

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
KHOA AN TOÀN THÔNG TIN**



**BÁO CÁO BÀI THỰC HÀNH
HỌC PHẦN: THỰC TẬP CƠ SỞ
MÃ HỌC PHẦN: INT13147**

BÀI THỰC HÀNH 4.1

LẬP TRÌNH CLIENT/SERVER ĐỂ TRAO ĐỔI THÔNG TIN AN TOÀN

Sinh viên thực hiện: B22DCAT176 Nguyễn Thị Thùy Linh

Giảng viên hướng dẫn: PGS.TS Hoàng Xuân Dậu

HỌC KỲ 2 NĂM HỌC 2024-2025

MỤC LỤC

MỤC LỤC	2
DANH MỤC CÁC HÌNH ẢNH	5
DANH MỤC CÁC TỪ VIẾT TẮT	3
CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH	5
I. Mục đích	5
II. Tìm hiểu lý thuyết.....	5
1. Socket	5
CHƯƠNG 2. NỘI DUNG THỰC HÀNH	9
I. Chuẩn bị môi trường	6
II. Các bước thực hiện	6
1. Lập trình client và server với TCP socket.....	6
2. Trao đổi thông điệp giữa client và server và đảm bảo tính toàn vẹn của thông điệp.....	9
TÀI LIỆU THAM KHẢO	15

DANH MỤC HÌNH ẢNH

MỤC LỤC.....	2
Hình ảnh 1: Cách thức hoạt động của Socket.	5
Hình ảnh 2: Lập trình Server.....	6
Hình ảnh 3: Lập trình Client	7
Hình ảnh 4: Thực hiện dir trên server	7
Hình ảnh 5: Thực hiện run server.py.....	8
Hình ảnh 6: Thực hiện run file client.py.....	8
Hình ảnh 7: Bắt gói tin server.....	9
Hình ảnh 8: Bắt gói tin client	9
Hình ảnh 9: Sửa code lại Server.	10
Hình ảnh 10: Sửa code lại Client.	10
Hình ảnh 11: Chạy file server.py.....	11
Hình ảnh 12: Chạy file client.py	11
Hình ảnh 13: Bắt gói file giá trị bấm	11
Hình ảnh 14: Bắt gói file client.....	12
Hình ảnh 15: Tạo key bên client mới.....	12
Hình ảnh 16: Thu được giá trị bấm của server.	13
Hình ảnh 17: Thu được giá trị bấm của client.	13
Hình ảnh 18: Bắt được gói tin giá trị bấm mới.	13
Hình ảnh 19: Bắt gói tin client.	14
TÀI LIỆU THAM KHẢO	15

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Thuật ngữ tiếng Anh/Giải thích	Thuật ngữ tiếng Việt/Giải thích
SIEM	Security Information and Event Management	Giải pháp giám sát, phát hiện và phản ứng nhanh với mối đe dọa.
DNS	Domain Name System	Giao thức tên miền
DHCP	Dynamic Host Configuration Protocol	Giao thức cấp phát địa chỉ IP tự động
IIS	Internet Information Services	Dịch vụ Web Server chính của Microsoft
WAS	Windows Process Activation Service	Dịch vụ hỗ trợ cho IIS

CHƯƠNG 1. GIỚI THIỆU CHUNG VỀ BÀI THỰC HÀNH

I. Mục đích

- Hiểu về cơ chế client/server và có thể tự lập trình client/server dựa trên socket, sau đó thực hiện cài đặt giao thức đơn giản để trao đổi thông tin an toàn.

II. Tìm hiểu lý thuyết

1. Scket

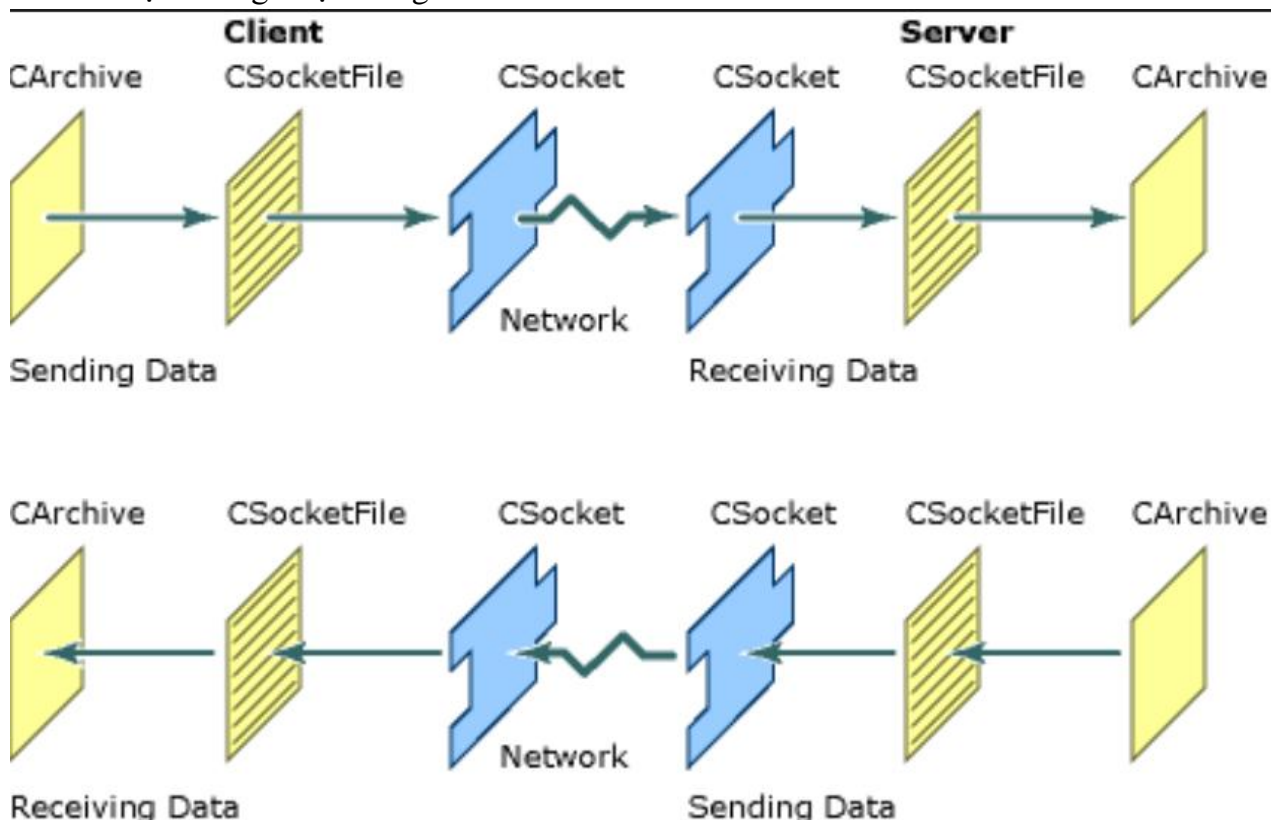
Socket là điểm cuối end-point tại liên kết truyền thông 2 chiều (two-way communication) và biểu diễn kết nối giữa Server – Client. Những lớp Socket hiện đang ràng buộc với 1 cổng port (thể hiện là 1 con số cụ thể) để những tầng TCP (hay TCP Layer hoàn toàn có thể định danh được ứng dụng mà dữ liệu gửi đến).

