

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA AN TOÀN THÔNG TIN

BỘ MÔN THỰC TẬP CƠ SỞ



BÀI 15:
Lập trình client/server để
trao đổi thông tin an toàn

Giảng viên : Nguyễn Ngọc Diệp

Sinh viên : Nguyễn Đức Anh

Mã sinh viên : B21DCAT031

Hệ : Đại học chính quy

Hà Nội, 4/2024

Table of Contents

1. Mục đích	3
2. Nội dung thực hành	3
2.1 Tìm hiểu lý thuyết	3
2.2 Tài liệu tham khảo	3
2.3 Chuẩn bị môi trường	3
2.4 Các bước thực hiện	3
a. Lập trình client và server với TCP socket.....	3
b. Trao đổi thông điệp giữa client và server và đảm bảo tính toàn vẹn của thông điệp khi trao đổi	5
3. Kết quả đạt được	9

1. 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.

2. Nội dung thực hành

2.1 Tìm hiểu lý thuyết

a. Socket là gì?

Đây chính 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.

b. Cơ chế hoạt động của Socket là gì?

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 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 tới 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 tin 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.

2.2 Tài liệu tham khảo

- Chapter 2: Application Layer V8.1 (9/2020) http://gaia.cs.umass.edu/kurose_ross/ppt.php
- Chương 2: Tầng ứng dụng (Bộ môn Mạng máy tính)
https://drive.google.com/file/d/1ma8X7qtdoIOMKCmMZ9yTVPboam0hIweZ/view?usp=drive_link

2.3 Chuẩn bị môi trường

- Môi trường Python hoặc Java để chạy được ứng dụng client/server đã lập trình.
- Phần mềm Wireshark

2.4 Các bước thực hiện

a. Lập trình client và server với TCP socket

- Lập trình Client:

```
server.py  client.py  X
client.py > ...
1  import socket
2  import hashlib
3  Host = "127.0.0.1"
4  Server_port = 30031
5  Format="utf8"
6
7  client = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
8  client.connect((Host,Server_port))
9
10 try:
11     while True:
12         data=input("Client gui toi server: ")
13         client.sendall(data.encode(Format))
14         if data == "quit": break
15         data_server= client.recv(1024).decode(Format)
16         print("Nhan tu server: ",data_server)
17 finally:
18     client.close()
```

```
C:\Windows\system32\cmd.exe - date
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>echo B21DCAT031 Nguyen Duc Anh && date
B21DCAT031 Nguyen Duc Anh
The current date is: 15/04/2024
Enter the new date: (dd-mm-yy)
```

-Lập trình Server:

```
server.py  X  client.py
server.py > ...
1  import socket
2
3  Host = "127.0.0.1" #loopback
4  Server_port = 30031
5  Format="utf8"
6  server=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
7
8  server.bind((Host,Server_port))
9  server.listen(1)
10 conn, addr = server.accept()
11
12 try:
13     print("Client address:",addr)
14     while True:
15         data_client=conn.recv(1024).decode(Format)
16         print("Nhan tu client: ",data_client)
17         if data_client == "quit": break
18         data=input("Server gui goi tin toi client: ")
19         conn.sendall(data.encode(Format))
20 finally:
21     server.close()
```

```
C:\Windows\system32\cmd.exe - date
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>echo B21DCAT031 Nguyen Duc Anh && date
B21DCAT031 Nguyen Duc Anh
The current date is: 15/04/2024
Enter the new date: (dd-mm-yy)
```

- Chạy chương trình:


```
server2.py client2.py X
client2.py > ...
1 import socket
2 import hashlib
3
4 Host = '127.0.0.1'
5 Server_port = 30031
6
7 def calculate_hash(data, key):
8     data_with_key = data + key
9     h = hashlib.sha1(data_with_key.encode())
10    return h.hexdigest()
11
12 def verify_integrity(data, key, received_hash):
13     calculated_hash = calculate_hash(data, key)
14     if calculated_hash == received_hash:
15         return True
16     else:
17         return False
18
19 client = socket.socket()
20 client.connect((Host, Server_port))
21
22 key = "AT031ismykey"
23
24 while True:
25     data = input("Thong diep duoc gui di la: ")
26
C:\Windows\system32\cmd.exe - date
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>echo B21DCAT031 NguyenDucAnh && date
B21DCAT031 NguyenDucAnh
The current date is: 17/04/2024
Enter the new date: (dd-mm-yy)

server2.py client2.py X
client2.py > ...
24 while True:
25     data = input("Thong diep duoc gui di la: ")
26
27     data_hash = calculate_hash(data, key)
28     print(data_hash)
29     client.send(data_hash.encode())
30     received_hash = client.recv(1024).decode()
31     if verify_integrity(data_hash, key, received_hash):
32         print("Data integrity verified")
33     else:
34         print("The received message has lost its integrity.")
35
36     key = input("Enter new key: ")
37     data_with_new_key = data + key
38
39     data_hash_new = calculate_hash(data_with_new_key, key)
40     client.send(data_hash_new.encode())
41
42     received_hash_new = client.recv(1024).decode()
43     if verify_integrity(data, key, received_hash_new):
44         print("Data integrity verified with new key")
45     else:
46         print("The received message has lost its integrity with new k
47         break
48     client.close()
49
0 0 0 0 0
```

