

Báo cáo bài thực hành số 15

Môn học

Thực tập cơ sở

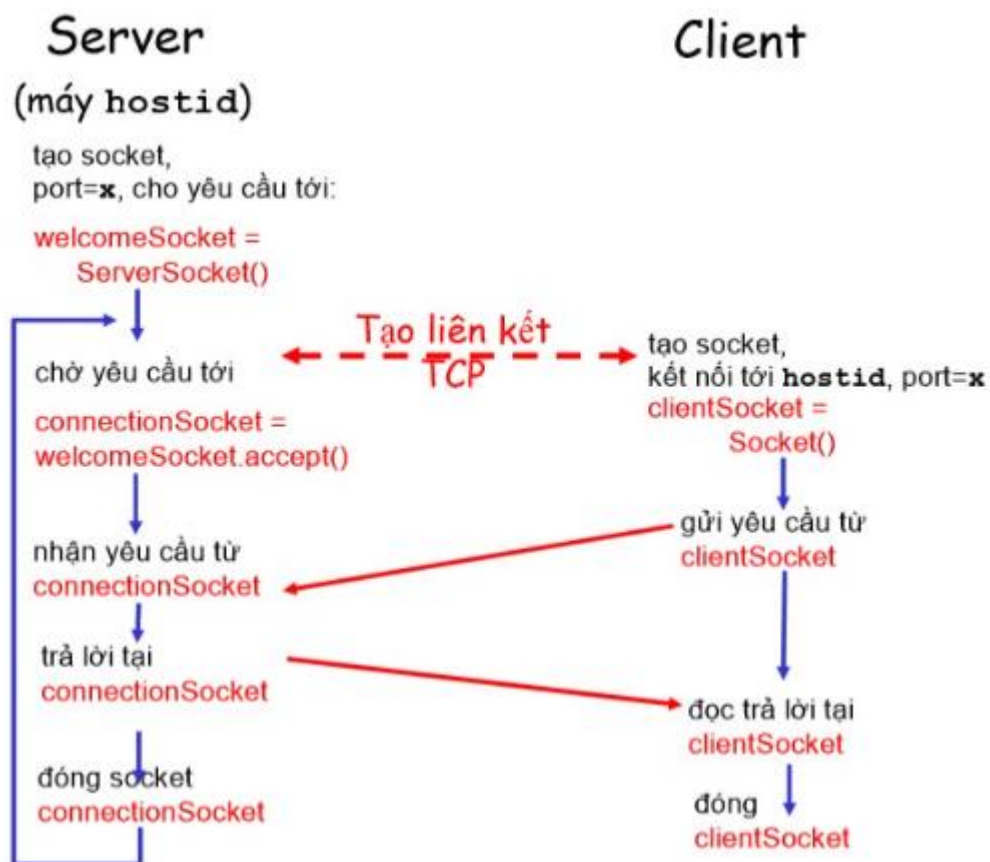
Giảng viên : Hoàng Xuân Dậu

Họ tên : Nguyễn Minh Phương

Mã SV: B19DCAT141

I. Lý thuyết:

- Socket
 - + Là điểm cuối (end-point) trong liên kết truyền thông hai chiều (two-way communication) biểu diễn kết nối giữa Client – Server
 - + Phân loại Socket:
 - Stream Socket: Dựa trên giao thức TCP, thiết lập giao tiếp 2 chiều; đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng tuần tự.
 - Datagram Socket: Dựa trên giao thức UDP, không yêu cầu có sự thiết lập kết nối giữa 2 process; ưu điểm là tốc độ giao thức nhanh.
- Lập trình socket với TCP