Cách thức hoạt động: Hiện tại, chức năng của Socket chính là kết nối giữa server và client thông qua UDP, TCP/IP để có thể truyền cũng như nhận dữ liệu thông qua Internet

Hiện tại giao diện của lập trình ứng dụng mạng chỉ có thể hoạt động nếu như đã có những thông tin liên quan đến thông số IP cũng như số hiệu cổng của hai ứng dụng cần phải trao đổi dữ liệu.

Như vậy hai ứng dụng đang cần truyền thông bắt buộc phải đáp ứng được những điều kiện cơ bản sau đây thì socket mới hoạt động, cụ thể:

- Hai ứng dụng hoàn toàn có thể nằm cùng trên một máy hay hai máy khác nhau.
- Đối với trường hợp nếu như hai ứng dụng cùng trên một máy thì hiệu số cổng bắt buộc không được trùng với nhau.



Hình ảnh 1: Cách thức hoạt động của Socket.

CHƯƠNG 2 : NỘI DUNG THỰC HÀNH

I. Chuẩn bị môi trường

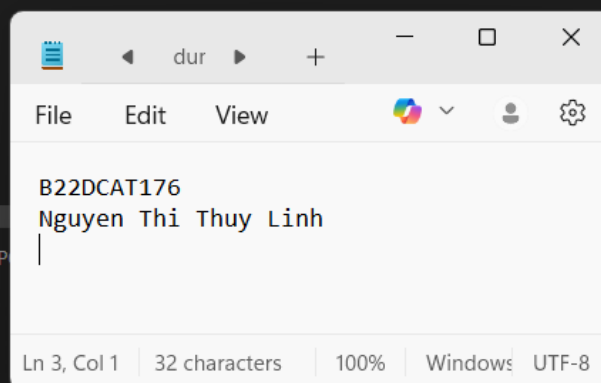
- Cài đặt công cụ ảo hóa
- Cài đặt các công cụ Wireshark

II. Các bước thực hiện

1. Lập trình client và server với TCP socket

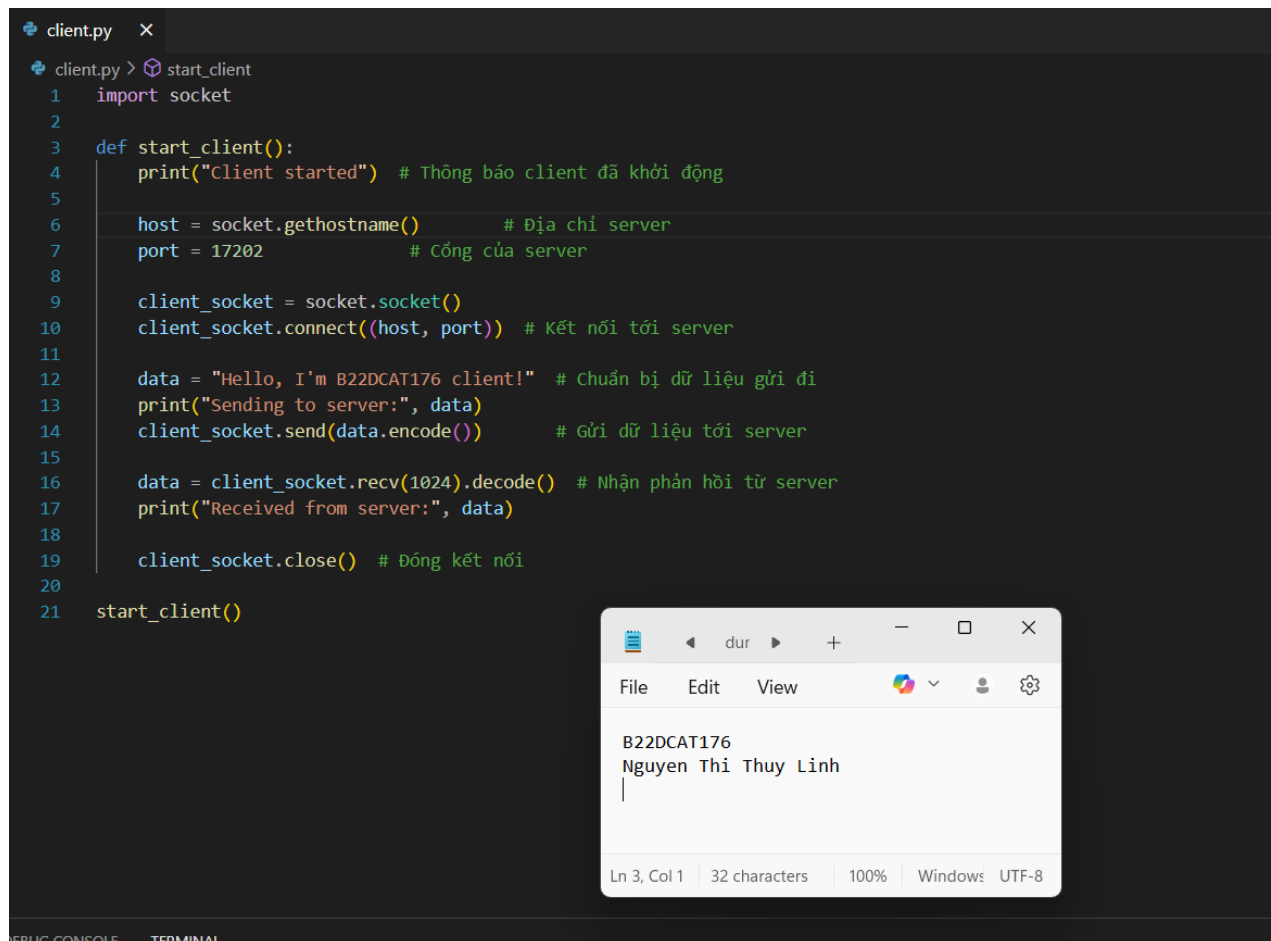
Lập trình Server:

```
CodeSocket > server.py > ...
1  import socket
2
3  def start_server():
4      print("Server started") # Thông báo server đã khởi động
5
6      host = socket.gethostname() # Địa chỉ localhost
7      port = 17202 # Cổng lắng nghe
8
9      server_socket = socket.socket()
10     server_socket.bind((host, port)) # Gắn địa chỉ và cổng
11     server_socket.listen(1) # Cho phép tối đa 1 kết nối đang chờ
12
13     conn, addr = server_socket.accept() # Chấp nhận kết nối từ client
14
15     data = conn.recv(1024).decode() # Nhận dữ liệu từ client
16     print("Received from client:", data)
17
18     data = "Hello, I'm B22DCAT176 server!" # Chuẩn bị dữ liệu phản hồi
19     print("Sending to client:", data)
20     conn.send(data.encode()) # Gửi phản hồi
21
22     conn.close() # Đóng kết nối
23
24     start_server()
```