Code Server:

```

server2.py > ...
1  import socket
2  import hashlib
3  Host = '127.0.0.1' #loopback
4  Server_port = 30031
5  key="AT031ismykey"
6  server=socket.socket()
7  server.bind((Host,Server_port))
8  server.listen(1)
9  print("Waiting...")
10 conn, addr = server.accept()
11 print("Client address: ",addr)
12
13 while True:
14     data=conn.recv(1024).decode()
15     if not data:
16         break
17     print("From connected user: "+str(data))
18     message=data+key
19     h=hashlib.sha1(message.encode())
20     result=h.hexdigest()
21     conn.send(result.encode())
22 server.close()

```

```

C:\Windows\system32\cmd.exe - date
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>echo B21DCAT031 NguyenDucAnh && date
B21DCAT031 NguyenDucAnh
The current date is: 16/04/2024
Enter the new date: (dd-mm-yy)

```

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 toàn 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 toàn vẹn sẽ in ra thông báo "Data integrity verified"

Chạy chương trình:

```

client2.py > calculate_hash
1  import socket
2  import hashlib
3
4  Host = '127.0.0.1'
5  Server_port = 30031
6
7  def calculate_hash(data, key):
8      data_with_key = data + key
9      h = hashlib.sha1(data_with_key.encode())
10     return h.hexdigest()
11
12 def verify_integrity(data, key, received_hash):
13     calculated_hash = calculate_hash(data, key)
14     if calculated_hash == received_hash:

```

```

PS D:\TTCS> & C:/Users/admin/AppData/Local/Programs/Python/Python312/python.exe d:/TTCS/server2.py
Waiting...
Client address: ('127.0.0.1', 65197)
From connected user: ec3907af9455930f4548f5843ef6c946c31ab6be

```

```

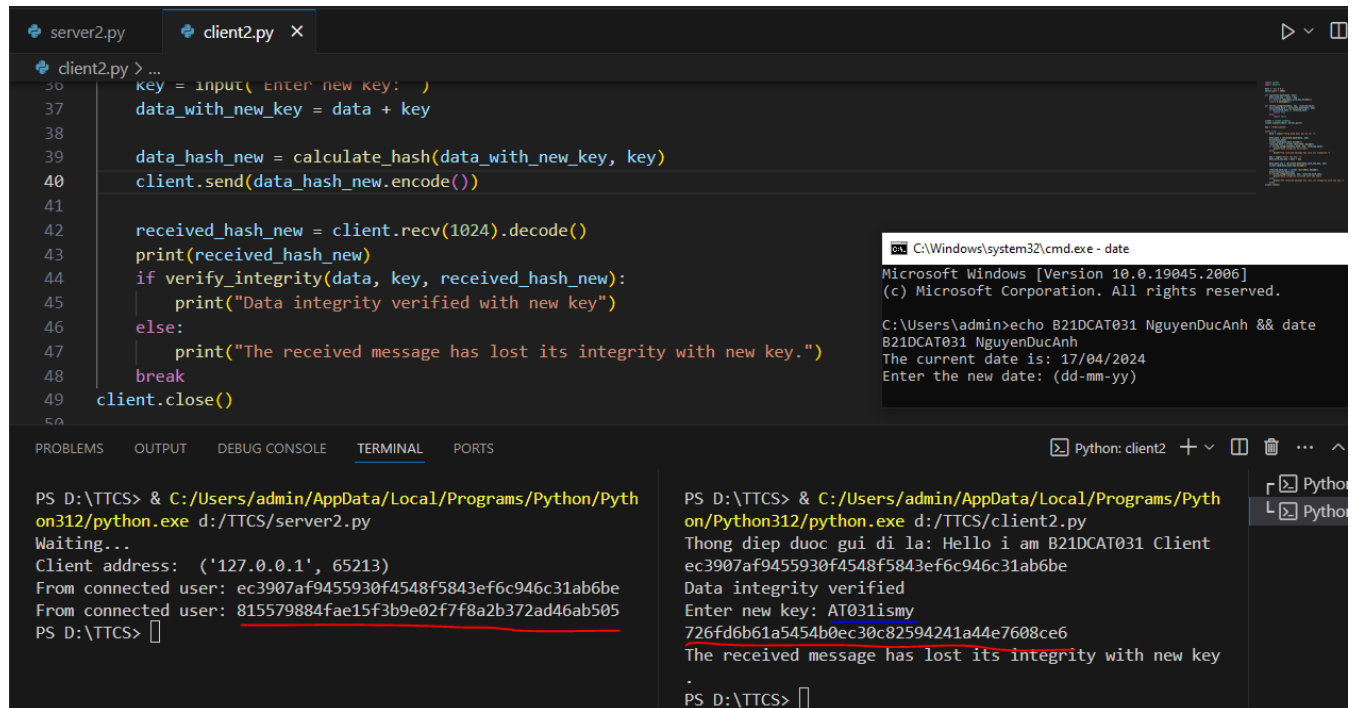
PS D:\TTCS> & C:/Users/admin/AppData/Local/Programs/Python/Python312/python.exe d:/TTCS/client2.py
Thong diep duoc gui di la: Hello i am B21DCAT031 Client
ec3907af9455930f4548f5843ef6c946c31ab6be
Data integrity verified
Enter new key: 

