

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



Môn: Thực tập cơ sở

BÁO CÁO THỰC TẬP CƠ SỞ
Bài 15: Lập trình client/server để
trao đổi thông tin an toàn

Họ và tên giảng viên: PGS.TS.Đỗ Xuân Chợt
Họ và tên: Phạm Thanh Tùng
Mã sinh viên: B20DCAT171
Lớp: D20CQAT03-B
Số điện thoại: 0856915668

Hà Nội 2023

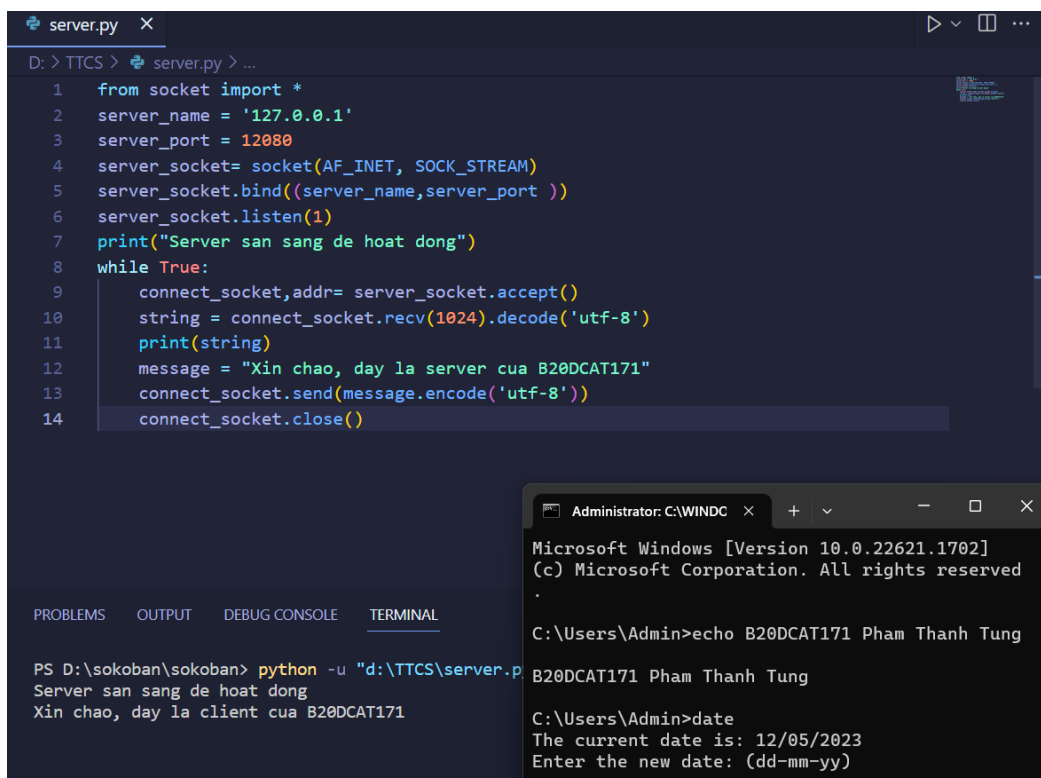
1. Nội dung lý thuyết:

- Socket là một điểm cuối của liên kết giao tiếp hai chiều giữa hai chương trình chạy trên mạng. Nghĩa là một socket được sử dụng để cho phép 1 chương trình giao tiếp với 1 chương trình khác.
- Các lớp Socket được sử dụng để tiến hành kết nối giữa client và server. Nó được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.
- Các lớp Socket được sử dụng để tiến hành kết nối giữa client và server. Nó được ràng buộc với một cổng port (thể hiện là một con số cụ thể) để các tầng TCP (TCP Layer) có thể định danh ứng dụng mà dữ liệu sẽ được gửi tới.
- Một đầu của kết nối ngang hàng của ứng dụng mạng phân tán dựa trên TCP/IP được mô tả bởi ổ cắm được xác định bởi:
 - o Địa chỉ Internet
 - o Giao thức giao tiếp: UDP, TCP
 - o Số cổng(Port)
- Các ứng dụng Socket thường là các ứng dụng C hoặc C ++ bằng cách sử dụng Socket API. Một số ngôn ngữ khác như Java, Python cũng cung cấp Socket API. Các ứng dụng Client/Server dựa trên Java, Python khai thác các dịch vụ Socket đó.