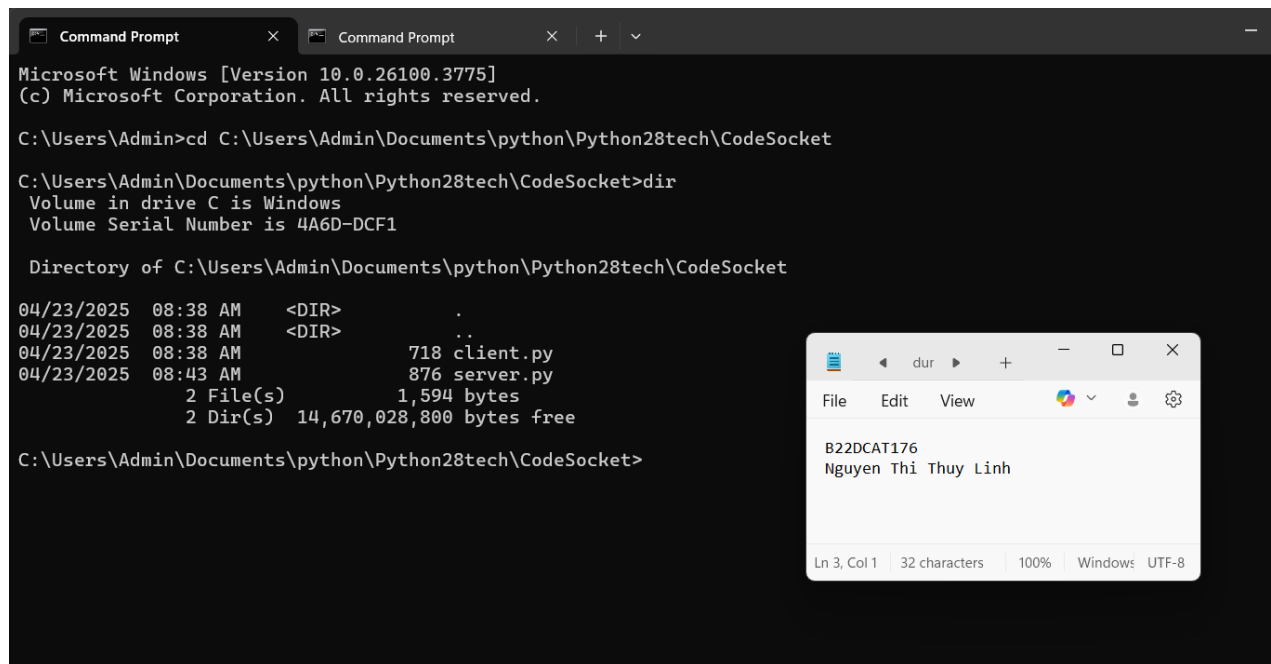
Hình ảnh 2: Lập trình Server

Lập trình Client.



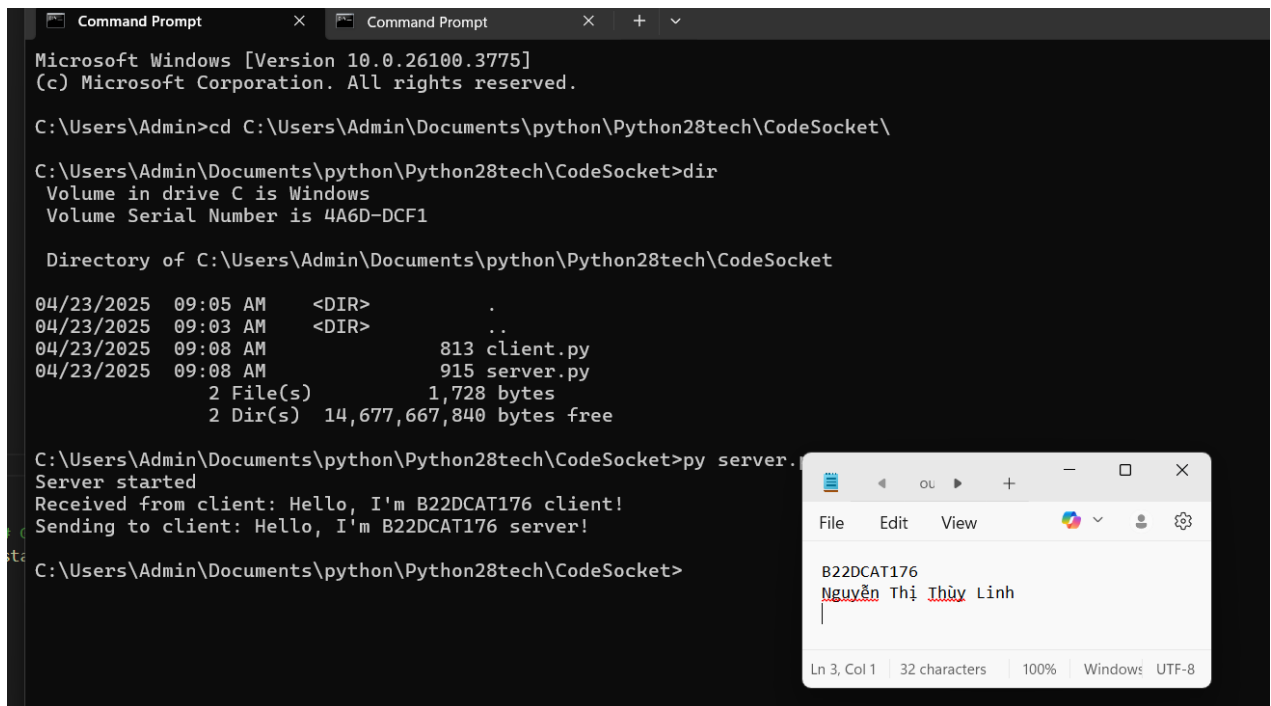
Hình ảnh 3: Lập trình Client

Tạo ra 2 tab trong Command Prompt để thực hiện chạy server và client. Chạy câu lệnh “dir” để xem liệt kê các thư mục và file con.



