

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



BÁO CÁO

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

Giảng viên hướng dẫn: Vũ Minh Mạnh

Sinh viên thực hiện: Nguyễn Quốc Vượng

Mã sinh viên: B21DCAT227

Lớp: D21CQAT03-B

Hà Nội, 2023

Môn học Thực tập cơ sở

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

1.1 Mục đích

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

1.2 Nội dung thực hành

1.2.1 Tìm hiểu lý thuyết

- Tìm hiểu về các khái niệm liên quan tới lập trình socket với TCP
- Tham khảo tài liệu: Chapter 2: Application Layer V8.1 (9/2020) tại địa chỉ http://gaia.cs.umass.edu/kurose_ross/ppt.php (chú ý ví dụ từ trang 105).

Một số phiên bản cũ hơn có thể lập trình bằng Java thay vì Python.

➤ Socket là gì?

- **Socket** là điểm cuối của một kết nối hai chiều giữa hai chương trình đang chạy trên mạng. Socket cung cấp cơ chế để truyền và nhận dữ liệu qua mạng.

➤ TCP là gì?

- **TCP (Transmission Control Protocol)** là một giao thức hướng kết nối, đảm bảo việc truyền dữ liệu tin cậy, theo thứ tự và không mất mát. TCP sử dụng cơ chế bắt tay ba bước (three-way handshake) để thiết lập kết nối và đảm bảo dữ liệu được truyền thành công.

➤ Các loại socket

- **Stream Socket (SOCK_STREAM):** Dùng cho TCP, đảm bảo truyền dữ liệu liên tục và tin cậy.
- **Datagram Socket (SOCK_DGRAM):** Dùng cho UDP, không đảm bảo truyền dữ liệu tin cậy hoặc theo thứ tự.

1.2.2 Chuẩn bị môi trường

- Môi trường Python hoặc Java để chạy được ứng dụng client/server đã lập

trình.

