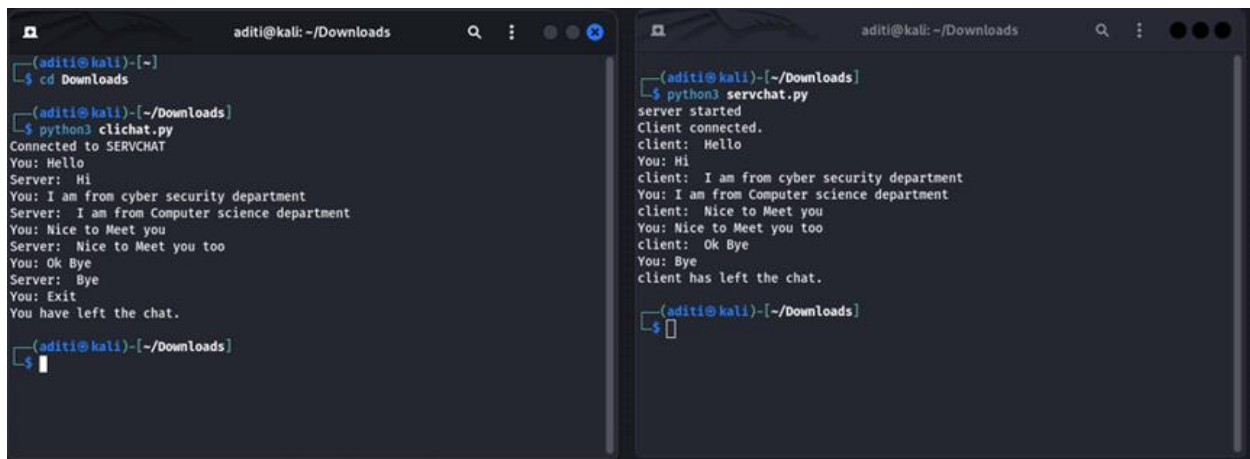


Understanding Network Communications

using socket programming

I wrote a socket program for client-server interaction that supports two-way communication. The program does not terminate until either the server or the client types a specific word, such as "EXIT" or "END." I chose the Python programming language for this implementation and included features for confidentiality and integrity. The client uses the RSA package to generate a public-private key pair and shares the public key with the server. The server selects a symmetric key and encrypts it using the client's public key before sending it to the client. The client then decrypts the message to obtain the symmetric key. I utilized the Fernet symmetric key cryptography for message encryption. Once the symmetric key is established, all messages exchanged between the client and server are encrypted. The sender also computes a hash (of your choice) of the ciphertext and sends it along with the ciphertext. Upon receiving a message, the receiver first verifies the hash and then decrypts the ciphertext.

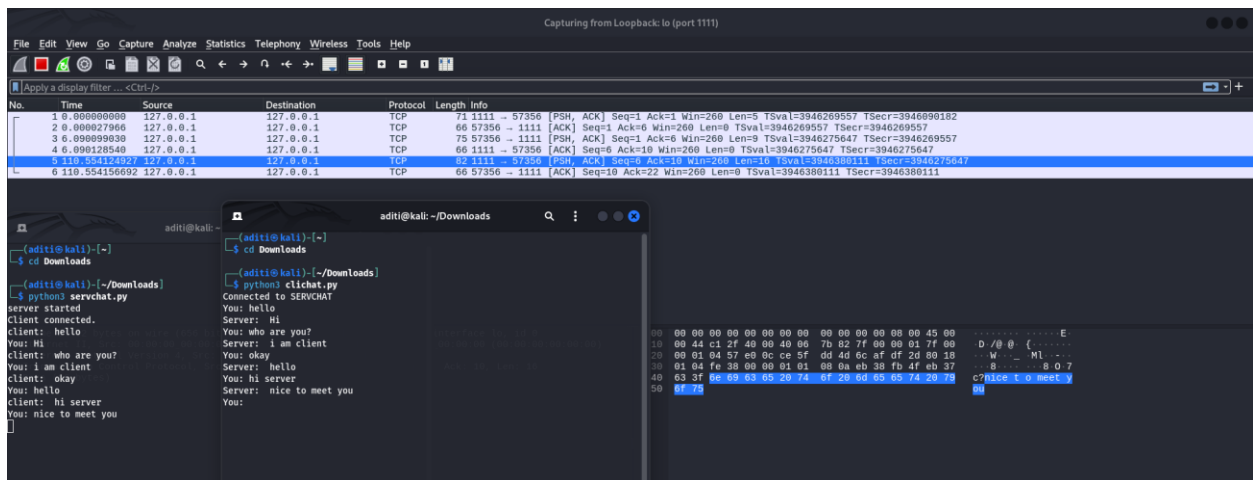
Communication without encryption



```
(aditi@kali)-[~]
└─$ cd Downloads
(aditi@kali)-[~/Downloads]
└─$ python3 clickat.py
Connected to SERVCHAT
You: Hello
Server: Hi
You: I am from cyber security department
Server: I am from Computer science department
You: Nice to Meet you
Server: Nice to Meet you too
You: Ok Bye
Server: Bye
You: Exit
You have left the chat.
(aditi@kali)-[~/Downloads]
└─$

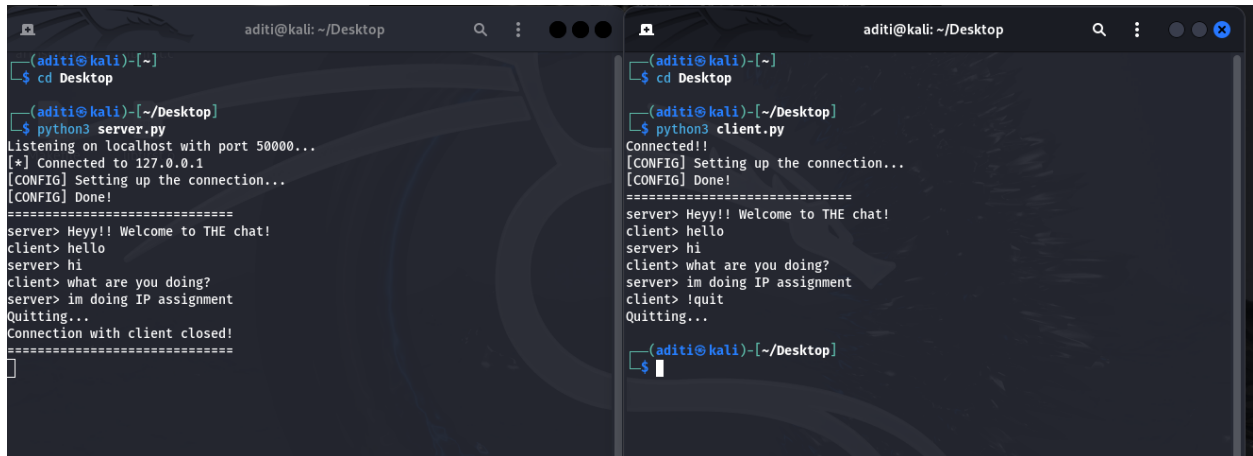
(aditi@kali)-[~/Downloads]
└─$ python3 servchat.py
server started
Client connected.
client: Hello
You: Hi
client: I am from cyber security department
You: I am from Computer science department
client: Nice to Meet you
You: Nice to Meet you too
client: Ok Bye
You: Bye
client has left the chat.
(aditi@kali)-[~/Downloads]
└─$
```