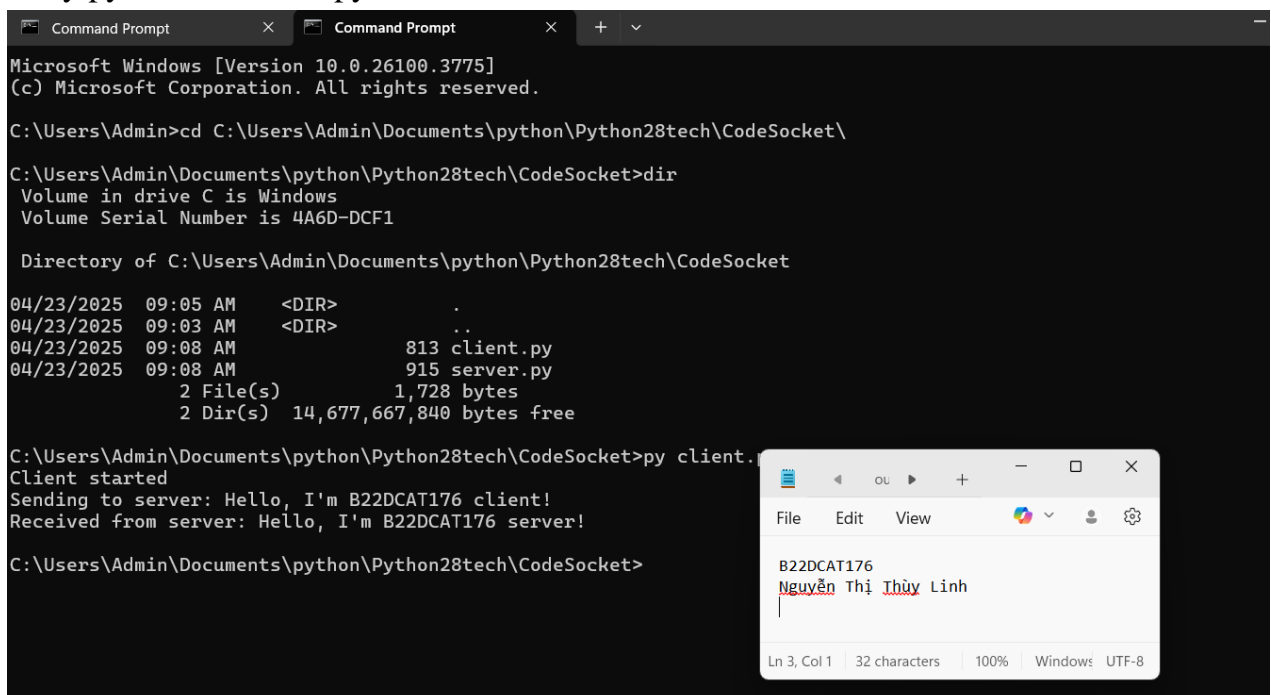
Hình ảnh 4: Thực hiện dir trên server

Chạy python file server.py



Hình ảnh 5: Thực hiện run server.py

Chạy python file client.py



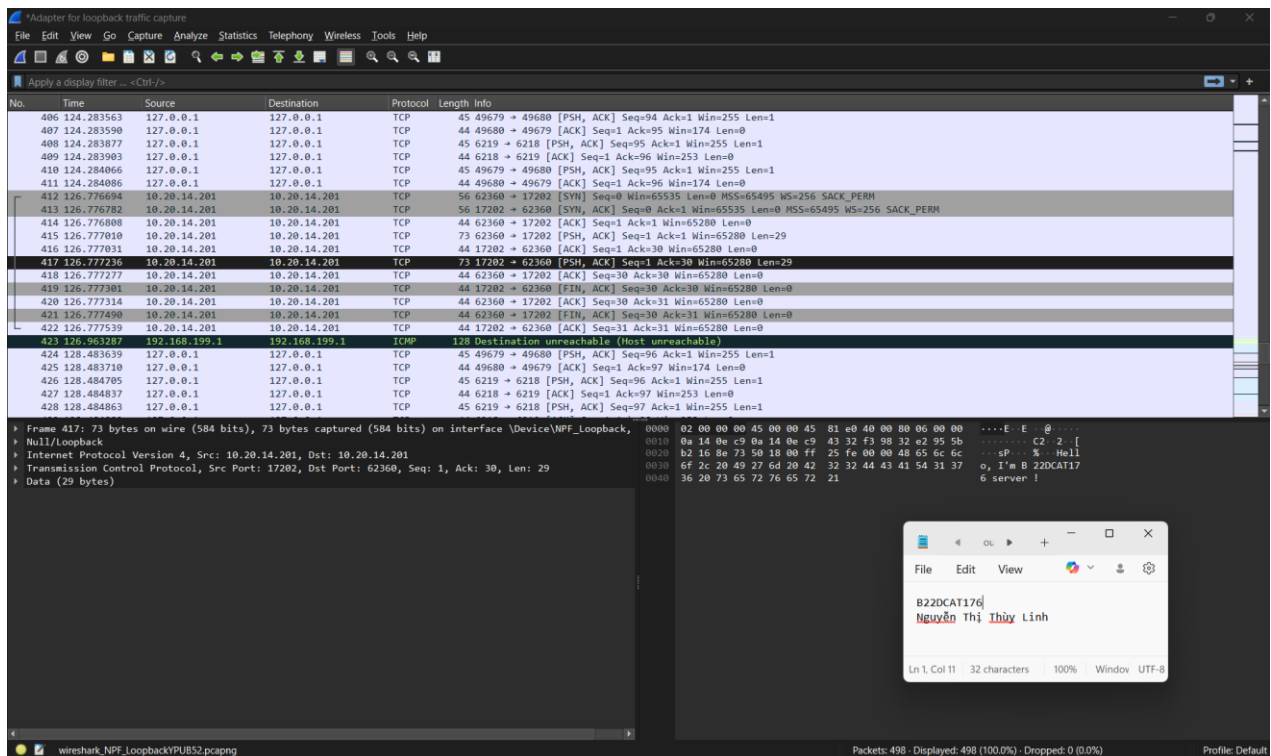
Hình ảnh 6: Thực hiện run file client.py

Mô tả:

* Client gửi thông điệp cá nhân hóa cho server: “Hello, I am B22DCAT176 client.”