2. Nội dung thực hành:

2.1. Lập trình client và server với TCP socket:

Mã nguồn của server và chạy thành công server:

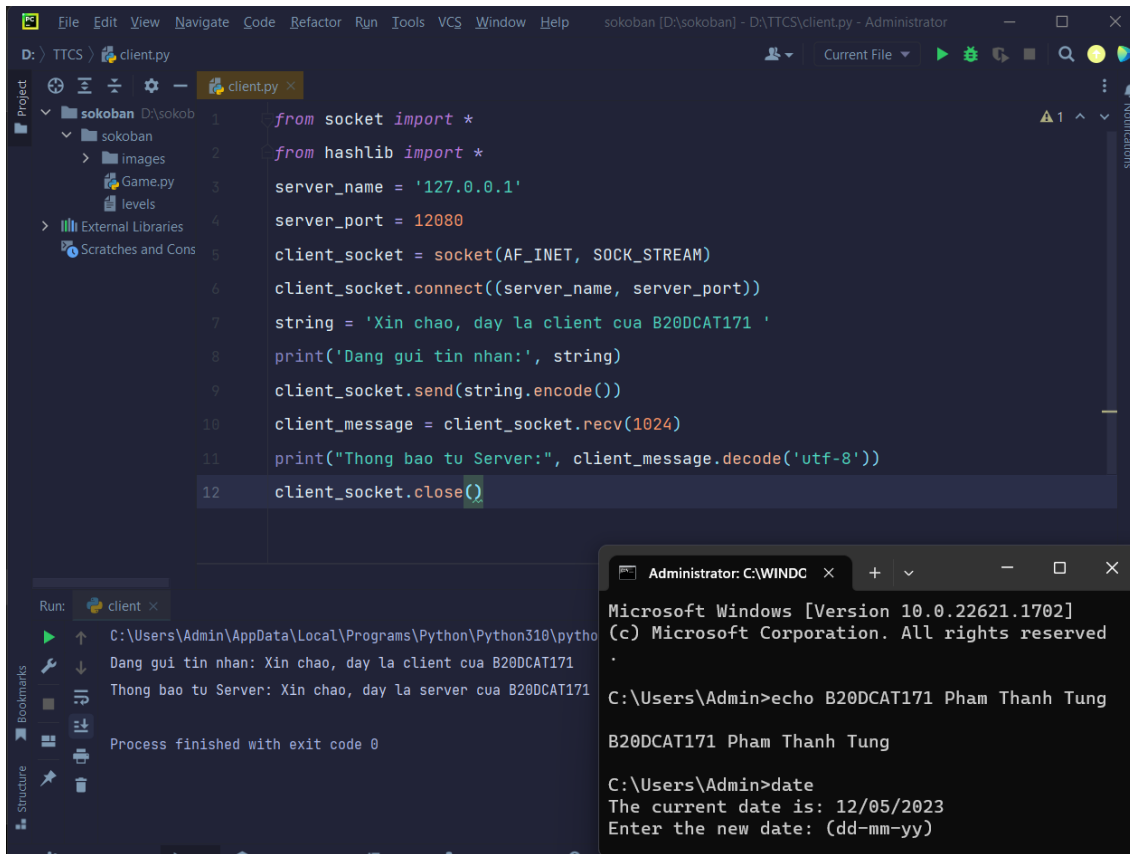


```
server.py X
D: > TTCS > server.py > ...
1 from socket import *
2 server_name = '127.0.0.1'
3 server_port = 12080
4 server_socket= socket(AF_INET, SOCK_STREAM)
5 server_socket.bind((server_name,server_port ))
6 server_socket.listen(1)
7 print("Server san sang de hoat dong")
8 while True:
9     connect_socket,addr= server_socket.accept()
10    string = connect_socket.recv(1024).decode('utf-8')
11    print(string)
12    message = "Xin chao, day la server cua B20DCAT171"
13    connect_socket.send(message.encode('utf-8'))
14    connect_socket.close()
```

```
Administrator: C:\WINDC x + - □ X
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved
.
C:\Users\Admin>echo B20DCAT171 Pham Thanh Tung
B20DCAT171 Pham Thanh Tung
C:\Users\Admin>date
The current date is: 12/05/2023
Enter the new date: (dd-mm-yy)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\sokoban\sokoban> python -u "d:\TTCS\server.p
Server san sang de hoat dong
Xin chao, day la client cua B20DCAT171
```

