

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



**Môn học: Thực Tập Cơ Sở**  
**Báo Cáo Bài Thực Hành 8**  
**Bắt dữ liệu mạng**

**Họ và tên:** Trần Thị Thu Phương

**Mã sinh viên:** B21DCAT151

**Nhóm môn học:** 04

**Giảng viên:** Đinh Trường Duy

Hà Nội, 1/2024

## Mục lục

<b>1. Mục đích .....</b>	<b>3</b>
<b>2. Nội dung thực hành .....</b>	<b>3</b>
<b>2.1. Cơ sở lý thuyết.....</b>	<b>3</b>
2.1.1. Sniffer là gì?.....	3
2.1.2. Tìm hiểu tính năng và hoạt động của công cụ bắt dữ liệu mạng Tcpdump..	3
2.1.3. Tìm hiểu tính năng và hoạt động của công cụ bắt dữ liệu mạng Wireshark	4
2.1.4. Tìm hiểu tính năng và hoạt động của công cụ bắt dữ liệu mạng Network Miner .....	5
<b>2.2. Nội dung thực hành .....</b>	<b>6</b>
2.2.1. Chuẩn bị môi trường.....	6
2.2.2. Sử dụng Tcpdump.....	7
2.2.3. Sử dụng Wireshark để bắt và phân tích các gói tin .....	11
2.2.4. Sử dụng Network Miner để bắt và phân tích các gói tin .....	14
<b>3. Kết luận .....</b>	<b>16</b>
<b>4. Tài liệu tham khảo.....</b>	<b>16</b>

## Danh mục hình ảnh

Cấu hình topo mạng.....	7
xem tất cả các interfaces trong hệ thống .....	8
Kích hoạt các interfaces (eth0, eth1) hoạt động ở chế độ hỗn hợp .....	8
Bắt gói tin trên dải mạng 192.168.100.0/24 .....	9
Dải Internal: 192.168.100.201→ 192.168.100.3.....	9
Trên máy Linux Sniffer, bắt gói tin trên dải 192.168.100.0/24 .....	9
Dải External: 10.10.19.202→ 10.10.19.148.....	10
Trên máy Linux Sniffer, bắt gói tin trên dải 10.10.19.0/24 .....	10
Các dữ liệu đã bắt trên dải Internal.....	11
Các dữ liệu đã bắt trên dải External.....	11
Windows 7 Attack kết nối tới ftp server trên máy Window Server Internal.....	12
Lọc gói tin theo giao thức ftp .....	13
Trên máy Kali Linux External kết nối ftp đến máy Ftp Server.....	14
Bắt gói tin trên dải 10.10.19.0/24 và lọc theo giao thức ftp .....	14
Kết nối đến trang web của Windows 2003 Server Internal.....	15
Chọn File/ index.html để xem dữ liệu gói tin vừa bắt được.....	16

## 1. Mục đích

Bài thực hành giúp sinh viên nắm được công cụ và cách thức bắt dữ liệu mạng, bao gồm:

- Sử dụng tcpdump để bắt gói tin mạng
- Sử dụng Wireshark để bắt và phân tích gói tin mạng (HTTP/HTTPS/FTP TCP/IP)
- Sử dụng Network Miner để bắt và phân tích gói tin mạng

## 2. Nội dung thực hành

### 2.1. Cơ sở lý thuyết

#### 2.1.1. Sniffer là gì?

Sniffer hay packet sniffer là một chương trình phần mềm nghe trộm gói tin (còn gọi là chương trình phân tích mạng, phân tích giao thức hay nghe trộm Ethernet), có khả năng chặn bắt và ghi lại lưu lượng dữ liệu qua một mạng viễn thông số hoặc một phần của một mạng. Khi các dòng dữ liệu di chuyển qua lại trong một mạng, chương trình sẽ chặn bắt các gói tin rồi giải mã và phân tích nội dung của nó theo đặc tả RFC hoặc các đặc tả thích hợp khác.

Tùy theo cấu trúc mạng (hub hay chuyển mạch) mà có thể nghe trộm tất cả hoặc chỉ một phần lưu lượng dữ liệu qua lại từ một máy trong mạng. Đối với mục đích giám sát mạng (network monitoring), có thể theo dõi tất cả các gói tin trong một mạng LAN bằng cách sử dụng một thiết bị chuyển mạch với một cổng theo dõi (lắp lại tất cả các gói tin đi qua các cổng của thiết bị chuyển mạch).

#### 2.1.2. Tìm hiểu tính năng và hoạt động của công cụ bắt dữ liệu mạng Tcpdump

Tcpdump là một công cụ dòng lệnh được sử dụng để ghi lại và phân tích gói tin trên mạng. Nó cho phép bạn theo dõi lưu lượng mạng đi qua một giao diện cụ thể trên hệ thống của bạn. Bằng cách sử dụng các cú pháp và tùy chọn khác nhau, bạn có thể lọc và hiển thị các gói tin theo nhiều tiêu chí khác nhau như địa chỉ IP, cổng, giao thức, và nhiều hơn nữa. Tcpdump là một công cụ mạnh mẽ được sử dụng rộng rãi trong quản trị hệ thống và mạng để chẩn đoán và gỡ lỗi vấn đề liên quan đến mạng.