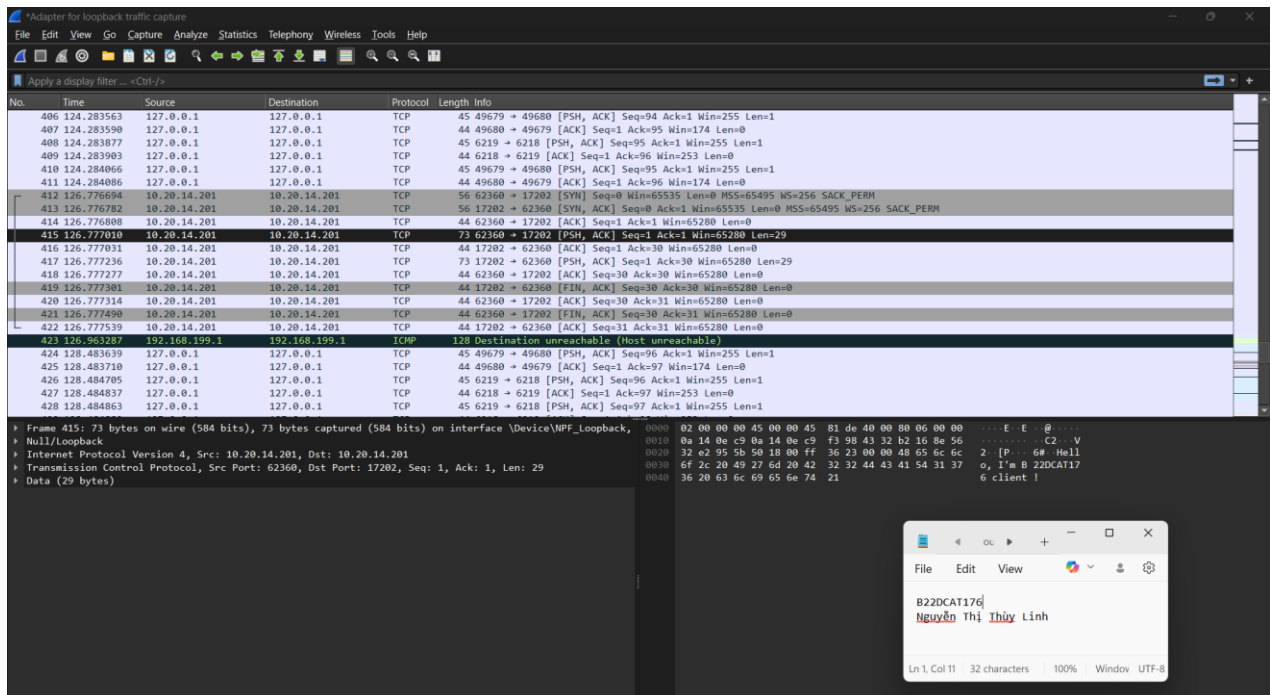
* Server nhận được hiển thị thông điệp nhận được và gửi lại thông điệp: server gửi lại “Hello, I am B22DCAT176 server”

- Sử dụng Wireshark để bắt các thông tin gửi từ client đến server và ngược lại
Bắt gói tin của server “Hello, I’m B22DCAT176 server”



Hình ảnh 7: Bắt gói tin server

Wireshark cũng bắt được gói tin mà Client gửi đến Server



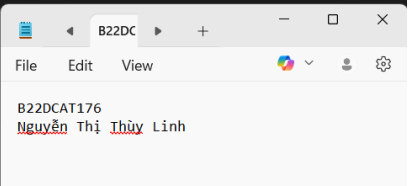
Hình ảnh 8: Bắt gói tin client

2. Trao đổi thông điệp giữa client và server và đảm bảo tính toàn vẹn của thông điệp khi trao đổi

Ta tiến hành sửa đổi lại code Client và Server

Code server:

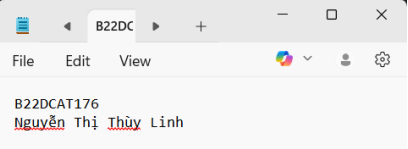
```
CodeSocket > server.py > start_server
1 import socket
2 import hashlib
3
4 def start_server():
5     print("Server started") # Báo hiệu server đã khởi động
6
7     host = socket.gethostname() # Lấy tên máy
8     port = 17202 # Cổng sử dụng
9
10    # Tạo socket TCP
11    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12    server_socket.bind((host, port)) # Gắn socket với cổng
13    server_socket.listen(1) # Lắng nghe kết nối
14    conn, addr = server_socket.accept() # Chấp nhận kết nối
15
16    key = "NguyễnThịThủyLinh-B22DCAT176" # Khóa bí mật dùng để hash
17
18    # Nhận dữ liệu từ client
19    data = conn.recv(1024).decode()
20    tmp_data = conn.recv(1024).decode()
21
22    print("Received from client:", data)
23    print("Received hashed data:", tmp_data)
24
25    # Tạo chuỗi hash bằng SHA-256 (data + key)
26    hashed_data = hashlib.sha256(data.encode("utf-8") + key.encode("utf-8")).hexdigest()
27    message = "Encrypted Successfully B22DCAT176!!!"
28
29    print("Sending to client:", message)
30    print("Hashed Data:", hashed_data)
31
32    # Kiểm tra tính toàn vẹn dữ liệu
33    if tmp_data != hashed_data:
34        message = "The received message has lost its integrity."
35
36    # Gửi phản hồi lại cho client
37    conn.send(message.encode())
38    conn.close() # Đóng kết nối
39
40    # Chạy server
41    start_server()
42
```



Hình ảnh 9: Sửa code lại Server.

Code Client.

```
CodeSocket > client.py > start_client
1 import socket
2 import hashlib
3
4 def start_client():
5     print("Client started") # Báo hiệu client đã khởi động
6
7     host = socket.gethostname() # Lấy tên máy để kết nối
8     port = 17202 # Cổng của server
9
10    # Tạo socket TCP
11    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12    client_socket.connect((host, port)) # Kết nối tới server
13
14    key = "NguyễnThịThủyLinh-B22DCAT176" # Khóa bí mật giống server
15    message = "Hello, I'm B22DCAT176 client!" # Thông điệp gửi
16
17    # Băm thông điệp + khóa bằng SHA-256
18    hashed_message = hashlib.sha256(message.encode("utf-8") + key.encode("utf-8")).hexdigest()
19
20    print("Sending to server:", message)
21    client_socket.send(message.encode()) # Gửi thông điệp
22
23    print("Hashed Message:", hashed_message)
24    client_socket.send(hashed_message.encode()) # Gửi bản băm
25
26    # Nhận phản hồi từ server
27    data = client_socket.recv(1024).decode()
28    print("Received from server:", data)
29
30    client_socket.close() # Đóng kết nối
31
32    # Chạy client
33    start_client()
34
```



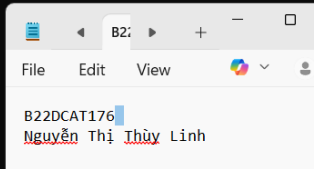
Hình ảnh 10: Sửa code lại Client.

- Mô tả: Server và Client sẽ đều sử dụng một giá trị key chung để tính toán giá trị băm của (thông điệp + key). Server sẽ gửi lại giá trị băm cho client, và client sẽ kiểm tra tính toàn vẹn của dữ liệu bằng cách so sánh giá trị băm nhận được từ server với giá trị băm của (thông điệp + key) mà nó đã tính toán trước đó.