```

Sau đó, client yêu cầu người dùng nhập giá trị key mới và tính toán lại giá trị băm của (thông điệp + key mới). Nó gửi giá trị băm mới đến server và kiểm tra tính toàn vẹn của dữ liệu với key mới.

Nếu tính toán vẹn được đảm bảo, client sẽ in ra thông báo "Data integrity verified with new key", nếu không, client sẽ in ra thông báo "The received message has lost its integrity with new key." Với tính năng này, client có thể thay đổi giá trị key và kiểm tra tính toàn vẹn của dữ liệu khi key được thay đổi. Điều này giúp đảm bảo tính toàn vẹn của dữ liệu trong trường hợp giá trị key bị lộ hoặc thay đổi.

Trong trường hợp dưới đây, ta sẽ thử nhập một key khác với key có ở Server để kiểm tra



The screenshot shows a VS Code editor with two tabs: `server2.py` and `client2.py`. The `client2.py` tab is active, displaying the following Python code:

```
36 key = input( Enter new key: )
37 data_with_new_key = data + key
38
39 data_hash_new = calculate_hash(data_with_new_key, key)
40 client.send(data_hash_new.encode())
41
42 received_hash_new = client.recv(1024).decode()
43 print(received_hash_new)
44 if verify_integrity(data, key, received_hash_new):
45     print("Data integrity verified with new key")
46 else:
47     print("The received message has lost its integrity with new key.")
48     break
49 client.close()
```

Below the code editor, the `TERMINAL` panel shows the output of running the client script. The terminal is split into two panes. The left pane shows the command prompt output:

```
PS D:\TTCS> & C:/Users/admin/AppData/Local/Programs/Python/Python312/python.exe d:/TTCS/server2.py
Waiting...
Client address: ('127.0.0.1', 65213)
From connected user: ec3907af9455930f4548f5843ef6c946c31ab6be
From connected user: 815579884fae15f3b9e02f7f8a2b372ad46ab505
PS D:\TTCS>
```

The right pane shows the output of running the client script:

```
PS D:\TTCS> & C:/Users/admin/AppData/Local/Programs/Python/Python312/python.exe d:/TTCS/client2.py
Thong diep duoc gui di la: Hello i am B21DCAT031 Client
ec3907af9455930f4548f5843ef6c946c31ab6be
Data integrity verified
Enter new key: AT031ismy
726fd6b61a5454b0ec30c82594241a44e7608ce6
The received message has lost its integrity with new key
PS D:\TTCS>
```

On the right side of the terminal, there is a small window showing the output of the `date` command:

```
C:\Windows\system32\cmd.exe - date
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>echo B21DCAT031 NguyenDucAnh && date
B21DCAT031 NguyenDucAnh
The current date is: 17/04/2024
Enter the new date: (dd-mm-yy)
```