II. Thực hành:

- Lập trình client

```
client.py X
Socket-Programming > client.py > ...

1  import socket
2  import threading
3  import time
4  import hash
5
6  # IP Address and Port
7  HOST = '127.0.0.1'
8  PORT = 8080
9
10 # Key to ensure integrity msg
11 SECRET_KEY = "5D2E44719232EA78CD2B32"
12
13 # Create socket
14 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
15 server_address = (HOST, PORT)
16 print('connecting to %s port ' %str(server_address))
17 s.connect(server_address)
18
19 # Function sent message to server
20 def sent_msg():
21     while True:
22         msg = input('\nSent to server: ')
23         if len(msg) == 0:
24             continue
25
26         time.sleep(0.2)
27
28         # Hash msg and send msg + hash code
29         hash_code = hash.get_hash_code(msg, SECRET_KEY)
30         data_sent = msg + '|' + hash_code
31
32         # Send data with TCP
33         s.sendall(bytes(data_sent, "utf8"))
34
35 # Function receive data to server
36 def rev_msg():
37     while True:
38         # Receive data
39         data = s.recv(1024)
40         msg_rev, hash_code_rev = data.decode("utf8").split('|')
41
42         # Check integrity of message
43         if hash.check_integrity_msg(msg_rev, SECRET_KEY, hash_code_rev):
44             print('\nReceive from server: ', msg_rev)
45         else:
46             print("\nThe received message has lost its integrity")
47
48         time.sleep(0.2)
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phuong>echo "Nguyen Minh Phuong - B19DCAT141"
"Nguyen Minh Phuong - B19DCAT141"

C:\Users\phuong>
```

```
client.py X
Socket-Programming > client.py > ...

51 try:
52     # Thread sent message
53     th1 = threading.Thread(target=sent_msg, name='t1')
54
55     # Thread receive data
56     th2 = threading.Thread(target=rev_msg, name='t2')
57
58     # Set two thread is Deamon
59     th1.daemon = True
60     th2.daemon = True
61
62     # Start thread
63     th1.start()
64     th2.start()
65
66     time.sleep(1)
67
68     # End thread
69     th1.join()
70     th2.join()
71
72 finally:
73     s.close()
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phuong>echo "Nguyen Minh Phuong - B19DCAT141"
"Nguyen Minh Phuong - B19DCAT141"

C:\Users\phuong>
```

- Lập trình server

```
server.py M X
Socket-Programming > server.py > ...
1  import socket
2  import sys
3  import threading
4  import time
5  import hash
6
7  # IP Address and Port
8  HOST = '127.0.0.1'
9  PORT = 8080
10
11 # Key to ensure integrity msg
12 SECRET_KEY = "5D2E44719232EA78CD2B32"
13
14 # Create socket
15 # Listen max 2 socket
16 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
17 s.bind((HOST, PORT))
18 s.listen(2)
19
20 # Function sent message to client
21 def sent_msg(client):
22     while True:
23         msg = input('\nSent to client: ')
24         if len(msg) == 0:
25             continue
26
27         time.sleep(0.2)
28
29         # Hash msg and send msg + hash code
30         hash_code = hash.get_hash_code(msg, SECRET_KEY)
31         data_sent = msg + '|' + hash_code
32
33         # Send data with TCP
34         client.sendall(bytes(data_sent, "utf8"))
35
36
37 # Function receive data to client
38 def rev_msg(client):
39     while True:
40         # Receive data
41         data = client.recv(1024)
42         msg_rev, hash_code_rev = data.decode("utf8").split('|')
43
44         # Check integrity of message
45         if hash.check_integrity_msg(msg_rev, SECRET_KEY, hash_code_rev):
46             print("\nReceive from client: " + msg_rev)
47         else:
48             print("\nThe received message has lost its integrity")
49         time.sleep(0.2)
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phuong>echo "Nguyen Minh Phuong - B19DCAT141"
"Nguyen Minh Phuong - B19DCAT141"

C:\Users\phuong>
```

```
server.py M X
Socket-Programing > server.py > rev_msg
51
52 while True:
53     # Accept client connect to server
54     # Create client socket
55     client, addr = s.accept()
56
57     try:
58         print('Connected by', addr)
59
60         # Thread sent message
61         th1 = threading.Thread(target=sent_msg, args=(client,), name='t1')
62
63         # Thread receive data
64         th2 = threading.Thread(target=rev_msg, args=(client,), name='t2')
65
66         # Set two thread is Deamon
67         th1.daemon = True
68         th2.daemon = True
69
70         # Start thread
71         th1.start()
72         th2.start()
73
74         time.sleep(1)
75
76         # End thread
77         th1.join()
78         th2.join()
79
80     finally:
81         client.close()
82
83 s.close()
```

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phuong>echo "Nguyen Minh Phuong - B19DCAT141"
"Nguyen Minh Phuong - B19DCAT141"

C:\Users\phuong>
```

- Chạy server sau đó chạy client

```
PS C:\Users\phuong\OneDrive\Desktop\TTCs\Socket-Programing> python server.py
Connected by ('127.0.0.1', 56899)

Sent to client:
Receive from client: NMP-B19DCAT141 client send

Sent to client: NMP-B19DCAT141 server send

Sent to client: []

PS C:\Users\phuong\OneDrive\Desktop\TTCs\Socket-Programing> python client.py
connecting to ('127.0.0.1', 8888) port