**Tcpdump sẽ giúp bạn phân các gói dữ liệu phù hợp với dòng lệnh mang theo, cụ thể:**

- Bắt bản tin và lưu bằng định dạng PCAP (có thể đọc bởi wireshark)
- Nhìn thấy trực tiếp các bản tin điều khiển hệ thống Linux sử dụng wireshark, xem chi tiết remote packet capture using Wireshark và tcpdump
- Có thể nhìn thấy các bản tin trên DUMP trên terminal
- Tạo các bộ lọc Filter để bắt bản tin cần thiết như: http, ssh, ftp...
- Ngoài ra tcpdump còn sử dụng nhiều option khác nhau nữa

**Định dạng chung của một dòng giao thức tcpdump:**

*time-stamp src > dst: flags data-seqno ack window urgent options*

Trong đó:

- Time-stamp: hiển thị thời gian gói tin được capture.
- Src và dst: hiển thị địa IP của người gửi và người nhận.
- Cờ Flag thì bao gồm các giá trị sau:
  - + S(SYN): Được sử dụng trong quá trình bắt tay của giao thức TCP.
  - + .(ACK): Được sử dụng để thông báo cho bên gửi biết là gói tin đã nhận được dữ liệu thành công.
  - + F(FIN): Được sử dụng để đóng kết nối TCP.
  - + P(PUSH): Thường được đặt ở cuối để đánh dấu việc truyền dữ liệu.
  - + R(RST): Được sử dụng khi muốn thiết lập lại đường truyền.
- Data-seqno: Số sequence number của gói dữ liệu hiện tại.
- ACK: Mô tả số sequence number tiếp theo của gói tin do bên gửi truyền (số sequence number mong muốn nhận được).
- Window: Vùng nhớ đệm có sẵn theo hướng khác trên kết nối này.
- Urgent: Cho biết có dữ liệu khẩn cấp trong gói tin.

### Cách hoạt động của Tcpdump

Tcpdump là một công cụ dòng lệnh được sử dụng để theo dõi và phân tích gói tin trên mạng. Nó hoạt động bằng cách lắng nghe và ghi lại các gói tin mạng đang đi qua một giao diện mạng cụ thể trên một máy tính. Khi được chạy, Tcpdump sẽ hiển thị thông tin về các gói tin này, bao gồm địa chỉ nguồn và đích, loại giao thức, dữ liệu payload, và nhiều thông tin khác. Người dùng có thể sử dụng các tùy chọn và bộ lọc để tinh chỉnh việc theo dõi và phân tích theo nhu cầu cụ thể của họ.

#### 2.1.3. Tìm hiểu tính năng và hoạt động của công cụ bắt dữ liệu mạng Wireshark

Wireshark là một công cụ phân tích gói tin mạng mạnh mẽ và đa năng. Nó cho phép bạn chụp, xem xét và phân tích gói tin trên mạng. Wireshark hỗ trợ nhiều loại giao thức mạng và cung cấp các tính năng như lọc gói tin, phân tích luồng dữ liệu, và đồ thị hoạt động mạng. Công cụ này thường được sử dụng để chẩn đoán và gỡ lỗi vấn đề liên quan đến mạng, cũng như để nghiên cứu bảo mật mạng và kiểm tra hiệu suất mạng. Wireshark có giao diện đồ họa dễ sử dụng và được hỗ trợ trên nhiều hệ điều hành khác nhau.

Wireshark là một phần mềm dùng để phân tích và giám sát lưu lượng mạng. Dưới đây là một số chức năng chính của Wireshark:

- **Phân tích Gói Tin:** Wireshark cho phép bạn theo dõi và phân tích từng gói tin dữ liệu trên mạng. Bạn có thể xem các thông tin chi tiết như nguồn, đích, loại gói tin, dữ liệu payload và nhiều thông tin khác.

- **Đánh giá Hiệu suất Mạng:** Wireshark cung cấp thông tin về thời gian phản hồi (response time), độ trễ (latency), và các thống kê khác, giúp đánh giá hiệu suất của mạng.
- **Phân tích Giao thức:** Wireshark hỗ trợ nhiều giao thức mạng khác nhau. Bạn có thể xem và phân tích giao thức HTTP, TCP, UDP, IP, DNS, và nhiều giao thức khác.
- **Điều tra Vấn đề Mạng:** Khi xảy ra vấn đề mạng, Wireshark là một công cụ mạnh mẽ để phân tích và xác định nguyên nhân của sự cố.
- **Bảo mật Mạng:** Wireshark có thể được sử dụng để phát hiện các hoạt động độc hại trên mạng. Nó cho phép bạn xem gói tin để phát hiện các tấn công mạng, như phishing hoặc kiểm soát truy cập không được ủy quyền.
- **Giáo dục và Học tập:** Wireshark là một công cụ hữu ích cho sinh viên, chuyên gia mạng, và người quan tâm đến việc hiểu rõ cách mạng hoạt động. Nó cung cấp một cách thức thực hành để nắm bắt và hiểu các khái niệm mạng.

### Cách hoạt động của Wireshark

Như đã đề cập ở trên, đây là một công cụ dùng để capture và phân tích các packet. Nó capture các lưu lượng mạng trên mạng cục bộ, sau đó sẽ lưu trữ nó để phân tích offline. Có thể capture các lưu lượng mạng từ các kết nối Ethernet, Bluetooth, Wireless (IEEE.802.11), Token Ring, Frame Relay...

Wireshark cho phép thiết lập filter (bộ lọc) trước khi bắt đầu capture hoặc thậm chí là trong quá trình phân tích. Do đó, ta có thể thu hẹp phạm vi tìm kiếm trong quá trình theo dõi mạng.

#### 2.1.4. Tìm hiểu tính năng và hoạt động của công cụ bắt dữ liệu mạng Network Miner

NetworkMiner là một công cụ phân tích mạng dành cho Windows. Nó cho phép người dùng thu thập dữ liệu từ mạng và phân tích thông tin như các máy chủ, giao thức, trình duyệt web, và nhiều hơn nữa. NetworkMiner tự động phát hiện các hoạt động mạng như kết nối TCP, truy vấn DNS và nó cũng có thể hỗ trợ trong việc phát hiện và phân loại các tập tin được truyền qua mạng. Nó thường được sử dụng để phát hiện các mối đe dọa mạng và phân tích dữ liệu từ gói tin đã chụp.

