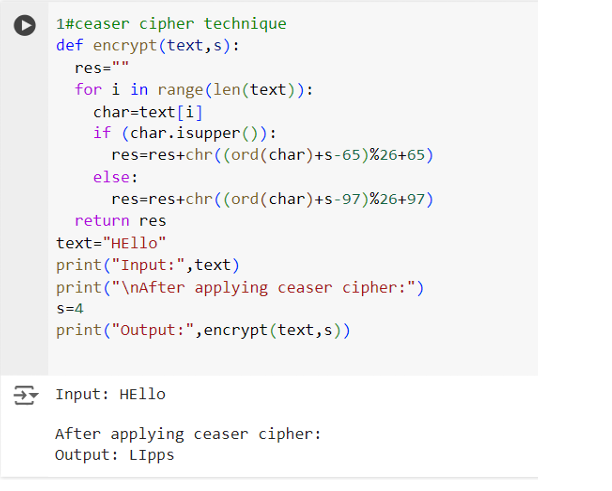
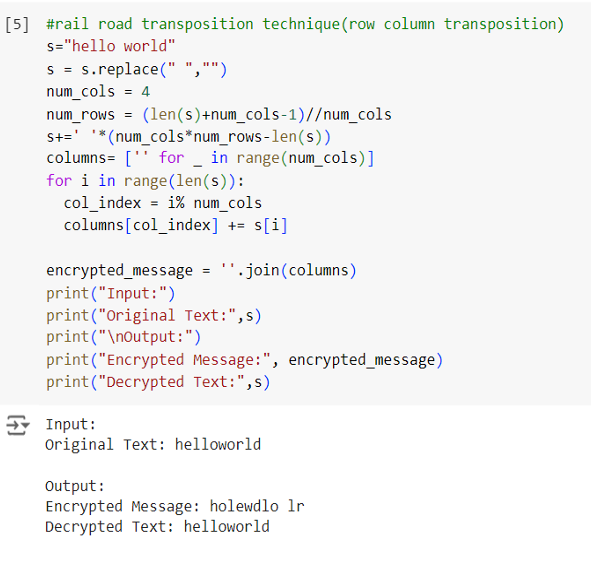
1. Implement the following substitution and Transposition techniques:

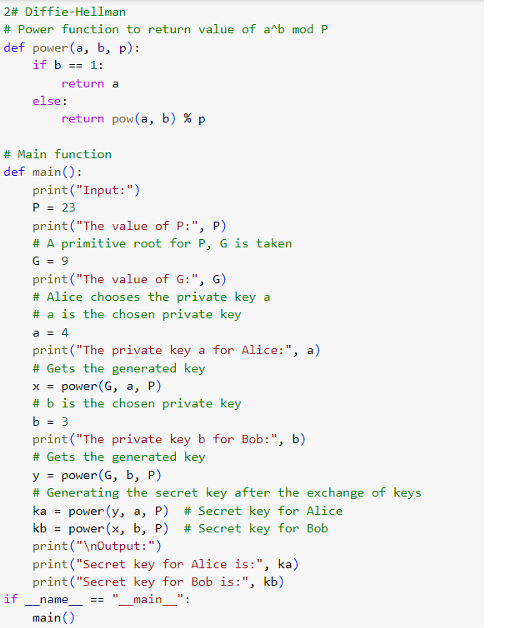
• Caesar Cipher

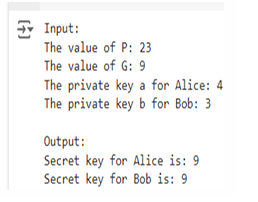
• Columnar Transposition Cipher





2. Implement Diffie-Hellman Algorithm to calculate key for Sender and Receiver.





3. Implement following Brute Force Attack

• Dictionary Attack



4. Demonstrate message exchange and data transmission between server and client to demonstrate TCP using Wireshark.

**tcp\_server.py**

import socket

HOST = '127.0.0.1' # Localhost

PORT = 65432 # Port to listen on

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as server\_socket:

server\_socket.bind((HOST, PORT))

server\_socket.listen()

print(f"Server started. Listening on {HOST}:{PORT}...")