Sent to server: NMP-B19DCAT141 client send

Sent to server:
Receive from server: NMP-B19DCAT141 server send

Sent to server: []
```

- Sử dụng Wireshark để bắt các thông tin đã gửi từ client đến server và ngược lại

*Adapter for loopback traffic capture

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port==8080

No.	Time	Source	Destination	Protocol	Length	Info
27	39.446803	127.0.0.1	127.0.0.1	TCP	56	8080 → 56099 [SYN, ACK] S
28	39.446843	127.0.0.1	127.0.0.1	TCP	44	56099 → 8080 [ACK] Seq=1
29	51.197698	127.0.0.1	127.0.0.1	TCP	135	56099 → 8080 [PSH, ACK] S
30	51.197728	127.0.0.1	127.0.0.1	TCP	44	8080 → 56099 [ACK] Seq=1
35	66.599089	127.0.0.1	127.0.0.1	TCP	135	8080 → 56099 [PSH, ACK] S
36	66.599113	127.0.0.1	127.0.0.1	TCP	44	56099 → 8080 [ACK] Seq=92

> Frame 35: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface \Device\NPF_Loopback
 > Null/Loopback
 > Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 > Transmission Control Protocol, Src Port: 8080, Dst Port: 56099, Seq: 1, Ack: 92, Len: 91

```

0000  02 00 00 00 45 00 00 83 42 9c 40 00 80 06 00 00  ....E...B:@....
0010  7f 00 00 01 7f 00 00 01 1f 90 db 23 f2 b6 be 75  .......#...u
0020  af 47 c2 5f 50 18 27 f9 74 13 00 00 4e 4d 50 2d  ..G.P...t...NMP-
0030  42 31 39 44 43 41 54 31 34 31 20 73 65 72 76 65  B19DCAT1 41 serve
0040  72 20 73 65 6e 64 7c 33 46 39 41 32 39 36 31 31  r_send|3 F9A29611
0050  35 37 41 32 34 44 36 30 34 37 35 33 38 33 41 43  57A24D60 475383AC
0060  34 39 38 41 44 42 39 32 44 32 35 31 39 38 34 34  498ADB92 D2519844
0070  33 31 37 46 32 42 43 31 41 37 34 36 43 42 42 35  317F2BC1 A746CBB5
0080  39 43 37 44 44 36 30                                9C7DD60
  
```

wireshark_NPF_LoopbackRKLHL1.pcapng || Packets: 102 · Displayed: 20 (19.6%) || Profile: Default

*Adapter for loopback traffic capture

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port==8080

No.	Time	Source	Destination	Protocol	Length	Info
27	39.446803	127.0.0.1	127.0.0.1	TCP	56	8080 → 56099 [SYN, ACK] S
28	39.446843	127.0.0.1	127.0.0.1	TCP	44	56099 → 8080 [ACK] Seq=1
29	51.197698	127.0.0.1	127.0.0.1	TCP	135	56099 → 8080 [PSH, ACK] S
30	51.197728	127.0.0.1	127.0.0.1	TCP	44	8080 → 56099 [ACK] Seq=1
35	66.599089	127.0.0.1	127.0.0.1	TCP	135	8080 → 56099 [PSH, ACK] S
36	66.599113	127.0.0.1	127.0.0.1	TCP	44	56099 → 8080 [ACK] Seq=92

> Frame 29: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface \Device\NPF_Loopback
 > Null/Loopback
 > Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
 > Transmission Control Protocol, Src Port: 56099, Dst Port: 8080, Seq: 1, Ack: 1, Len: 91

```

0000  02 00 00 00 45 00 00 83 42 96 40 00 80 06 00 00  ....E...B:@....
0010  7f 00 00 01 7f 00 00 01 db 23 1f 90 af 47 c2 04  .......#...G..
0020  f2 b6 be 75 50 18 27 f9 6e 3a 00 00 4e 4d 50 2d  ..uP...n...NMP-
0030  42 31 39 44 43 41 54 31 34 31 20 63 6c 69 65 6e  B19DCAT1 41 clien
0040  74 20 73 65 6e 64 7c 36 43 30 31 36 30 45 37 46  t_send|6 C0160E7F
0050  45 30 39 35 36 36 38 31 44 45 31 30 36 31 43 42  E0956681 DE1061CB
0060  30 35 35 36 46 46 42 35 34 42 39 43 31 41 39 36  0556FFB5 4B9C1A96
0070  39 39 36 35 38 38 35 39 42 41 38 36 46 45 37 42  99658859 BA86FE7B
0080  46 45 31 39 46 36 44                                FE19F6D
  
```

wireshark_NPF_LoopbackRKLHL1.pcapng || Packets: 149 · Displayed: 20 (13.4%) || Profile: Default

- Từ client và server, sửa đổi để sao cho: khi gửi thông điệp sẽ gửi kèm theo giá trị băm của (thông điệp+key) để phía bên kia kiểm tra xác minh tính toàn vẹn. Hai bên có thể thống nhất một giá trị key trước đó.
 - + Tạo hash module trong Python:

hash.py

Socket-Programming > hash.py > ...