Những điểm nổi bật của NetworkMiner phải kể đến:

- Giám sát hầu như mọi gói tin trao đổi ra vào máy chủ, trong đó cho phép phát hiện ảnh, các file dữ liệu và tài khoản đăng nhập.
- Dữ liệu hiển thị ở dạng rất dễ hiểu.
- Dung lượng nhẹ (phiên bản 2.6 sau khi giải nén chỉ chiếm 47,9 MB), không cần cài đặt (chỉ cần tải về, giải nén là sử dụng được ngay) và rất dễ sử dụng.

- Có hai phiên bản miễn phí và pro (trả phí) để lựa chọn. Trong đó, phiên bản trả phí cho phép tìm kiếm trực tuyến thông tin về địa chỉ IP.
- Khả năng phân tích email trao đổi qua các giao thức SMTP, POP3 và IMAP.
- Nâng cấp khả năng phát hiện mật khẩu, phát hiện trao đổi dữ liệu qua giao thức FTP, những dấu hiệu bất thường trong trao đổi dữ liệu qua giao thức HTTP và HTTP/2.
- Nâng cấp khả năng tương thích với hệ điều hành Linux.
- Hỗ trợ phân tích các gói tin qua giao thức GRE, PPPoE, VXLAN, OpenFlow, MPLS và EoMPLS.

### Cách hoạt động của NetworkMiner

NetworkMiner là một công cụ phân tích mạng có khả năng thu thập dữ liệu từ gói tin mạng trên một giao diện cụ thể trên máy tính. Sau đó, nó phân tích các gói tin để trích xuất thông tin quan trọng như địa chỉ IP, tên miền, thông tin trình duyệt web và các tập tin được truyền qua mạng. Dữ liệu được hiển thị trên giao diện người dùng và có thể được lưu trữ dưới dạng tập tin PCAP để phân tích và thẩm định sau này.

#### 2.1.5. Một số điều cần biết thêm

##### Chế độ hỗn độn trên card mạng

"Chế độ hỗn độn" (hay còn gọi là "promiscuous mode") trên một card mạng là một trạng thái hoạt động đặc biệt của card mạng. Khi một card mạng hoạt động trong chế độ hỗn độn, nó sẽ bắt đầu nhận tất cả các gói tin trên mạng, bao gồm cả những gói tin không địa chỉ cho chính nó.

Chế độ hỗn độn thường được sử dụng cho các mục đích giám sát, phân tích mạng, hoặc để phát hiện các vấn đề về bảo mật mạng.

Kích hoạt chế độ hỗn độn trên Linux:

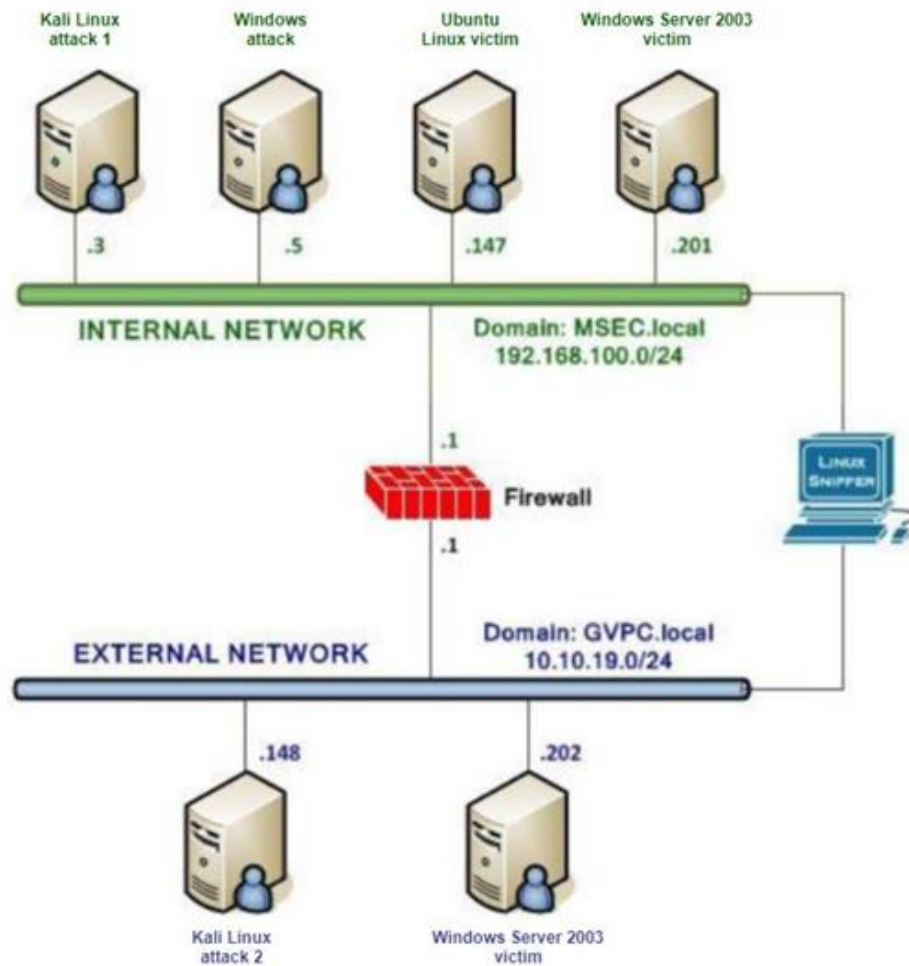
```
sudo ifconfig <tên card mạng> promisc
```

## 2.2. Nội dung thực hành

### 2.2.1. Chuẩn bị môi trường

- Phần mềm VMWare Workstation( hoặc các phần mềm hỗ trợ ảo hóa khác).
- Các file máy ảo VMware và hệ thống mạng đã cài đặt trong bài thực hành 5 trước đó: máy trạm, máy Kali Linux, máy chủ Windows và Linux. Chú ý: chỉ cần bật các máy cần sử dụng trong bài lab.
- Topo mạng như đã cấu hình trong bài 5.