Mã nguồn của client và chạy thành công client:



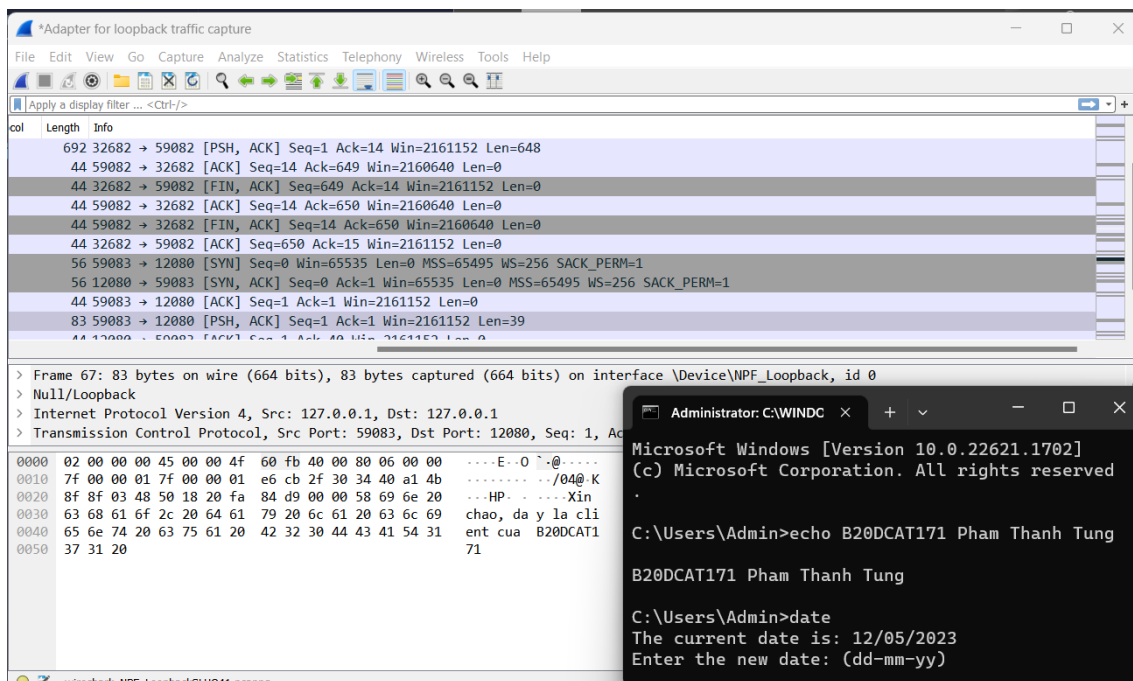
The screenshot shows a Python IDE with a project named 'sokoban'. The file 'client.py' is open, displaying the following code:

```
1 from socket import *
2 from hashlib import *
3 server_name = '127.0.0.1'
4 server_port = 12080
5 client_socket = socket(AF_INET, SOCK_STREAM)
6 client_socket.connect((server_name, server_port))
7 string = 'Xin chao, day la client cua B20DCAT171 '
8 print('Dang gui tin nhan:', string)
9 client_socket.send(string.encode())
10 client_message = client_socket.recv(1024)
11 print("Thong bao tu Server:", client_message.decode('utf-8'))
12 client_socket.close()
```