```

1  import hmac
2  import hashlib
3  import binascii
4
5  def get_hash_code(msg, key):
6      # Convert hex to bin
7      # Encode string with utf8
8      # Hash msg with SHA256, convert to hex
9      key = binascii.unhexlify(key)
10     msg = msg.encode()
11     return hmac.new(key, msg, hashlib.sha256).hexdigest().upper()
12
13 def check_integrity_msg(msg, key, code_rev):
14     # Hash msg with key and compare
15     if code_rev == get_hash_code(msg, key):
16         return True
17     else:
18         return False
19
20 if __name__ == "__main__":
21     print(get_hash_code("PYTHON | language", "5D2A7139232AD22B23"))

```

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phuong>echo "Nguyen Minh Phuong - B19DCAT141"
"Nguyen Minh Phuong - B19DCAT141"

C:\Users\phuong>

Function sent message to client

def sent_msg(client):

while True:

msg = input('\nSent to client: ')

if len(msg) == 0:

continue

time.sleep(0.2)

Hash msg and send msg + hash code

hash_code = hash.get_hash_code(msg, SECRET_KEY)

data_sent = msg + '|' + hash_code

Send data with TCP

client.sendall(bytes(data_sent, "utf8"))

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phuong>echo "Nguyen Minh Phuong - B19DCAT141"
"Nguyen Minh Phuong - B19DCAT141"

C:\Users\phuong>

Function receive data to client

def rev_msg(client):

while True:

Receive data

data = client.recv(1024)

msg_rev, hash_code_rev = data.decode("utf8").split('|')

Check integrity of message

if hash.check_integrity_msg(msg_rev, SECRET_KEY, hash_code_rev):

print("\nReceive from client: " + msg_rev)

else:

print("\nThe received message has lost its integrity")

time.sleep(0.2)

C:\WINDOWS\system32\cmd.exe

Microsoft Windows [Version 10.0.19043.1645]
(c) Microsoft Corporation. All rights reserved.

C:\Users\phuong>echo "Nguyen Minh Phuong - B19DCAT141"
"Nguyen Minh Phuong - B19DCAT141"

C:\Users\phuong>

- Thay đổi giá trị key tại client và thực hiện gửi lại, nếu không đáp ứng tính toàn vẹn cần thông báo: “The received message has lost its integrity.”

The screenshot shows a code editor with two files: `server.py` and `client.py`. Both files contain a `SECRET_KEY` variable. The `server.py` file has a key of `"5D2E44719232EA78CD2B32"`, and the `client.py` file has a key of `"5D2E44719232EA55CD5B55"`. Below the code editor, a terminal window shows the execution of the programs. The server terminal shows the message "The received message has lost its integrity" being sent to the client. The client terminal shows the message "NWP-B19DCAT141 client send" being sent to the server.

```

server.py
10
11 # Key to ensure integrity msg
12 SECRET_KEY = "5D2E44719232EA78CD2B32"
13
14 # Create socket
15 # Listen max 2 socket
16 s = socket.socket(socket.AF_INET,
17 s.bind((HOST, PORT))
18 s.listen(2)
19

client.py
10 # Key to ensure integrity msg
11 SECRET_KEY = "5D2E44719232EA55CD5B55"
12

Terminal
PS C:\Users\phun\OneDrive\Desktop\TTCs\Socket-Programming> python server.py
Connected to ('127.0.0.1', 58844)

Sent to client:
The received message has lost its integrity

Sent to client: []

PS C:\Users\phun\OneDrive\Desktop\TTCs\Socket-Programming> python client.py
connecting to ('127.0.0.1', 8080) port

Sent to server: NWP-B19DCAT141 client send

Sent to server: []

```

- Bắt được các bản tin trao đổi giữa client và server trong Wireshark