Nếu tính toán vẹn không được đảm bảo, client sẽ in ra thông báo "*The received message has lost its integrity*". Ngược lại, nếu thông tin đảm bảo tính toán vẹn sẽ in ra thông báo "*Data integrity verified*"

Chạy chương trình với server.py

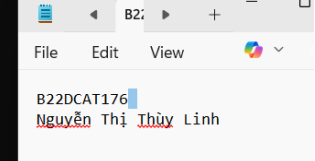
```
C:\Users\Admin\Documents\python\Python28tech\CodeSocket>py server.py
Server started
Received from client: Hello, I'm B22DCAT176 client!
Received hashed data: d7cbcd7720cf4ffeba0ced1b3f9c25fad2543d37ae5d7ad4edd7fd90c9934ae5
Sending to client: Encrypted Successfully B22DCAT176!!!
Hashed Data: d7cbcd7720cf4ffeba0ced1b3f9c25fad2543d37ae5d7ad4edd7fd90c9934ae5
C:\Users\Admin\Documents\python\Python28tech\CodeSocket>
```



Hình ảnh 11: Chạy file server.py

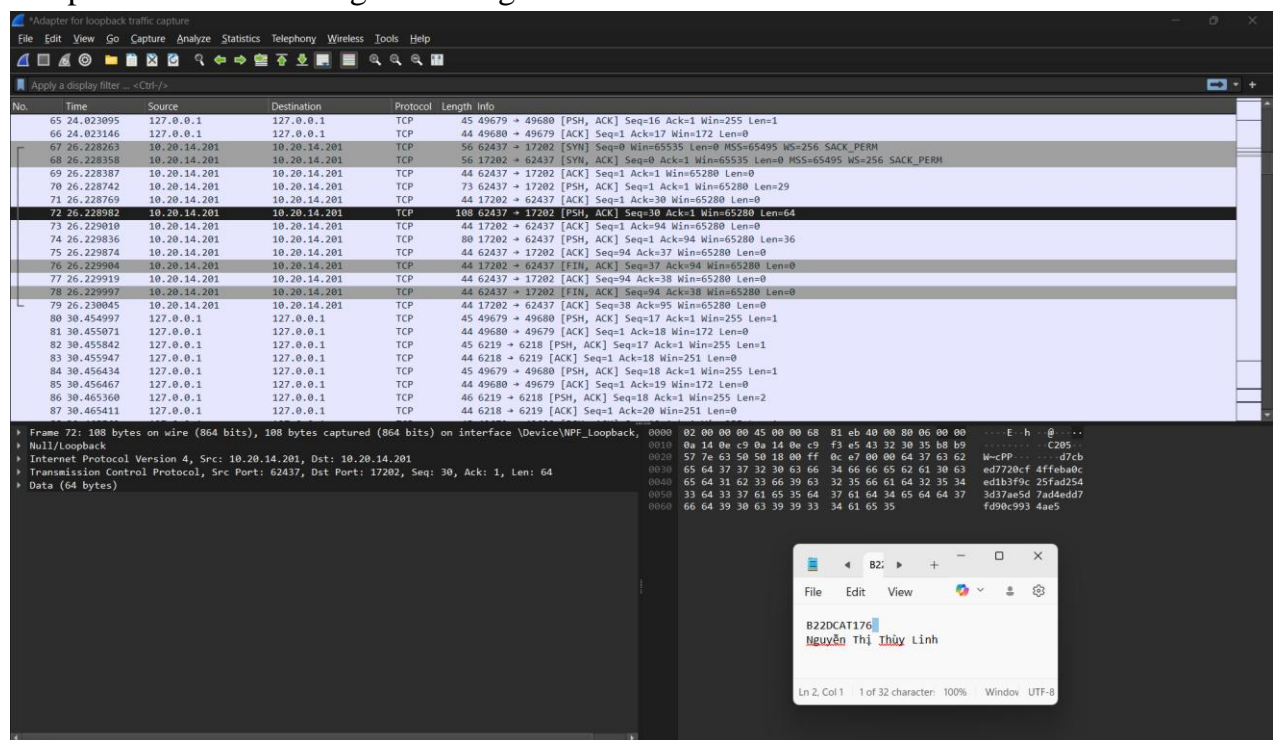
Chạy chương trình với client.py

```
C:\Users\Admin\Documents\python\Python28tech\CodeSocket>py client.py
Client started
Sending to server: Hello, I'm B22DCAT176 client!
Hashed Message: d7cbcd7720cf4ffeba0ced1b3f9c25fad2543d37ae5d7ad4edd7fd90c9934ae5
Received from server: Encrypted Successfully B22DCAT176!!!
C:\Users\Admin\Documents\python\Python28tech\CodeSocket>
```



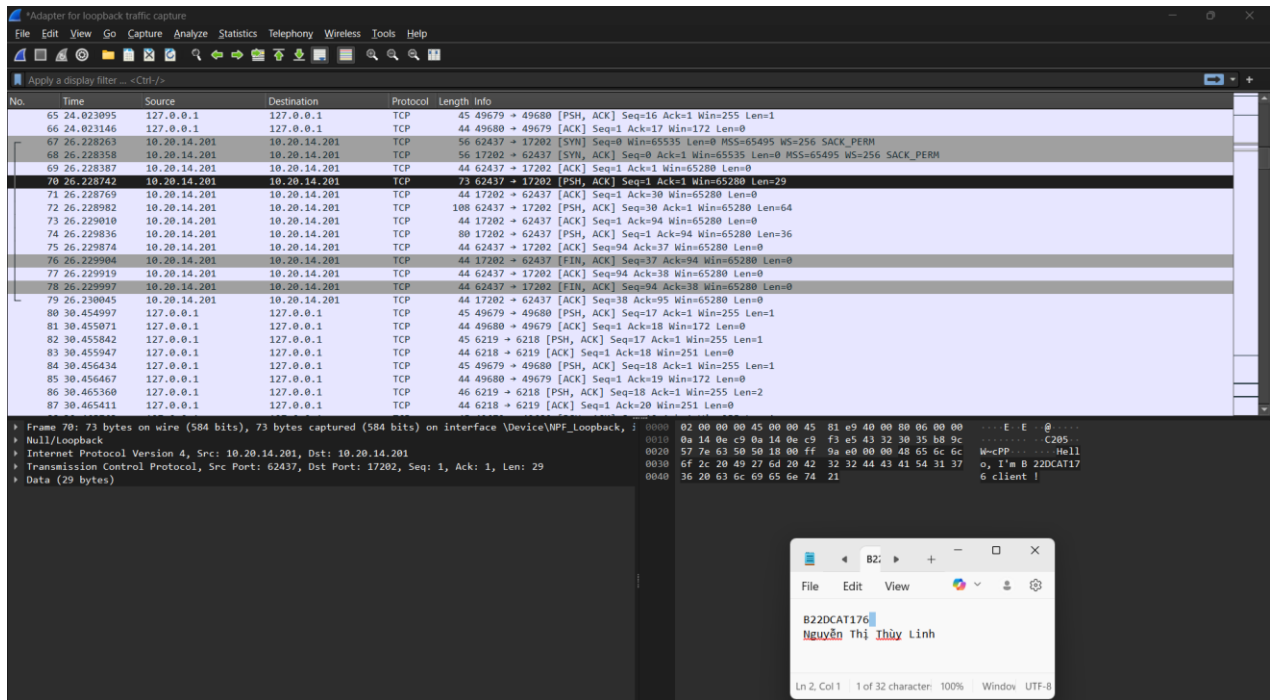
Hình ảnh 12: Chạy file client.py

-Tiếp theo tiến hành bắt gói tin bằng Wireshark.