The Run window shows the output of the program:

```
Run: client x
C:\Users\Admin\AppData\Local\Programs\Python\Python310\python.exe
Dang gui tin nhan: Xin chao, day la client cua B20DCAT171
Thong bao tu Server: Xin chao, day la server cua B20DCAT171
Process finished with exit code 0
```

Bắt gói tin chương trình gửi đi bằng Wireshark:



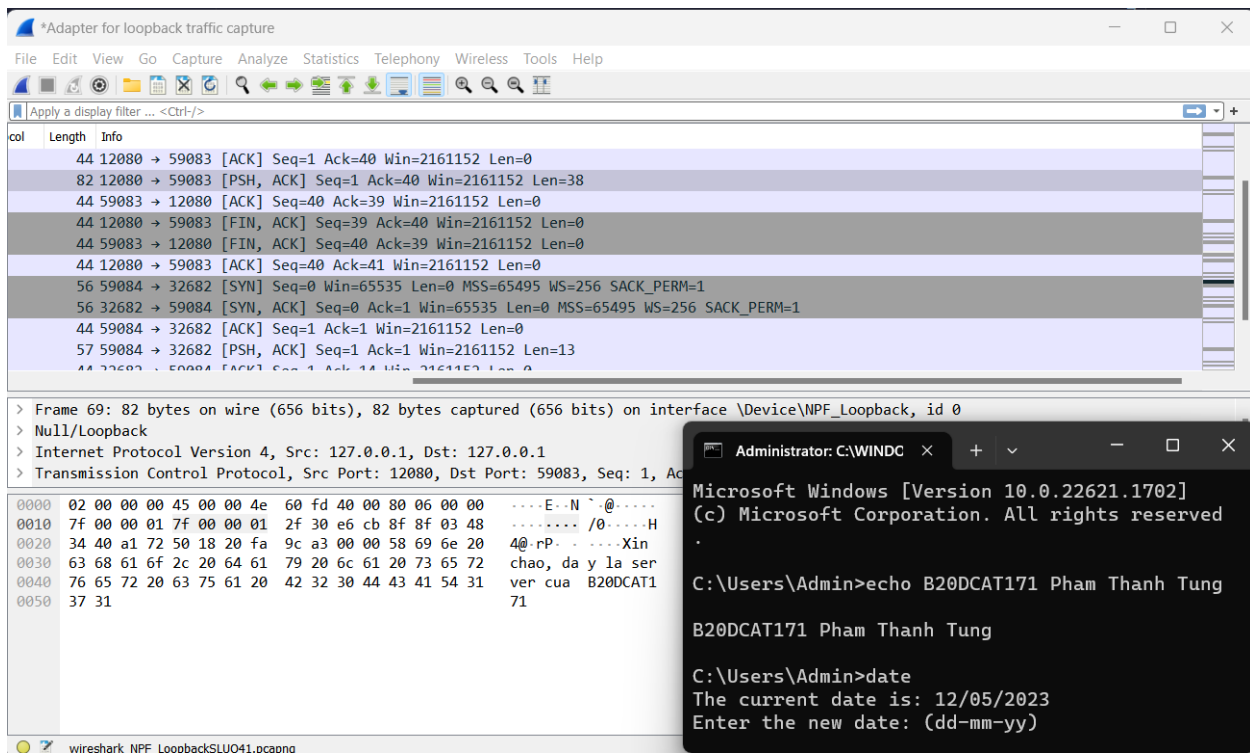
The screenshot shows Wireshark capturing network traffic on the interface \Device\NPF_{...}. The packet list shows several TCP segments. The packet details pane shows the selected packet (Frame 67) as an Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1, and Transmission Control Protocol, Src Port: 59083, Dst Port: 12080, Seq: 1, Ack: 1.