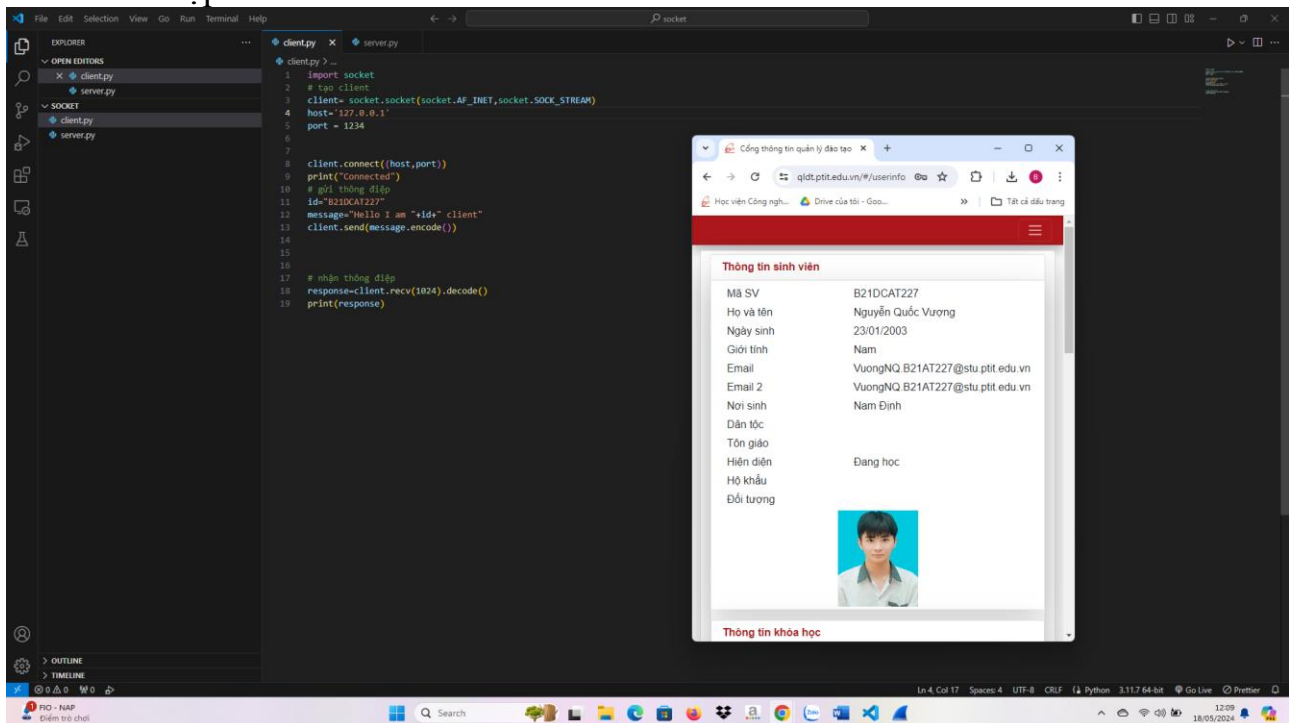
- Phần mềm Wireshark

1.2.3 Các bước thực hiện và kết quả cần đạt

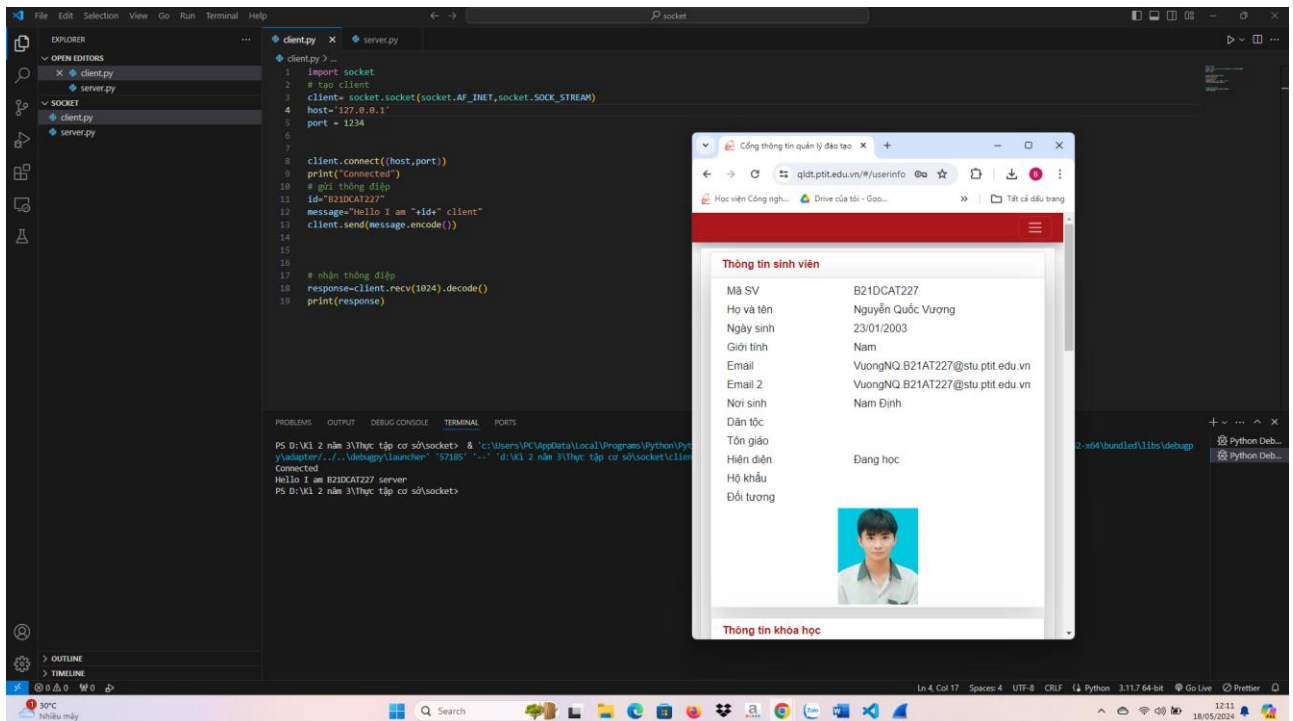
1.2.3.1 Lập trình client và server với TCP socket