## Bài 8: Bắt dữ liệu mạng



Cấu hình topo mạng

### 2.2.2. Sử dụng Tcpdump

- Đăng nhập Linux Sniffer và xem tất cả các interfaces trong hệ thống (root@bt:~#ifconfig -a)



## Bài 8: Bắt dữ liệu mạng

```
(tranhphuong@tranthithuphuong-b21dcat151)-[/home/tranhphuong]
PS> ifconfig -a
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.3 netmask 255.255.255.0 broadcast 192.168.100.255
    ether 00:0c:29:72:1a:b3 txqueuelen 1000 (Ethernet)
    RX packets 450 bytes 35373 (34.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 289 bytes 32952 (32.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.19.148 netmask 255.255.255.0 broadcast 10.10.19.255
    inet6 fe80::20c:29ff:fe72:1abd prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:72:1a:bd txqueuelen 1000 (Ethernet)
    RX packets 1 bytes 243 (243.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 796 (796.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 207 bytes 19888 (19.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 207 bytes 19888 (19.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(tranhphuong@tranthithuphuong-b21dcat151)-[/home/tranhphuong]
PS> date
Sat Mar 16 12:30:47 EDT 2024
```

*xem tất cả các interfaces trong hệ thống*

- Kích hoạt các interfaces (eth0, eth1) hoạt động ở chế độ hỗn hợp

```
(tranhphuong@tranthithuphuong-b21dcat151)-[/home/tranhphuong]
PS> sudo ifconfig eth0 promisc

(tranhphuong@tranthithuphuong-b21dcat151)-[/home/tranhphuong]
PS> sudo ifconfig eth1 promisc

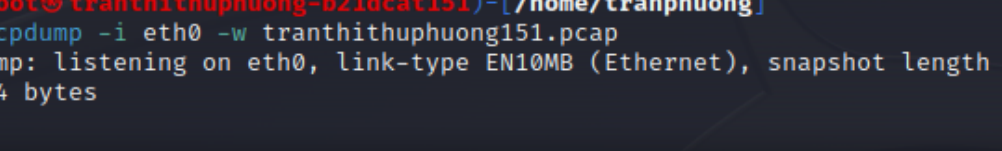
(tranhphuong@tranthithuphuong-b21dcat151)-[/home/tranhphuong]
PS> date
Sat Mar 16 12:32:24 EDT 2024
```

*Kích hoạt các interfaces (eth0, eth1) hoạt động ở chế độ hỗn hợp*

- Sau đó khởi động tcpdump. Bắt gói tin trên dải mạng 192.168.100.0/24 và gửi vào một file (thời gian chờ dữ liệu trong khoảng 5 phút).

```
(root@tranthithuphuong-b21dcat151)-[/home/tranphuong]
# tcpdump -i eth0 -w tranthithuphuong151.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length
262144 bytes

```



The image shows two terminal windows. The top window is a terminal session on a host named 'tranthithuphuong-b21dcat151' at the root user. The user has navigated to the directory '/home/tranphuong' and executed the command 'tcpdump -i eth0 -w tranthithuphuong151.pcap'. The terminal output shows 'tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes'. The bottom window is a standard Ubuntu terminal window titled 'tranphuong@tranthithuphuong-b21dcat151: ~'. It has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The terminal shows the user 'tranphuong' at the same host, in the '~' directory, typing the command 'date'. The output of the command is 'Sat Mar 16 12:46:24 EDT 2024'.

```
tranphuong@tranthithuphuong-b21dcat151: ~
File Actions Edit View Help
(tranphuong@tranthithuphuong-b21dcat151)-[~]
$ date
Sat Mar 16 12:46:24 EDT 2024

```

*Bắt gói tin trên dải mạng 192.168.100.0/24*

- Đăng nhập Window Server 2003 và tiến hành ping đến dải mạng internal và dải mạng external.

```
C:\Users\Administrator>echo "Tran Thi Thu Phuong - B21DCAT151"
"Tran Thi Thu Phuong - B21DCAT151"

C:\Users\Administrator>date
The current date is: Sat 03/16/2024
Enter the new date: (mm-dd-yy)

C:\Users\Administrator>ping 192.168.100.3

Pinging 192.168.100.3 with 32 bytes of data:
Reply from 192.168.100.3: bytes=32 time=2ms TTL=64
Reply from 192.168.100.3: bytes=32 time=1ms TTL=64
Reply from 192.168.100.3: bytes=32 time=1ms TTL=64
Reply from 192.168.100.3: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.100.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 2ms, Average = 1ms
```