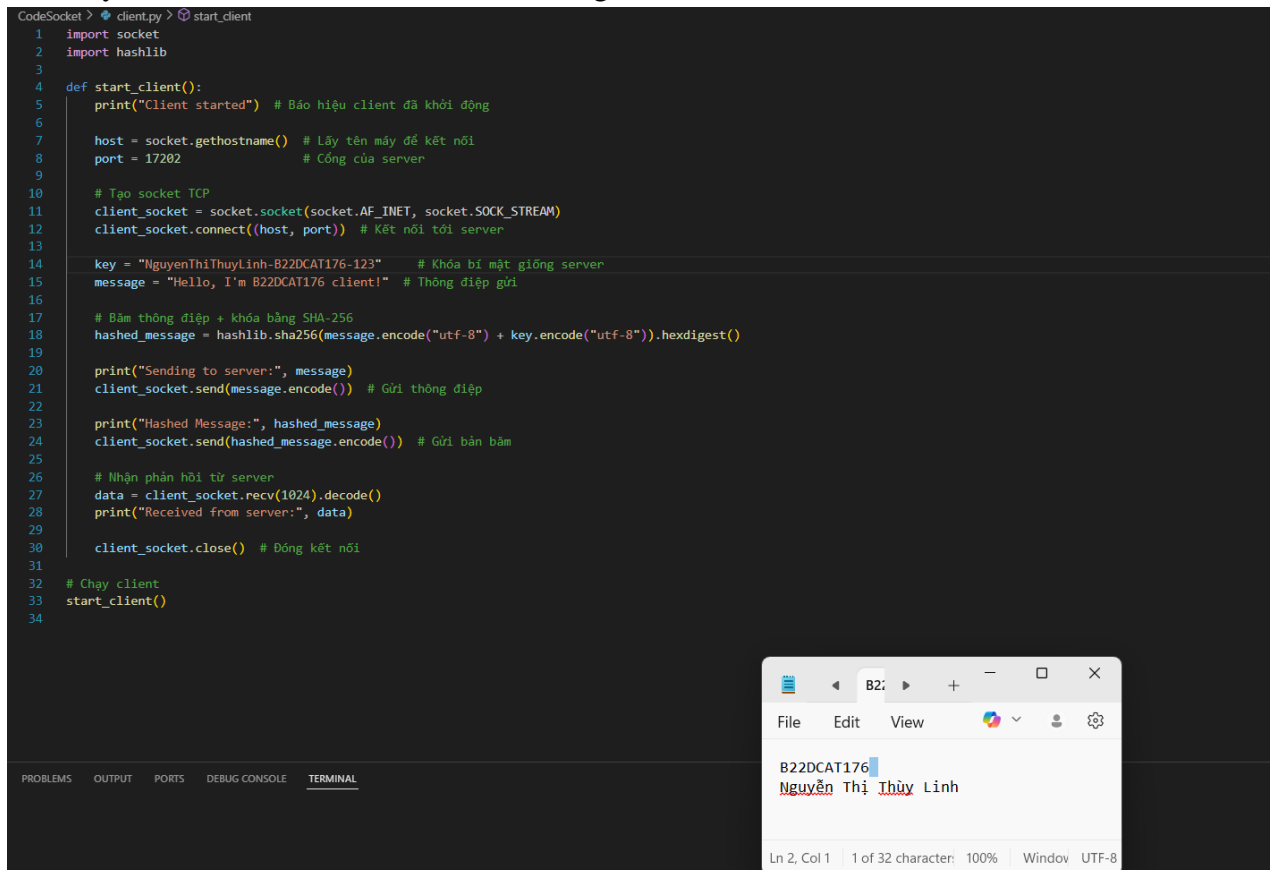
Hình ảnh 13: Bắt gói file giá trị băm

Bắt gói tin của client:



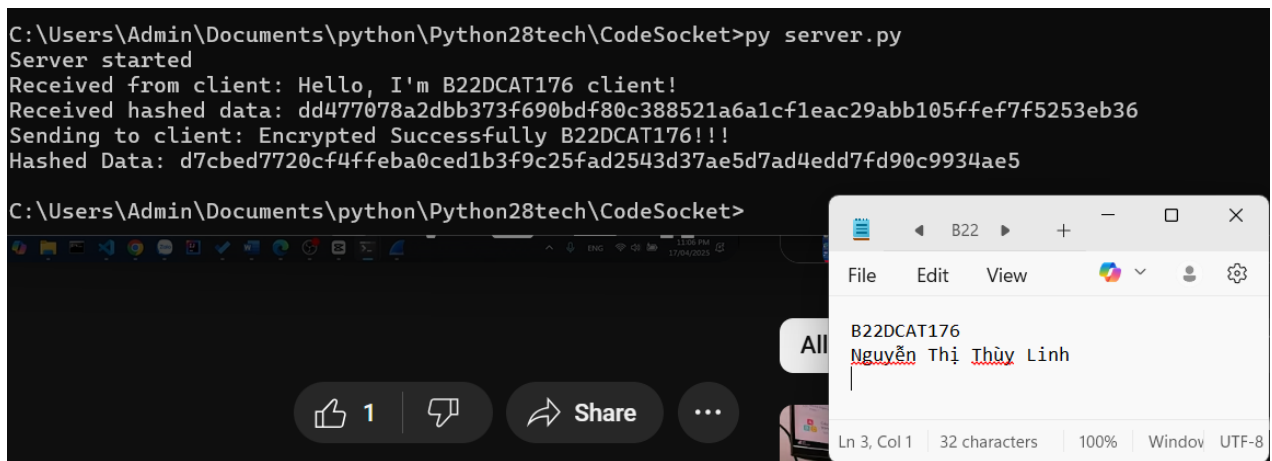
Hình ảnh 14: Bắt gói file client.

- Sửa key của client thêm -123 để thu được giá trị băm mới.