a) Các bước thực hiện

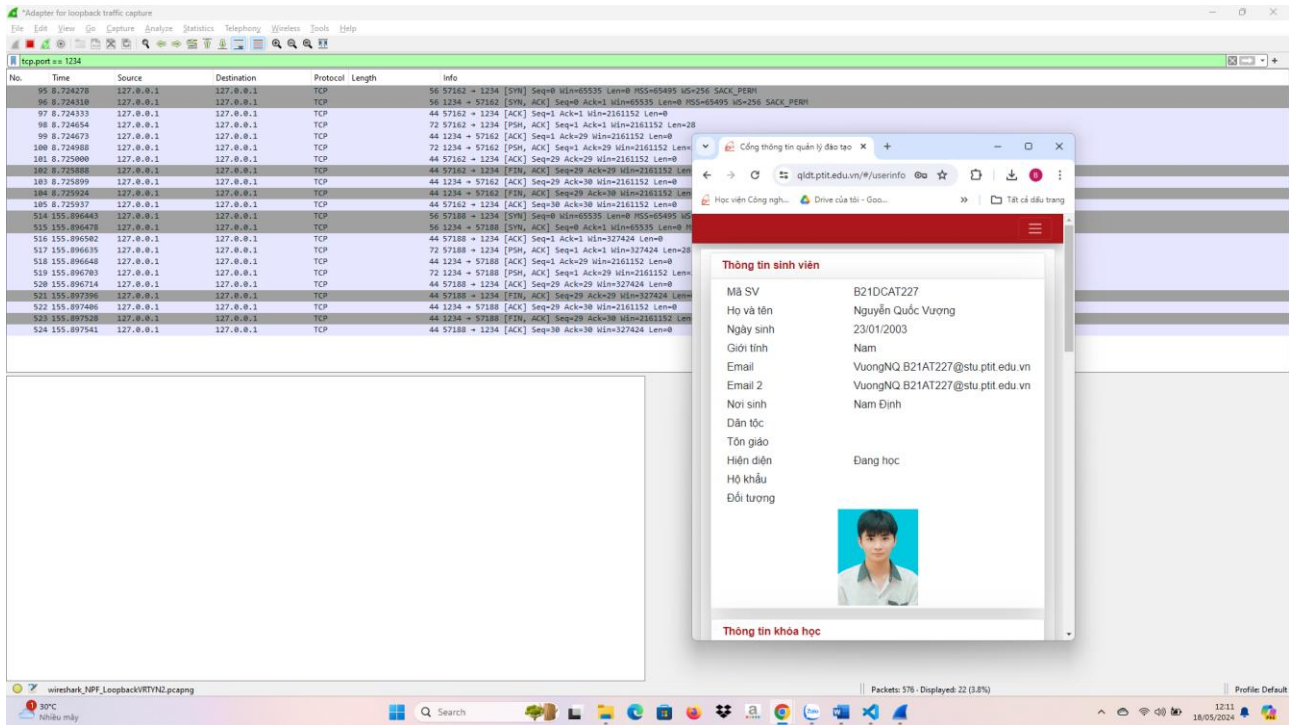
- Lập trình client



- Lập trình server