*Dải Internal: 192.168.100.201 → 192.168.100.3*

```
(root@tranthithuphuong-b21dcat151)-[/home/tranphuong]
# tcpdump -i eth0
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:08:26.061128 IP 192.168.100.147 > 192.168.100.3: ICMP echo request, id 2406, seq 2706, length 64
13:08:26.061179 IP 192.168.100.3 > 192.168.100.147: ICMP echo reply, id 2406, seq 2706, length 64
13:08:27.063365 IP 192.168.100.147 > 192.168.100.3: ICMP echo request, id 2406, seq 2707, length 64
13:08:27.063414 IP 192.168.100.3 > 192.168.100.147: ICMP echo reply, id 2406, seq 2707, length 64
13:08:28.064387 IP 192.168.100.147 > 192.168.100.3: ICMP echo request, id 2406, seq 2708, length 64
13:08:28.064440 IP 192.168.100.3 > 192.168.100.147: ICMP echo reply, id 2406, seq 2708, length 64
13:08:28.215899 ARP, Request who-has 192.168.100.147 tell 192.168.100.3, length 28
13:08:28.218370 ARP, Reply 192.168.100.147 is-at 00:0c:29:67:7c:cb (oui Unknown), length 46
13:08:28.226229 ARP, Request who-has 192.168.100.3 tell 192.168.100.147, length 46
13:08:28.226262 ARP, Reply 192.168.100.3 is-at 00:0c:29:72:1a:b3 (oui Unknown), length 28
13:08:29.066598 IP 192.168.100.147 > 192.168.100.3: ICMP echo request, id 2406, seq 2709, length 64
13:08:29.066696 IP 192.168.100.3 > 192.168.100.147: ICMP echo reply, id 2406, seq 2709, length 64
```

Trên máy Linux Sniffer, bắt gói tin trên dải 192.168.100.0/24

## Bài 8: Bắt dữ liệu mạng

```
C:\Users\Administrator>echo "Tran Thi Thu Phuong - B21DCAT151"
"Tran Thi Thu Phuong - B21DCAT151"

C:\Users\Administrator>date
The current date is: Sat 03/16/2024
Enter the new date: (mm-dd-yy)

C:\Users\Administrator>ping 10.10.19.148

Pinging 10.10.19.148 with 32 bytes of data:
Reply from 10.10.19.148: bytes=32 time=2ms TTL=64
Reply from 10.10.19.148: bytes=32 time=1ms TTL=64
Reply from 10.10.19.148: bytes=32 time<1ms TTL=64
Reply from 10.10.19.148: bytes=32 time=1ms TTL=64

Ping statistics for 10.10.19.148:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 1ms
```

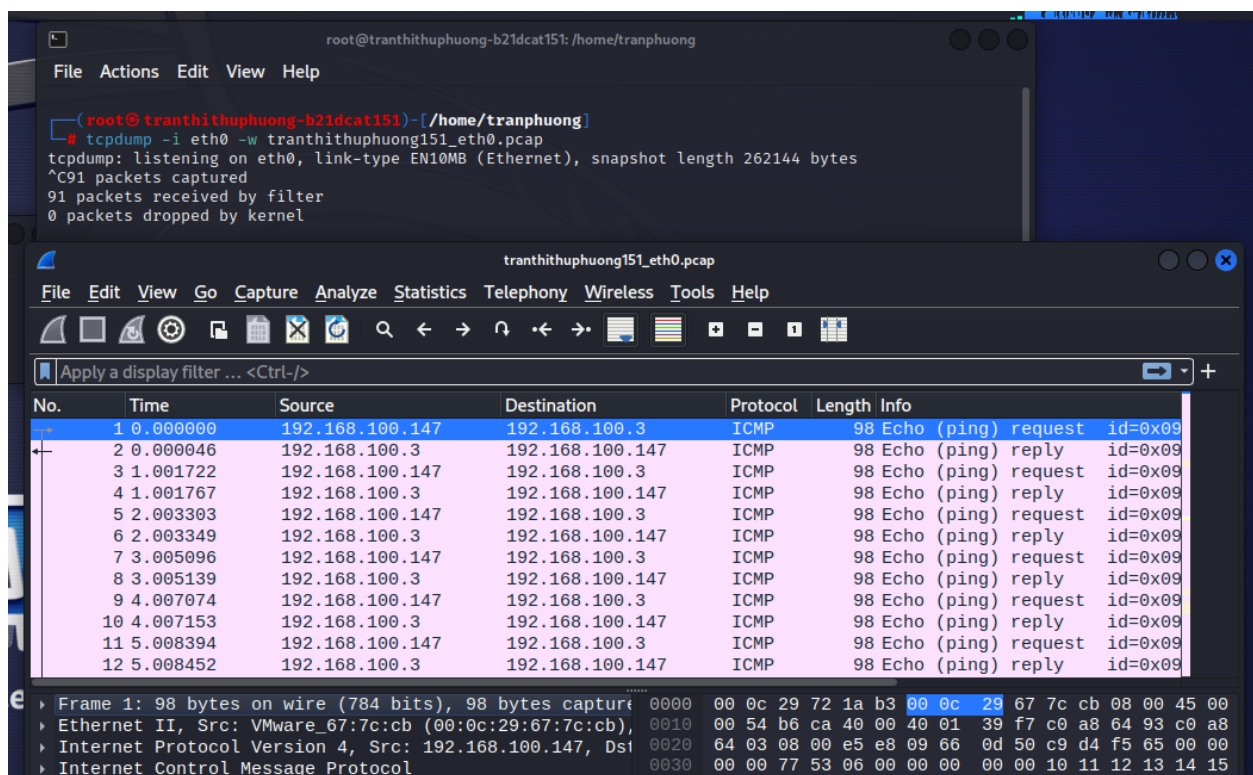
*Dải External: 10.10.19.202 → 10.10.19.148*

```
(root@tranthithuphuong-b21dcat151)-[/home/tranphuong]
# tcpdump -i eth1
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
13:10:52.207831 ARP, Request who-has 10.10.19.1 tell 10.10.19.202, length 46
13:11:26.391584 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 5, length 40
13:11:26.391637 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 5, length 40
13:11:27.426380 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 6, length 40
13:11:27.426424 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 6, length 40
13:11:28.458050 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 7, length 40
13:11:28.458096 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 7, length 40
13:11:29.489176 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 8, length 40
13:11:29.489226 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 8, length 40
13:11:30.364222 ARP, Request who-has 10.10.19.1 tell 10.10.19.202, length 46
13:11:31.207461 ARP, Request who-has 10.10.19.148 (00:0c:29:72:1a:bd (oui Unknown)) tell 10.10.19.202, length 46
13:11:31.207463 ARP, Request who-has 10.10.19.1 tell 10.10.19.202, length 46
13:11:31.207486 ARP, Reply 10.10.19.148 is-at 00:0c:29:72:1a:bd (oui Unknown), length 28
13:11:31.511681 ARP, Request who-has 10.10.19.202 tell 10.10.19.148, length 28
13:11:31.513075 ARP, Reply 10.10.19.202 is-at 00:0c:29:13:17:3c (oui Unknown), length 46
13:11:32.207118 ARP, Request who-has 10.10.19.1 tell 10.10.19.202, length 46
```