Bắt được các gói tin bằng Wireshark:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	65226 → 30031 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
2	0.000049	127.0.0.1	127.0.0.1	TCP	56	30031 → 65226 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
3	0.000118	127.0.0.1	127.0.0.1	TCP	44	65226 → 30031 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	7.690688	127.0.0.1	127.0.0.1	TCP	84	65226 → 30031 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=40
5	7.690741	127.0.0.1	127.0.0.1	TCP	44	30031 → 65226 [ACK] Seq=1 Ack=41 Win=2619648 Len=0
6	7.691016	127.0.0.1	127.0.0.1	TCP	84	30031 → 65226 [PSH, ACK] Seq=1 Ack=41 Win=2619648 Len=40
7	7.691026	127.0.0.1	127.0.0.1	TCP	44	65226 → 30031 [ACK] Seq=41 Ack=41 Win=2619648 Len=0
8	12.536974	127.0.0.1	127.0.0.1	TCP	84	65226 → 30031 [PSH, ACK] Seq=41 Ack=41 Win=2619648 Len=40
9	12.537003	127.0.0.1	127.0.0.1	TCP	44	30031 → 65226 [ACK] Seq=41 Ack=81 Win=2619648 Len=0
10	12.537231	127.0.0.1	127.0.0.1	TCP	84	30031 → 65226 [PSH, ACK] Seq=41 Ack=81 Win=2619648 Len=40
11	12.537247	127.0.0.1	127.0.0.1	TCP	44	65226 → 30031 [ACK] Seq=81 Ack=81 Win=2619648 Len=0
12	12.537485	127.0.0.1	127.0.0.1	TCP	44	65226 → 30031 [FIN, ACK] Seq=81 Ack=81 Win=2619648 Len=0
13	12.537515	127.0.0.1	127.0.0.1	TCP	44	30031 → 65226 [ACK] Seq=81 Ack=82 Win=2619648 Len=0
14	12.538758	127.0.0.1	127.0.0.1	TCP	44	30031 → 65226 [FIN, ACK] Seq=81 Ack=82 Win=2619648 Len=0
15	12.538813	127.0.0.1	127.0.0.1	TCP	44	65226 → 30031 [ACK] Seq=82 Ack=82 Win=2619648 Len=0

> Frame 7: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface 0
 > Null/loopback
 > Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 > Transmission Control Protocol, Src Port: 65226, Dst Port: 30031, Seq: 65226, Win: 0, Len: 0

```
0000  02 00 00 00 45 00 00 28 d7 0b 40 00 80 06 00 00  ....E..( .
0010  7f 00 00 01 7f 00 00 01 fe ca 75 4f db 33 e4 e1  ....
0020  5b 94 31 4c 50 10 27 f9 c8 c8 00 00             [..LP..
```

Wireshark - Follow TCP Stream (tcp.stream eq 0) - Adapter for loopback traffic capture

Kiểm tra một gói tin

Wireshark - Follow TCP Stream (tcp.stream eq 0) - Adapter for loopback traffic capture

ec3907af9455930f4548f5843ef6c946c31ab6be08ec8e66a846fcd40025341c82bb1fcb82fd4d908815579884fae15f3b9e02f7f8a2b372ad46ab505726fd6b61a5454b0ec30c82594241a44e7608ce6

Source

000000 127.0.0.1

000049 127.0.0.1

000118 127.0.0.1

590688 127.0.0.1

590741 127.0.0.1

591016 127.0.0.1

591026 127.0.0.1

.536974 127.0.0.1

.537003 127.0.0.1

.537231 127.0.0.1

.537247 127.0.0.1

.537485 127.0.0.1

.537515 127.0.0.1

.538758 127.0.0.1

.538813 127.0.0.1

44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface 0

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 65226, Dst Port: 30031, Seq: 65226, Win: 0, Len: 0

2 client pkt(s), 2 server pkt(s), 3 turn(s).

Entire conversation (160 bytes)

Show data as ASCII

Find:

Find Next

Filter Out This Stream

Print

Save as...

Back

Close

Help

```
C:\Windows\system32\cmd.exe - date
Microsoft Windows [Version 10.0.19045.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\admin>echo B21DCAT031 NguyenDucAnh && date
B21DCAT031 NguyenDucAnh
The current date is: 17/04/2024
Enter the new date: (dd-mm-yy)
```

3. Kết quả đạt được

- Lập trình cơ chế Client/Server dựa trên Socket.
- Thực hiện mã hóa để đảm bảo tính toàn vẹn của việc truyền các gói tin.