The packet bytes pane shows the raw data of the captured packet:

```
0000 02 00 00 00 45 00 00 4f 60 fb 40 00 80 06 00 00 ....E..0...
0010 7f 00 00 01 7f 00 00 01 e6 cb 2f 30 34 40 a1 4b .....:/04@.K
0020 8f 8f 03 48 50 18 20 fa 84 d9 00 00 58 69 6e 20 ...HP. ....Xin
0030 63 68 61 6f 2c 20 64 61 79 20 6c 61 20 63 6c 69 chao, da y la cli
0040 65 6e 74 20 63 75 61 20 42 32 30 44 43 41 54 31 ent cua B20DCAT1
0050 37 31 20 71
```

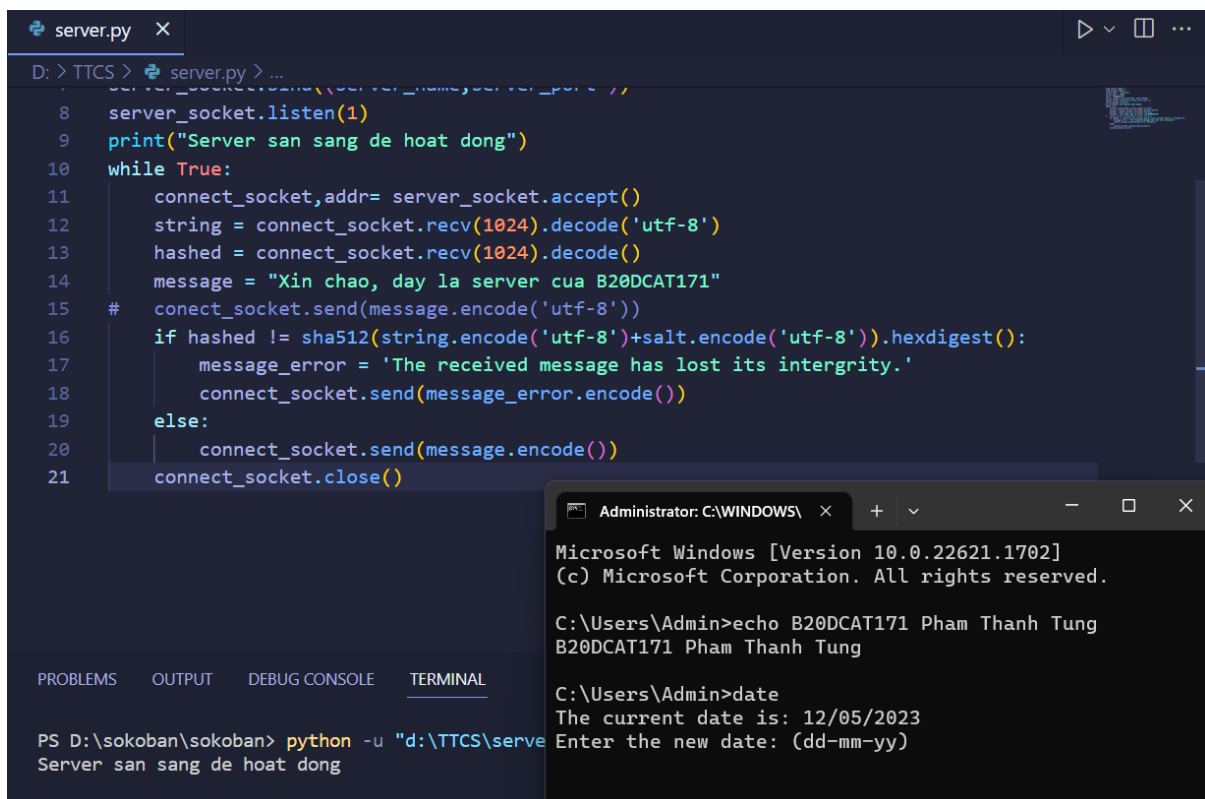
The terminal window shows the output of the program:

```
Administrator: C:\WINDOWS
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved
.
C:\Users\Admin>echo B20DCAT171 Pham Thanh Tung
B20DCAT171 Pham Thanh Tung
C:\Users\Admin>date
The current date is: 12/05/2023
Enter the new date: (dd-mm-yy)
```