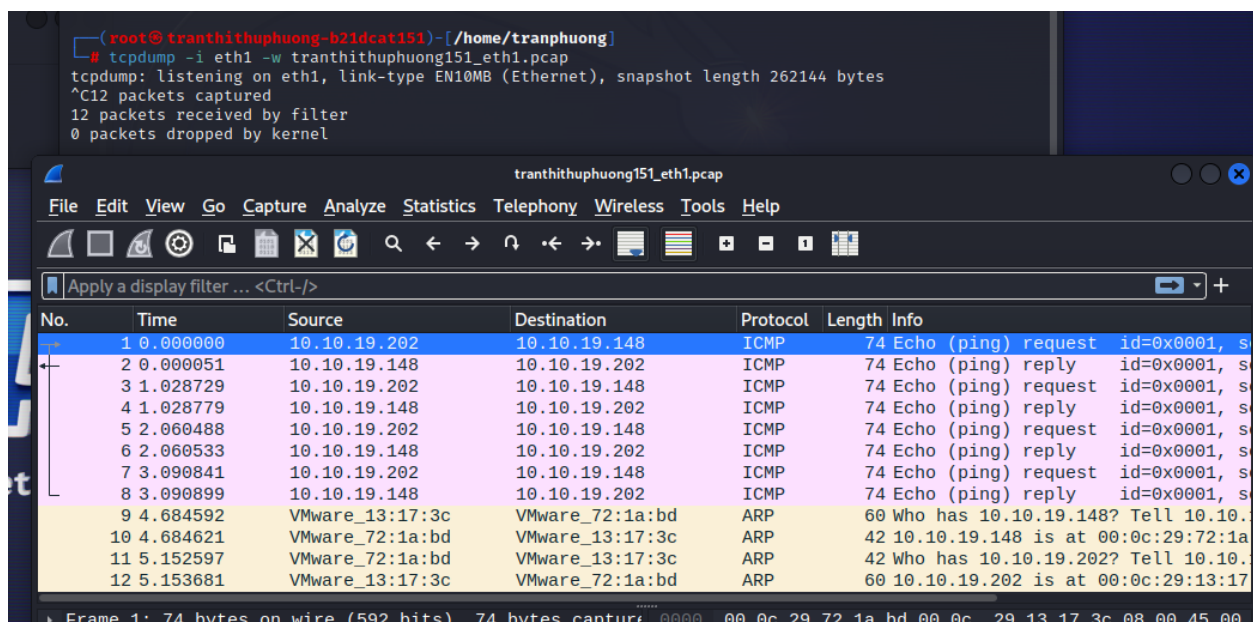
*Trên máy Linux Sniffer, bắt gói tin trên dải 10.10.19.0/24*

- Trên máy Linux Sniffer, tiến hành bắt gói tin bằng tcpdump, và lưu dữ liệu vào file pcap.