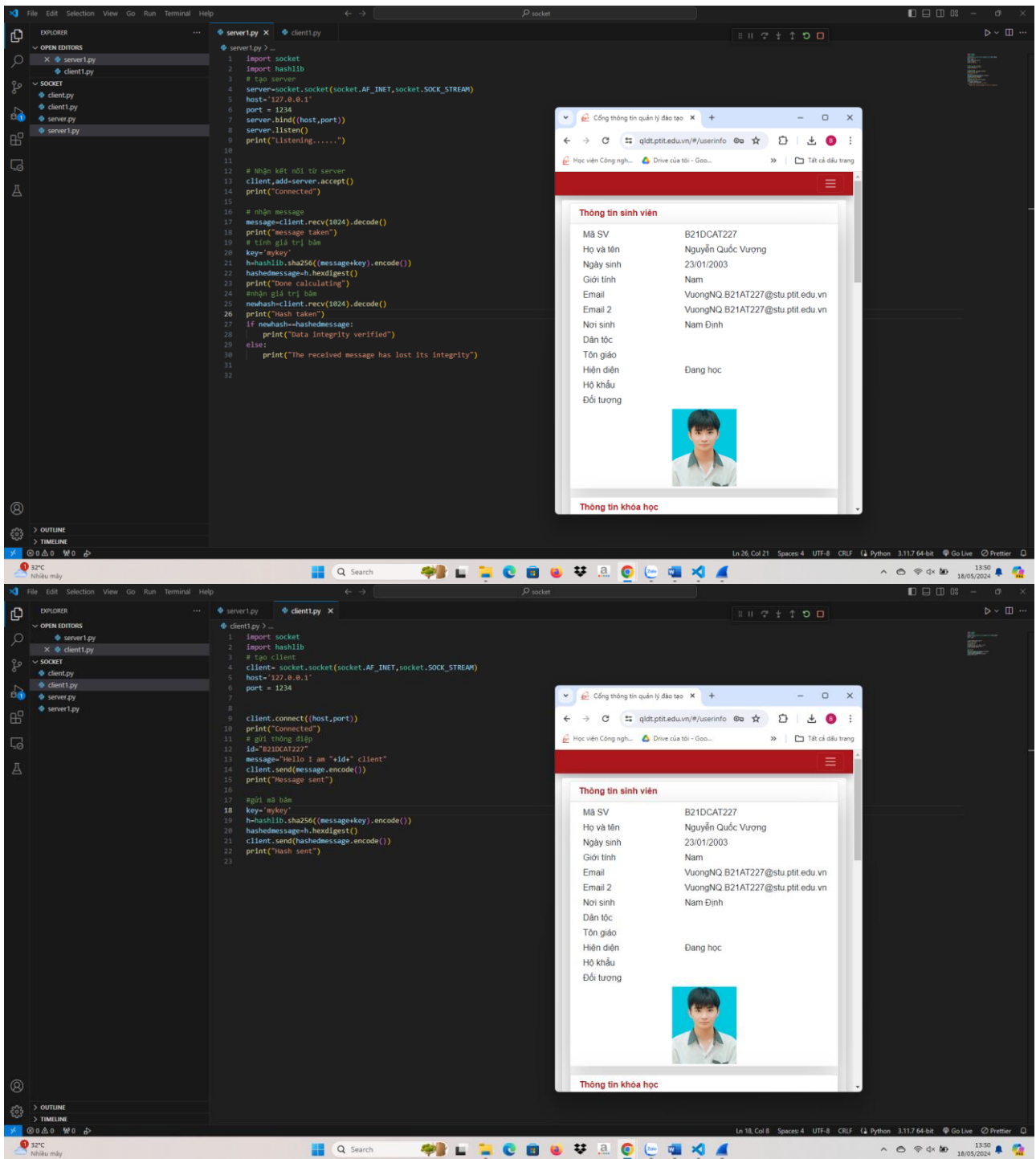
- Sử dụng Wireshark để bắt các thông tin đã gửi từ client đến server và ngược lại

