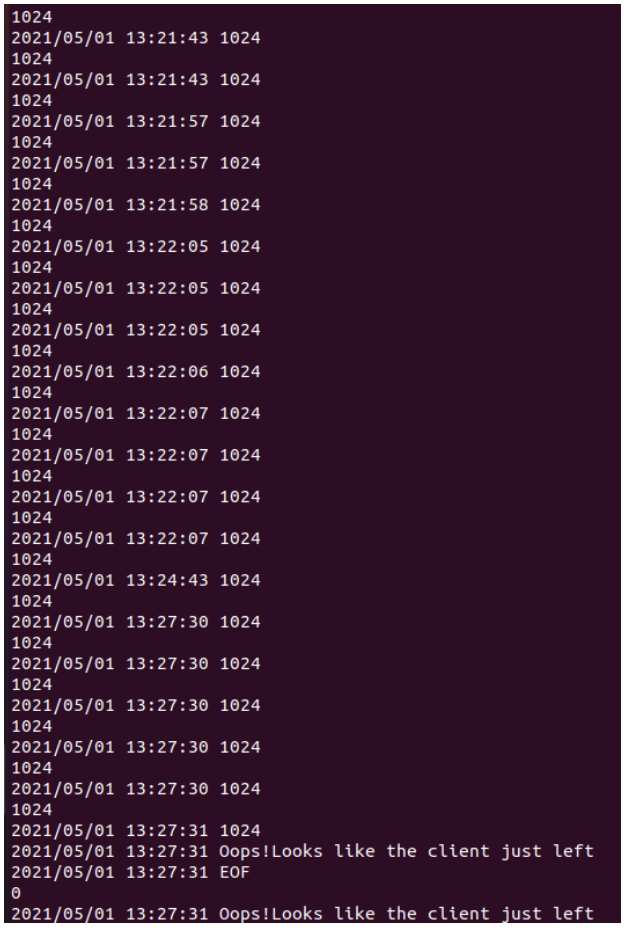
On the client machine,when exited:

deepika@ubuntu:~$ exit
logout
Connection to localhost closed.

```
deepika@ubuntu:~$ exit
logout
Connection to localhost closed.
deepika05@deepika05-virtual-machine:~/Documents/CSE508$
```

The server machine responded with:

2021/05/01 13:27:31 Oops!Looks like the client just left
2021/05/01 13:27:31 EOF
0
2021/05/01 13:27:31 Oops!Looks like the client just left



➔ **Test case 2**

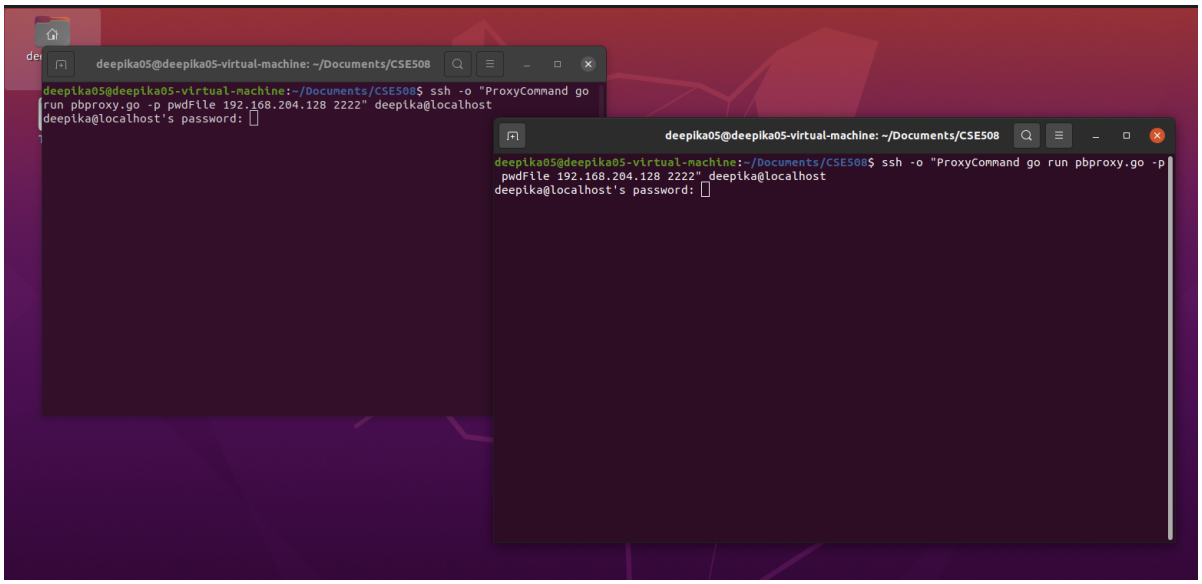
Similarly, trying to establish a second connection
We run the command : go run pbproxy.go -p pwdFile -l 2222 localhost 22 on the server machine.

```
deepika@ubuntu:~/Documents/CSE508$ go run pbproxy.go -p pwdFile -l 2222 localhost 22
Server-Proxy mode: localhost:22
Waiting for the client to connect
Yay! Client has just connected.
Waiting for the client to connect
1024
2021/05/01 14:36:09 1024
1024
2021/05/01 14:36:09 1024
1024
2021/05/01 14:36:09 1024
1024
2021/05/01 14:36:09 1024
1024
2021/05/01 14:36:09 1024
1024
2021/05/01 14:36:09 1024
Yay! Client has just connected.
Waiting for the client to connect
1024
2021/05/01 14:36:16 1024
1024
2021/05/01 14:36:16 1024
1024
2021/05/01 14:36:16 1024
1024
2021/05/01 14:36:17 1024
1024
2021/05/01 14:36:17 1024
1024
2021/05/01 14:36:17 1024
```

The following output was received since 2 clients were connected.

On the multiple clients' side:

Command :ssh -o "ProxyCommand go run pbproxy.go -p pwdFile 192.168.204.128 2222" deepika@localhost
was used on 2 machines.
Output received:
deepika@localhost's password:



- Prompted for the password and once entered, able to ssh into it successfully.

So, the server didn’t exit after one connection was established and it waited for a new connection.