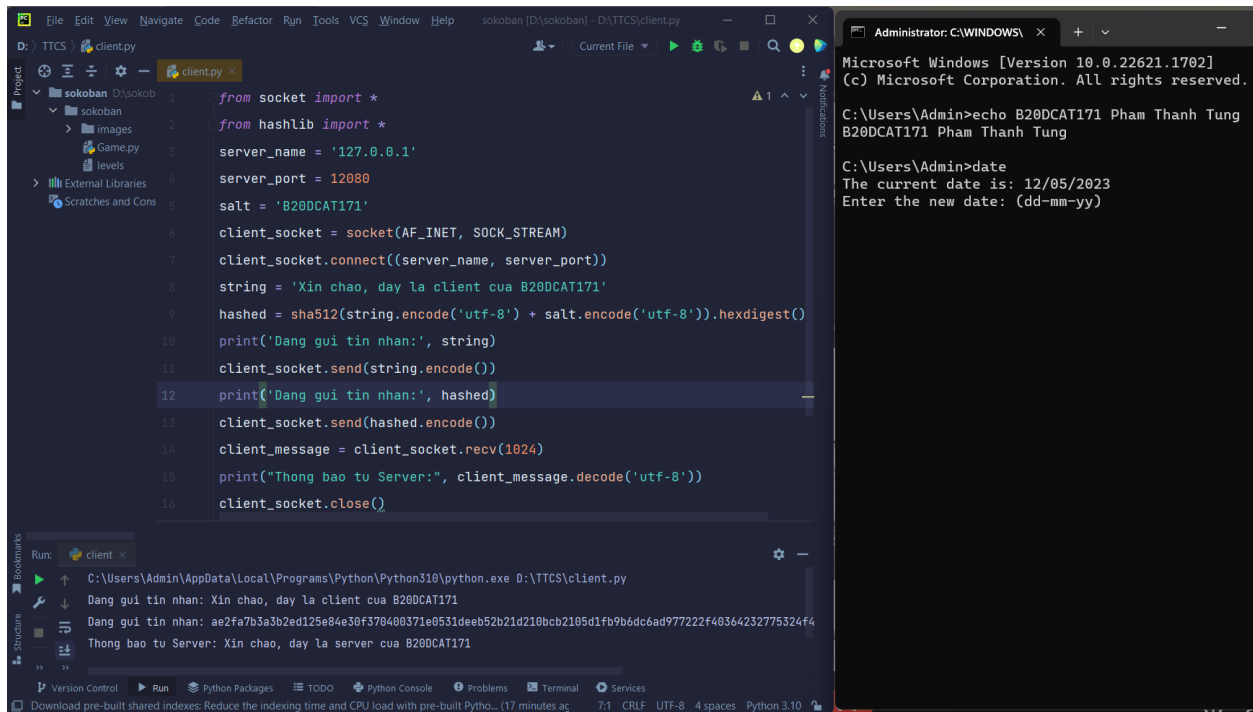


2.2. Trao đổi thông điệp giữa client và server và đảm bảo tính toàn vẹn của thông điệp khi trao đổi

Mã nguồn của server và chạy thành công server:



Mã nguồn của client và chạy thành công client:



The screenshot shows a Python IDE with a file named `client.py` and a terminal window. The code in `client.py` is as follows:

```
1 from socket import *
2 from hashlib import *
3 server_name = '127.0.0.1'
4 server_port = 12080
5 salt = 'B20DCAT171'
6 client_socket = socket(AF_INET, SOCK_STREAM)
7 client_socket.connect((server_name, server_port))
8 string = 'Xin chào, đây là client của B20DCAT171'
9 hashed = sha512(string.encode('utf-8') + salt.encode('utf-8')).hexdigest()
10 print('Đang gửi tin nhắn:', string)
11 client_socket.send(string.encode())
12 print('Đang gửi tin nhắn:', hashed)
13 client_socket.send(hashed.encode())
14 client_message = client_socket.recv(1024)
15 print("Thông báo từ Server:", client_message.decode('utf-8'))
16 client_socket.close()
```