conn, addr = server\_socket.accept()

with conn:

print(f"Connected by {addr}")

while True:

data = conn.recv(1024)

if not data:

break

print(f"Received: {data.decode()}")

conn.sendall(data) # Echo the data back to the client

**tcp\_client.py**

import socket

HOST = '127.0.0.1'  # The server's hostname or IP address

PORT = 65432        # The port used by the server

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as client\_socket:

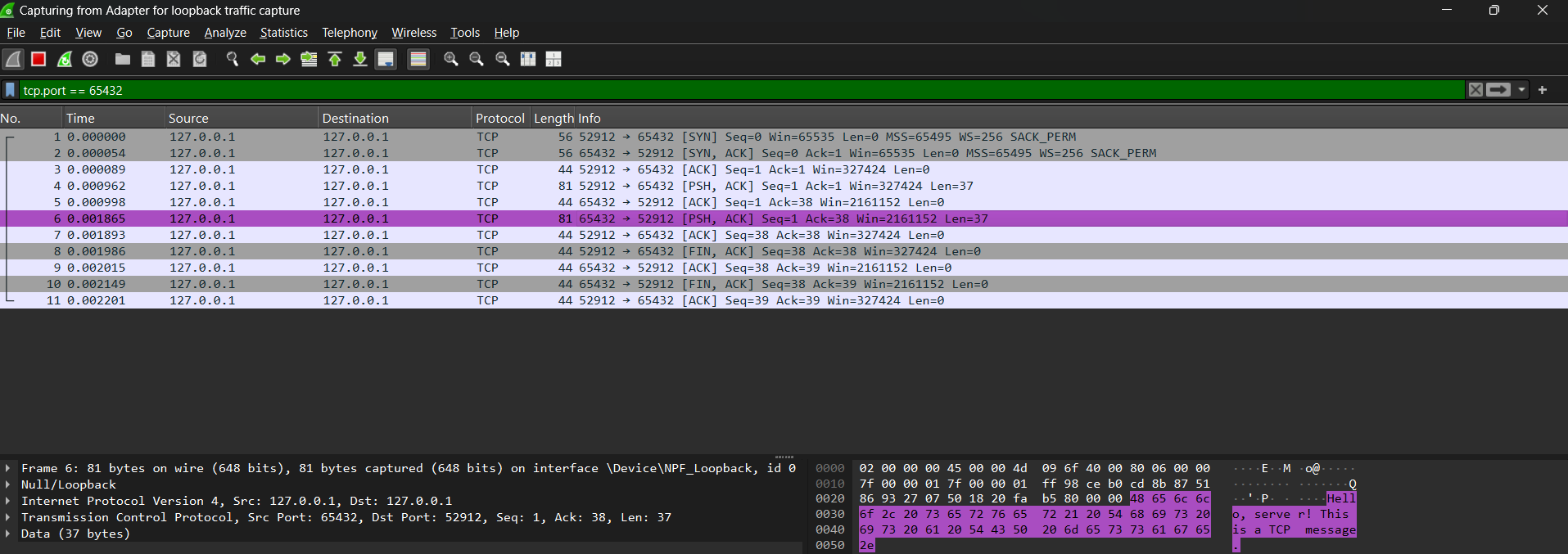
    client\_socket.connect((HOST, PORT))

    print("Connected to the server.")