## Bài 8: Bắt dữ liệu mạng



*Các dữ liệu đã bắt trên dải Internal*



*Các dữ liệu đã bắt trên dải External*

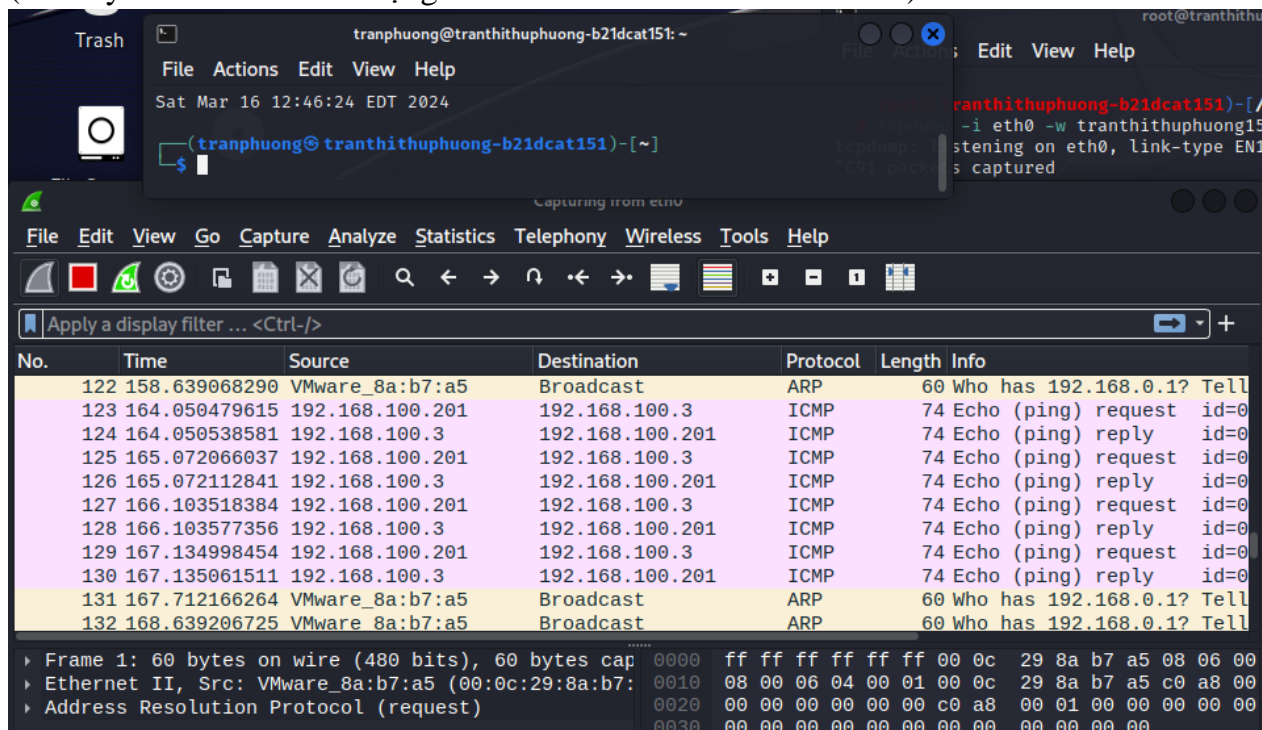
### 2.2.3. Sử dụng Wireshark để bắt và phân tích các gói tin

- Có thể tải Wireshark ở đây: <http://www.wireshark.org/download.html>

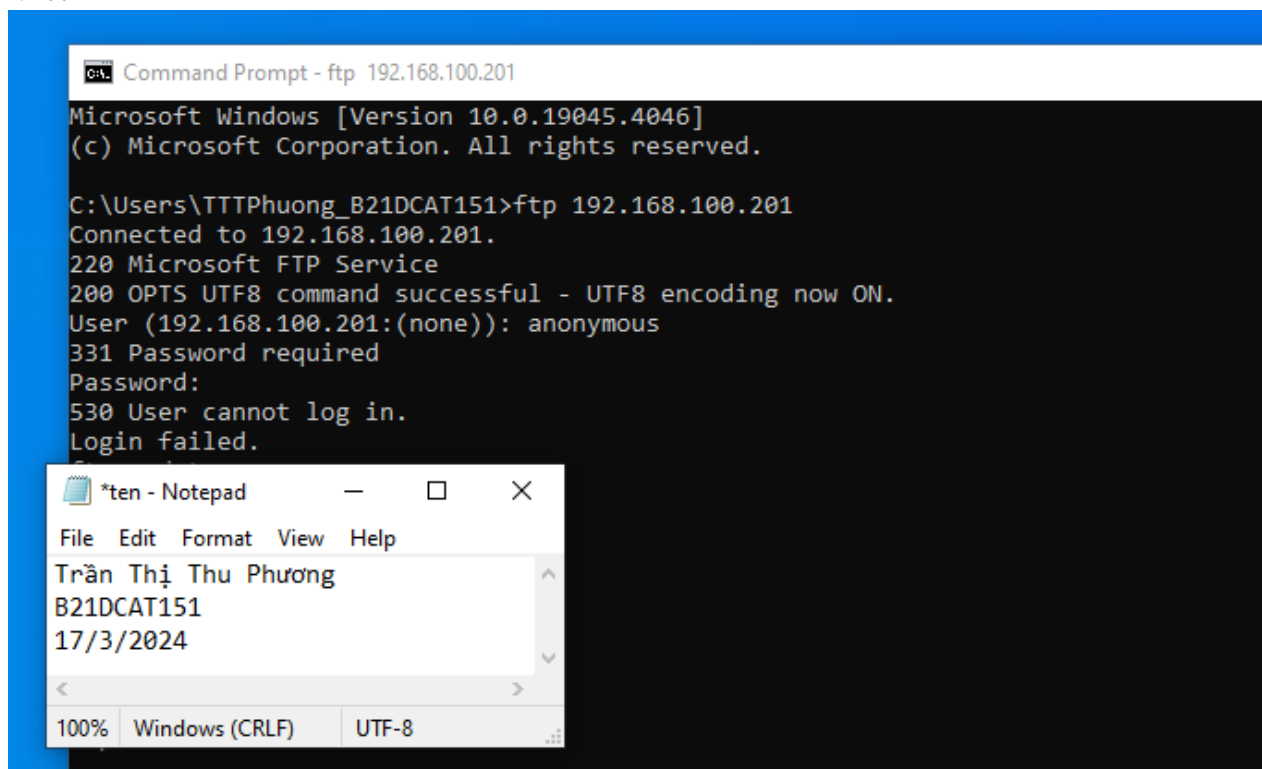


## Bài 8: Bắt dữ liệu mạng

- Trên máy Linux Sniffer, bật các interfaces eth0, eth1 và khởi động Wireshark. Trong Capture Interfaces chọn Start ở dòng eth0 để bắt gói tin trên dải mạng 192.168.100.0 (do máy em cấu hình dải mạng 192.168.100.0 trên interface eth0)



