CS-475 Cortney Reed
Assignment 4 06/07/2020

Assignment Discussion

The purpose of Assignment 4 was to create a simple network security protocol similar to that of SSH. There are two programs the client and server both written in Python, the client sends encrypted messages to the server, and the server decrypts those messages. The way it works is once the server program is up and running, the user starts the client program. Both programs communicate over the internet using Python's socket module (TCP protocol). The client programs display two options; option one is to log in into the server, and option two is to generate public and private RSA keys. Since the server authenticates based on the client's username, if the client does not have a public and private key for their username, you start with option two.

Option two starts by asking the client for their username, and based on that using the pyCrypto module, the program generates a 1024-bit RSA key pair. The program then exports the public and private keys to a directory. Along with the keys, the program appends the client's entered username to a file that contains all the clients. After the username has private and public keys created, the program brings you back to the start menu, and you can log in to the server. When you select option one, it asks for a username; if the public and private keys exist for that username, it grabs the public key, and using the pyCrypto module encrypts the client's name then sends it to the server.

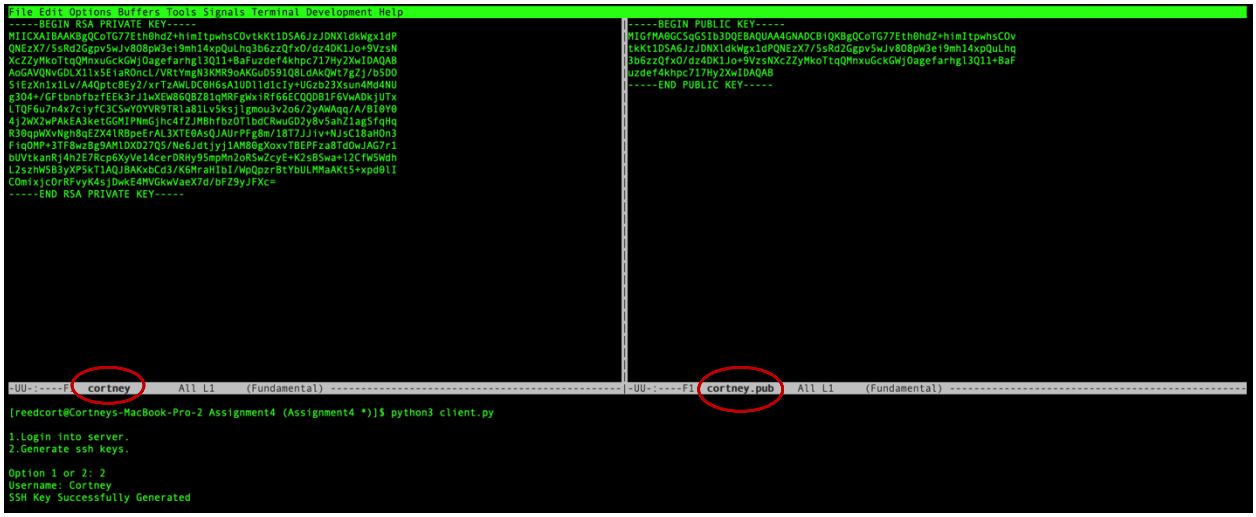
The server receives the encrypted message, opens the client list, and iterates through the entire list to find the associated private key and attempts to decrypt the message. Once the encrypted message is decrypted and matches the corresponding name on the client list, it sends a randomly generated a 32-byte session key along with an 8-byte initialization vector(IV). The session key and IV are used to encrypt and decrypt using pyCrypto's AES CTR block cipher. After the client program receives the session key and IV, the user can type any message, and the client program encrypts and sends it to the server program to display the encrypted and decrypted message.

To test if Assignment 4 works properly, I tested three things, the program's ability to generate unique public and private keys. Next, the program's ability to encrypt and decrypt messages using the public and private key. Lastly, the program's ability to encrypt and decrypt AES CTR block cipher. Below are some images of the test cases I ran to ensure my program worked properly.

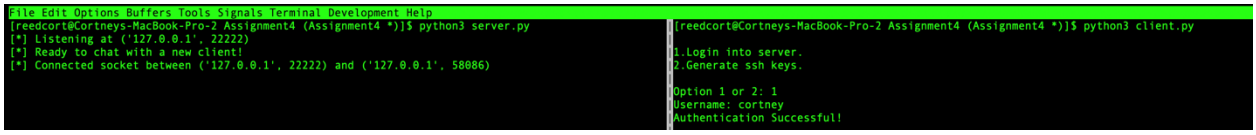
Even though the program works correctly, there are several security flaws. The first being that a person can log in to the server as long they have the client's username. Meaning an imposter with someone's username can log in as someone else. Another vulnerability is that a person could potentially intercept an encrypted transmitted message and can crack the AES CTR block size by cycling through a series of randomly generated IVs and session keys. A method to prevent imposters is by adding a second security measure like a password. Also, to prevent someone from intercepting and decoding the message, you could generate a larger session key to make it harder to decrypt.

Test Case:

1. Generate keys



2. Successfully connect to server and authenticate client



3. Server is abled to decrypt messages sent from client