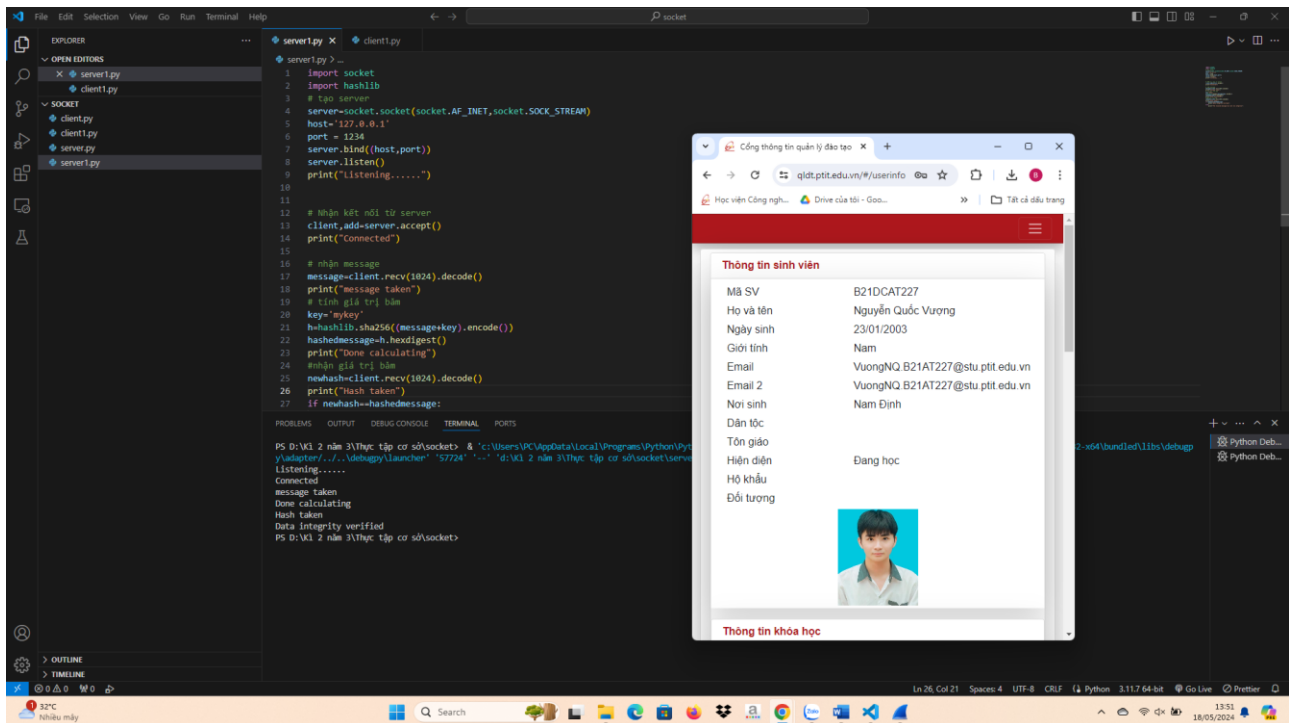


1.2.3.2 Trao đổi thông điệp giữa client và server và đảm bảo tính toàn vẹn của thông điệp khi trao đổi

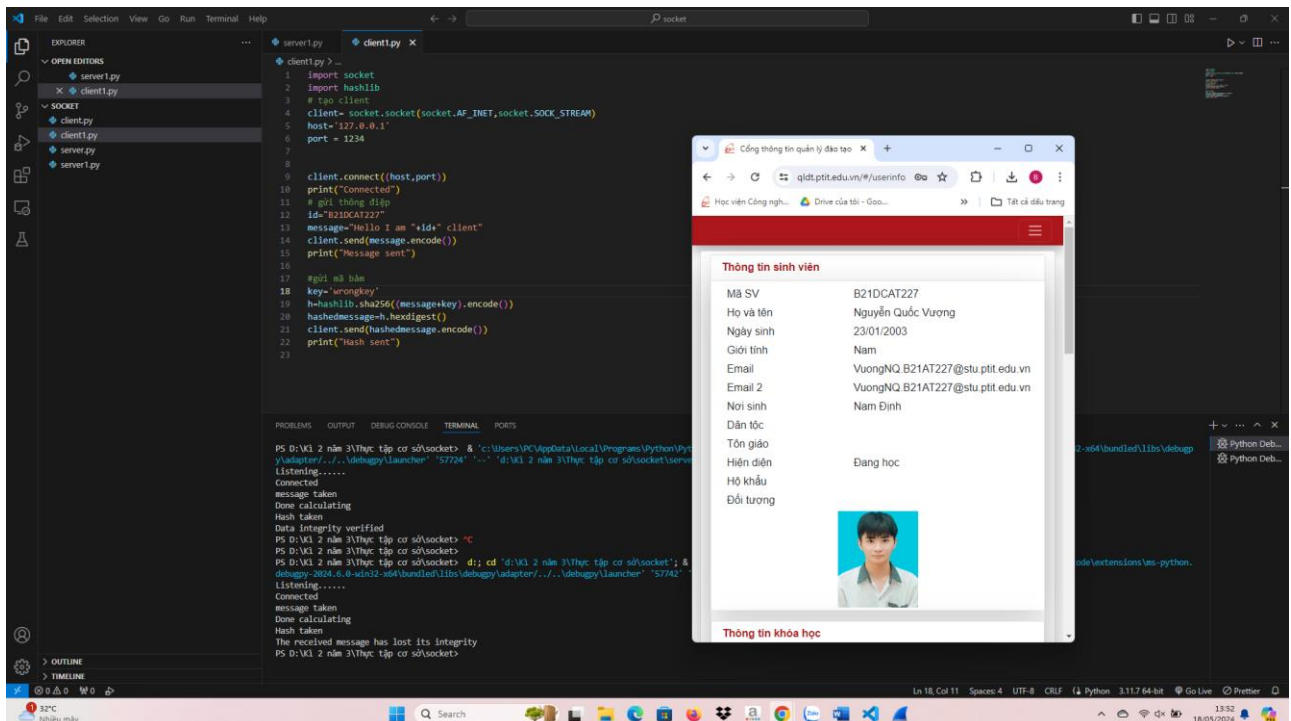
a) Các bước thực hiện

- 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.





- 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."