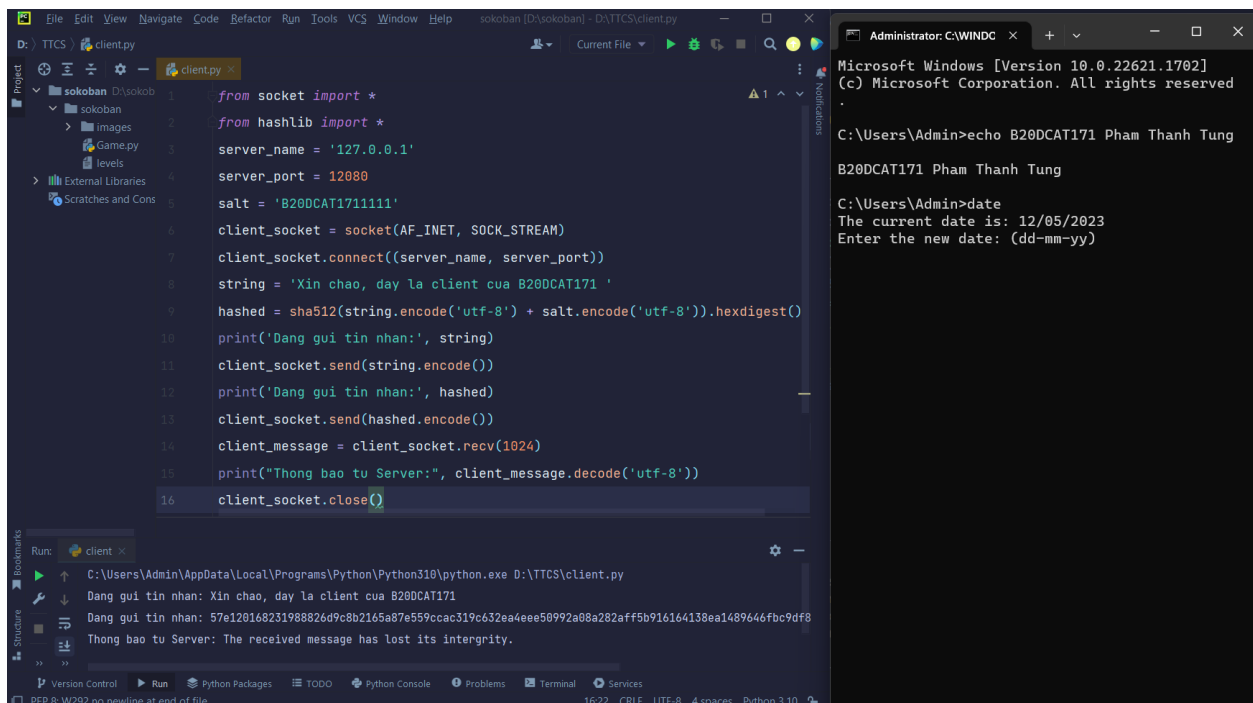
The terminal window shows the following output:

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>echo B20DCAT171 Pham Thanh Tung
B20DCAT171 Pham Thanh Tung

C:\Users\Admin>date
The current date is: 12/05/2023
Enter the new date: (dd-mm-yy)
```

Thay đổi key của client, chạy lại client, và nhận được thông báo “The received message has lost its integrity.”:



The screenshot shows the same Python IDE with a modified `client.py` file. The code is identical to the previous one, except for the salt value, which has been changed to `'B20DCAT1711111'` on line 5. The terminal window shows the following output:

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>echo B20DCAT171 Pham Thanh Tung
B20DCAT171 Pham Thanh Tung

C:\Users\Admin>date
The current date is: 12/05/2023
Enter the new date: (dd-mm-yy)
```

The IDE's Run console shows the following output:

```
Đang gửi tin nhắn: Xin chào, đây là client của B20DCAT171
Đang gửi tin nhắn: 57e12016823198826d9c8b2165a87e559ccac319c632ea4ee50992a08a282aff5b9164138ea1489646fbc9df8
Thông báo từ Server: The received message has lost its integrity.
```

Gói tin chứa mã hash từ client:

The screenshot shows a Wireshark capture of a TCP connection from 127.0.0.1 to 127.0.0.1. The connection is established with a SYN exchange, followed by an ACK. The client then sends a PSH, ACK packet with a payload containing a command. The server responds with a FIN, ACK packet, indicating the end of the connection. A packet details pane for Frame 46 shows the command 'echo B20DCAT171 Pham Thanh Tung' and the date '12/05/2023'.