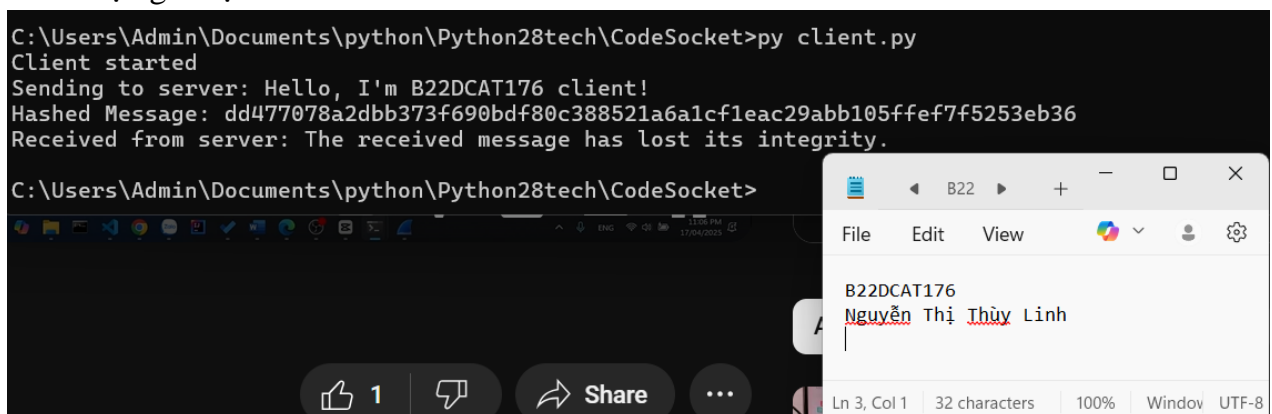
Hình ảnh 15: Tạo key bên client mới

Giá trị của server.py vẫn được giữ nguyên



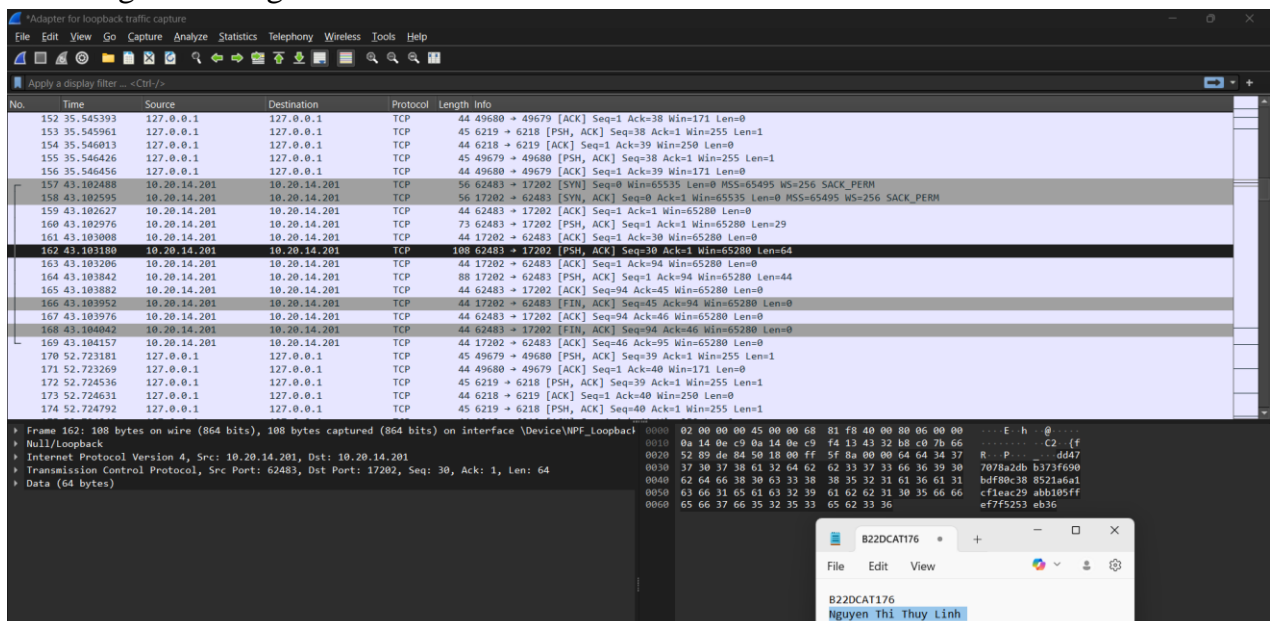
Hình ảnh 16: Thu được giá trị băm của server.

Thu được giá trị băm mới của client.



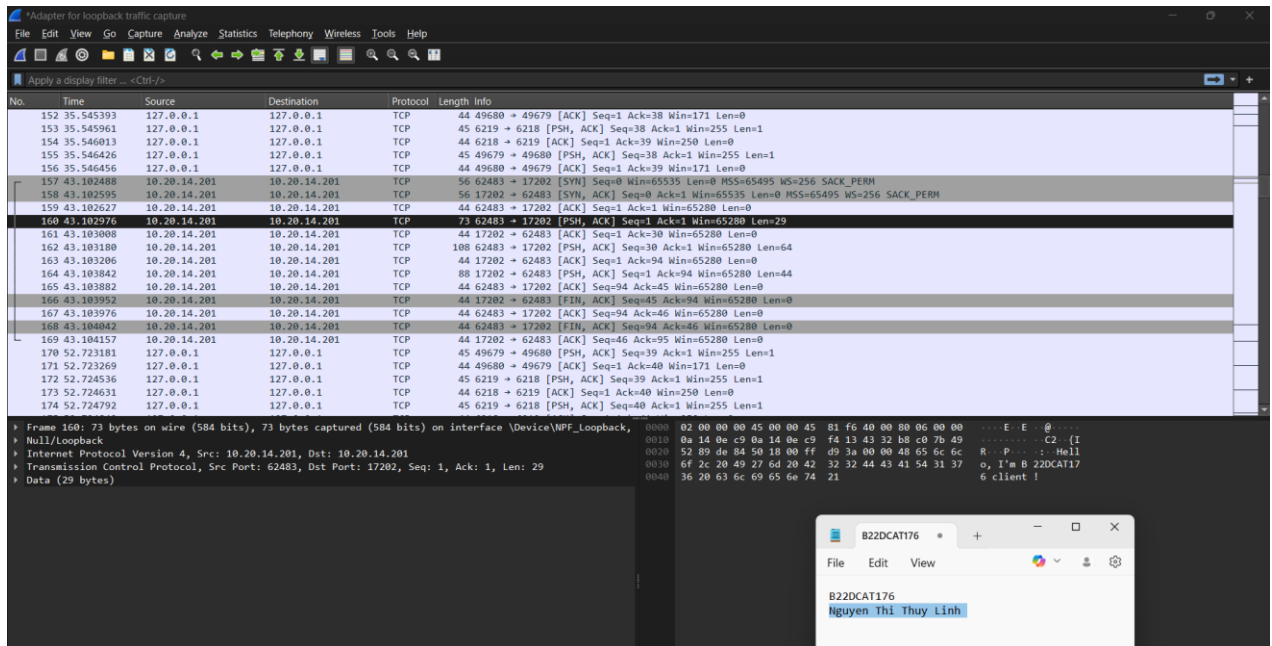
Hình ảnh 17: Thu được giá trị băm của client.

Bắt các gói tin bằng Wireshark.



Hình ảnh 18: Bắt được gói tin giá trị băm mới.

Bắt gói tin client.



Hình ảnh 19: Bắt gói tin client.

TÀI LIỆU THAM KHẢO

- [1] Phạm Hoàng Duy, Bài giảng Hệ điều hành Windows và Linux/Unix, Học viện Công Nghệ Bưu Chính Viễn Thông, 2016.
- [2] Tom Carpenter, Microsoft Windows Server Operating System Essentials, Sybex, 2011.
- [3] Wale Soyinka, Linux Administration A Beginners Guide, McGraw-Hill Osborne Media, 2012.