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

Adapter for loopback traffic capture

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 1234

No.	Time	Source	Destination	Protocol	Length	Info
12671	5551.152879	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [ACK] Seq=1 Ack=29 Win=2161152 Len=0
12672	5551.167897	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [FIN, ACK] Seq=29 Ack=1 Win=32748 Len=0
12673	5551.167898	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [ACK] Seq=1 Ack=30 Win=2161152 Len=0
12674	5551.167898	127.0.0.1	127.0.0.1	TCP	72	1234 → 5768 [PSH, ACK] Seq=1 Ack=30 Win=2161152 Len=28
12675	5551.171212	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [FIN, ACK] Seq=30 Ack=30 Win=0 Len=0
12963	5634.586323	127.0.0.1	127.0.0.1	TCP	56	5768 → 1234 [SYN] Seq=0 Win=65535 Len=0 MSS=65495
12964	5634.586362	127.0.0.1	127.0.0.1	TCP	56	1234 → 5768 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
12965	5634.586384	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
12966	5634.586485	127.0.0.1	127.0.0.1	TCP	72	5768 → 1234 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=28
12967	5634.586621	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [ACK] Seq=1 Ack=29 Win=2161152 Len=0
12968	5634.586853	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [FIN, ACK] Seq=29 Ack=1 Win=2161152 Len=0
12969	5634.586775	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [ACK] Seq=1 Ack=30 Win=2161152 Len=0
12970	5634.586867	127.0.0.1	127.0.0.1	TCP	72	1234 → 5768 [PSH, ACK] Seq=1 Ack=30 Win=2161152 Len=28
12971	5634.586889	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [RST, ACK] Seq=30 Ack=29 Win=0 Len=0
13244	5667.192455	127.0.0.1	127.0.0.1	TCP	56	5768 → 1234 [SYN] Seq=0 Win=65535 Len=0 MSS=65495
13245	5667.192180	127.0.0.1	127.0.0.1	TCP	56	1234 → 5768 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
13246	5667.192123	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
13247	5667.192379	127.0.0.1	127.0.0.1	TCP	72	5768 → 1234 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=28
13248	5667.192396	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [ACK] Seq=1 Ack=29 Win=2161152 Len=0
13249	5667.192582	127.0.0.1	127.0.0.1	TCP	108	5768 → 1234 [PSH, ACK] Seq=29 Ack=1 Win=2161152 Len=28
13250	5667.192590	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [ACK] Seq=1 Ack=93 Win=2161152 Len=0
13251	5667.192790	127.0.0.1	127.0.0.1	TCP	72	1234 → 5768 [PSH, ACK] Seq=1 Ack=93 Win=2161152 Len=28
13252	5667.192711	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [ACK] Seq=93 Ack=29 Win=2161152 Len=0
13253	5667.193635	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [FIN, ACK] Seq=93 Ack=29 Win=2161152 Len=0
13254	5667.193849	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [ACK] Seq=29 Ack=94 Win=2161152 Len=0
13255	5667.193878	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [FIN, ACK] Seq=94 Ack=94 Win=2161152 Len=0
13256	5667.193896	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [ACK] Seq=94 Ack=30 Win=2161152 Len=0
13513	5724.375889	127.0.0.1	127.0.0.1	TCP	56	5768 → 1234 [SYN] Seq=0 Win=65535 Len=0 MSS=65495
13514	5724.375961	127.0.0.1	127.0.0.1	TCP	56	1234 → 5768 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
13515	5724.375943	127.0.0.1	127.0.0.1	TCP	44	5768 → 1234 [ACK] Seq=1 Ack=1 Win=2161152 Len=0
13516	5724.376068	127.0.0.1	127.0.0.1	TCP	72	5768 → 1234 [PSH, ACK] Seq=1 Ack=1 Win=2161152 Len=28
13517	5724.376081	127.0.0.1	127.0.0.1	TCP	44	1234 → 5768 [ACK] Seq=1 Ack=29 Win=2161152 Len=0

Frame 12966: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface \Device\NPF_{...}_Loopback, id 0

Null/Loopback

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

Transmission Control Protocol, Src Port: 5768, Dst Port: 1234, Seq: 1, Ack: 1, Len: 28

Data (28 bytes)

0000 82
0010 7F
0020 30
0030 6F
0040 97

Cổng thông tin quản lý đào tạo

qldt.ptit.edu.vn/#/userinfo

Học viên Công nghệ... Drive của tôi - Goo...

Tất cả dấu trang

Thông tin sinh viên

Mã SV: B21DCAT227

Họ và tên: Nguyễn Quốc Vương

Ngày sinh: 23/01/2003

Giới tính: Nam

Email: VươngNQ.B21AT227@stu.ptit.edu.vn

Email 2: VươngNQ.B21AT227@stu.ptit.edu.vn

Nơi sinh: Nam Định


Dân tộc: [Blank]

Tôn giáo: [Blank]

Hiện diện: Đang học

Hồ khẩu: [Blank]

Đối tượng: [Blank]



Thông tin khóa học