Wireshark to capture messages

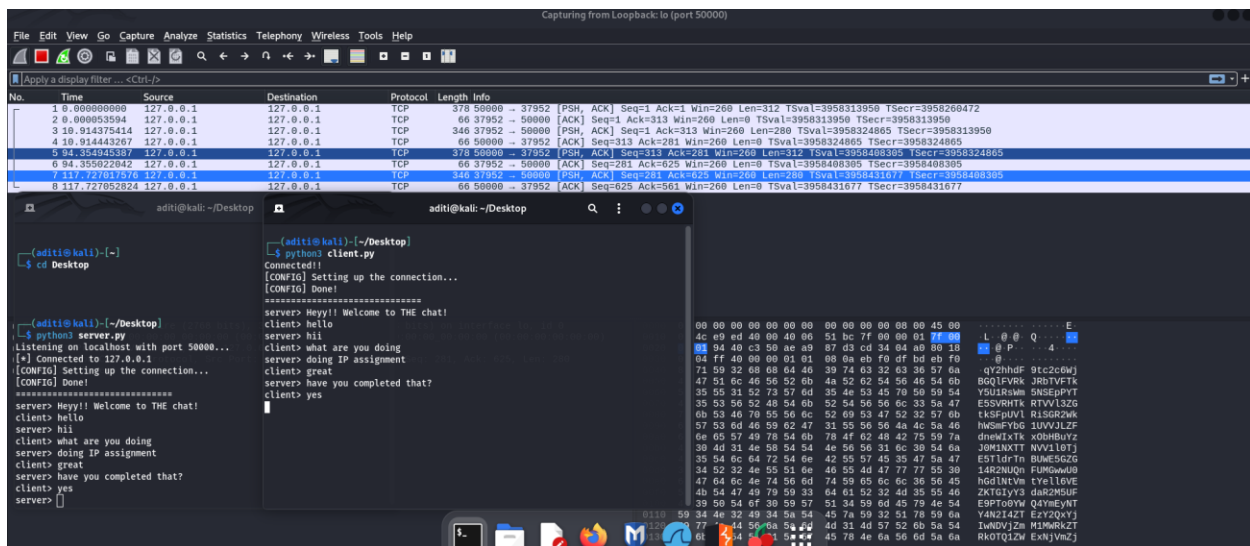


Here, we can see that without encryption “nice to meet you” message is visible in Wireshark.

Communication with encryption



Wireshark to capture messages



With encryption, we will not be able to capture the communication in Wireshark.

Here I used RSA, Fernet and HMAC. RSA ensures secure key exchange, Fernet ensures message confidentiality, and HMAC ensures message integrity.

1. RSA Encryption (Asymmetric):

- RSA is used during the key exchange phase. The client generates an RSA key pair (public and private keys).
- The server uses the client's RSA public key to encrypt the symmetric key and HMAC key.
- The client decrypts the keys using its RSA private key.
- This ensures that only the intended recipient (the client with the correct private key) can decrypt the server's message.

2. Symmetric Encryption (Fernet):

- Once the symmetric key is securely exchanged, the actual chat messages are encrypted using symmetric encryption with the Fernet module from the cryptography library.
- Fernet is a symmetric encryption scheme based on AES (Advanced Encryption Standard) with a 32-byte (256-bit) key. Messages are encrypted with this key for confidentiality.

3. HMAC (Hash-based Message Authentication Code):

- HMAC is used for integrity checking. It ensures that messages have not been tampered with during transmission.
- The server and client generate an HMAC for each message using a shared secret (the HMAC key), which they verify before processing messages. If the HMAC doesn't match, it indicates that the message's integrity is compromised.