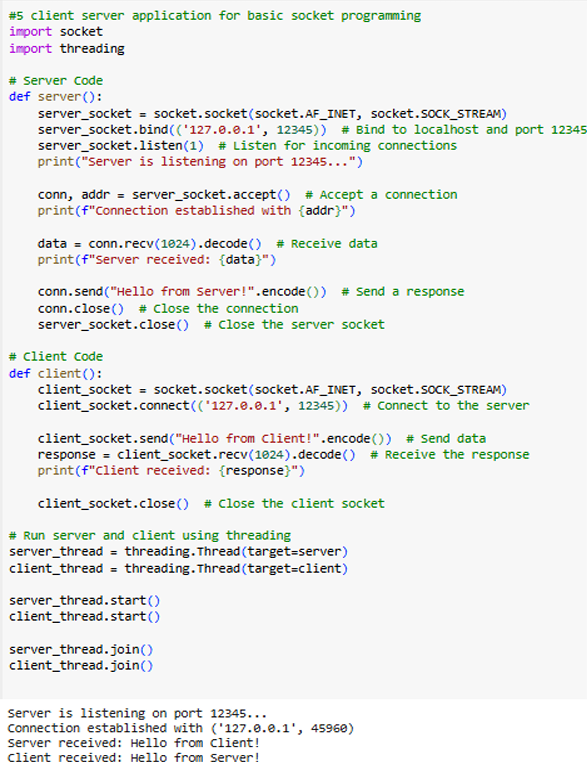
    client\_socket.sendall(b'Hello, server! This is a TCP message.')

    data = client\_socket.recv(1024)

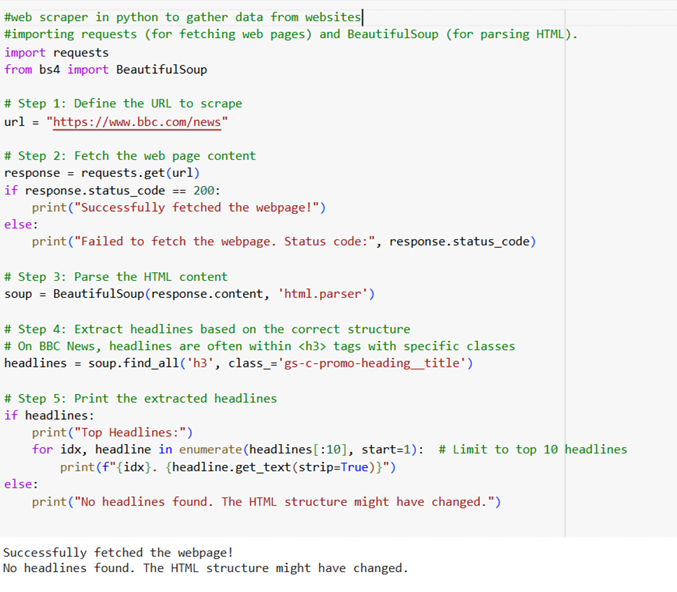
print(f"Received back: {data.decode()}")

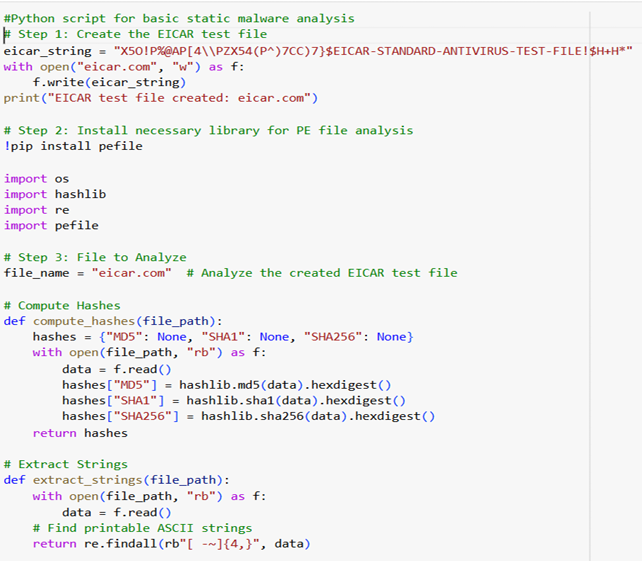


5. To build a simple python client server application to understand basic socket programming.



6. How to create web scraper in python to gather data from websites.



7. Write a python scripts for basic static malware analysis.