No.	Time	Source	Destination	Protocol	Length	Info
311	21.567321	127.0.0.1	127.0.0.1	TCP	44	12080 → 58481 [ACK] Seq=1 Ack=168 Win=2161152 Len=0
312	21.567363	127.0.0.1	127.0.0.1	TCP	89	12080 → 58481 [PSH, ACK] Seq=1 Ack=168 Win=2161152 Len=45
313	21.567377	127.0.0.1	127.0.0.1	TCP	44	58481 → 12080 [ACK] Seq=168 Ack=46 Win=2161152 Len=0
314	21.567394	127.0.0.1	127.0.0.1	TCP	44	12080 → 58481 [FIN, ACK] Seq=46 Ack=168 Win=2161152 Len=0
315	21.567403	127.0.0.1	127.0.0.1	TCP	44	58481 → 12080 [FIN, ACK] Seq=168 Ack=46 Win=2161152 Len=0
316	21.567414	127.0.0.1	127.0.0.1	TCP	44	12080 → 58481 [ACK] Seq=47 Ack=169 Win=2161152 Len=0
317	21.779008	127.0.0.1	127.0.0.1	TCP	56	58482 → 32682 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256
318	21.779038	127.0.0.1	127.0.0.1	TCP	56	32682 → 58482 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
319	21.779062	127.0.0.1	127.0.0.1	TCP	44	58482 → 32682 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
320	21.779094	127.0.0.1	127.0.0.1	TCP	57	58482 → 32682 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=13

Bắt các gói tin bằng Wireshark: Gói tin thông báo “The received message has lost its integrity.” từ server:

The screenshot shows a Wireshark capture of a TCP connection from 127.0.0.1 to 127.0.0.1. The connection is established with a SYN exchange, followed by an ACK. The client then sends a PSH, ACK packet with a payload containing a command. The server responds with a FIN, ACK packet, indicating the end of the connection. A packet details pane for Frame 312 shows the command 'echo B20DCAT171 Pham Thanh Tung' and the date '12/05/2023'. The packet data pane shows the command and the date, with a note indicating that the received message has lost its integrity.

No.	Time	Source	Destination	Protocol	Length	Info
311	21.567321	127.0.0.1	127.0.0.1	TCP	44	12080 → 58481 [ACK] Seq=1 Ack=168 Win=2161152 Len=0
312	21.567363	127.0.0.1	127.0.0.1	TCP	89	12080 → 58481 [PSH, ACK] Seq=1 Ack=168 Win=2161152 Len=45
313	21.567377	127.0.0.1	127.0.0.1	TCP	44	58481 → 12080 [ACK] Seq=168 Ack=46 Win=2161152 Len=0
314	21.567394	127.0.0.1	127.0.0.1	TCP	44	12080 → 58481 [FIN, ACK] Seq=46 Ack=168 Win=2161152 Len=0
315	21.567403	127.0.0.1	127.0.0.1	TCP	44	58481 → 12080 [FIN, ACK] Seq=168 Ack=46 Win=2161152 Len=0
316	21.567414	127.0.0.1	127.0.0.1	TCP	44	12080 → 58481 [ACK] Seq=47 Ack=169 Win=2161152 Len=0
317	21.779008	127.0.0.1	127.0.0.1	TCP	56	58482 → 32682 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256
318	21.779038	127.0.0.1	127.0.0.1	TCP	56	32682 → 58482 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM=1
319	21.779062	127.0.0.1	127.0.0.1	TCP	44	58482 → 32682 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
320	21.779094	127.0.0.1	127.0.0.1	TCP	57	58482 → 32682 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=13