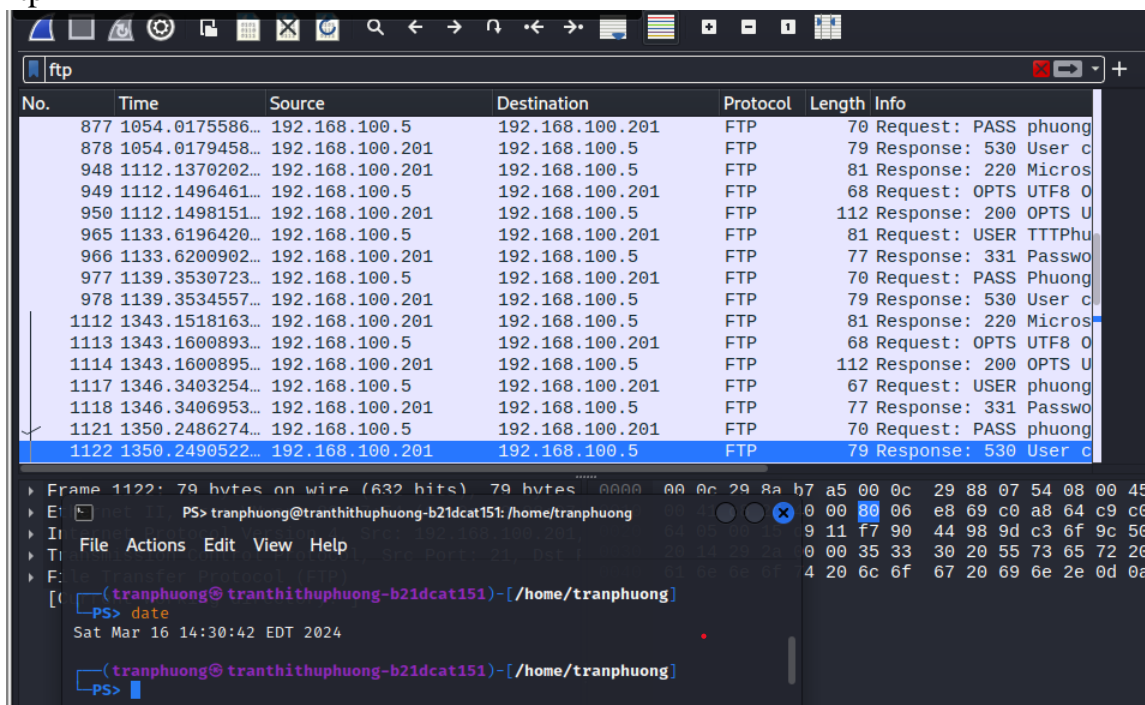
- Trên máy Windows 7 Attack kết nối tới ftp server trên máy Window Server Internal Victim



*Windows 7 Attack kết nối tới ftp server trên máy Window Server Internal*

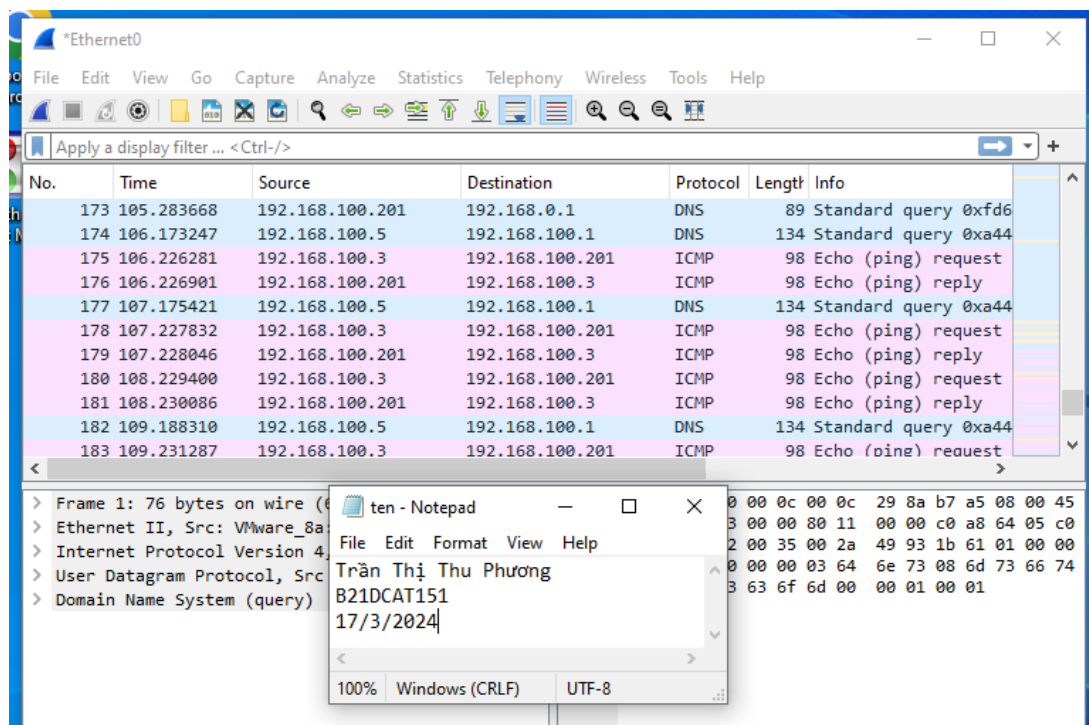
## Bài 8: Bắt dữ liệu mạng

- Trên Linux Sniffer dùng quá trình bắt gói tin và tiến hành lọc gói tin theo giao thức ftp



### Lọc gói tin theo giao thức ftp

- Trên máy Windows attack (192.168.100.5), trong Capture Interfaces chọn Start ở dòng eth1 để bắt gói tin trên dải mạng 192.168.100.0 (khi ping từ máy 192.168.100.3 đến máy 192.168.100.201)



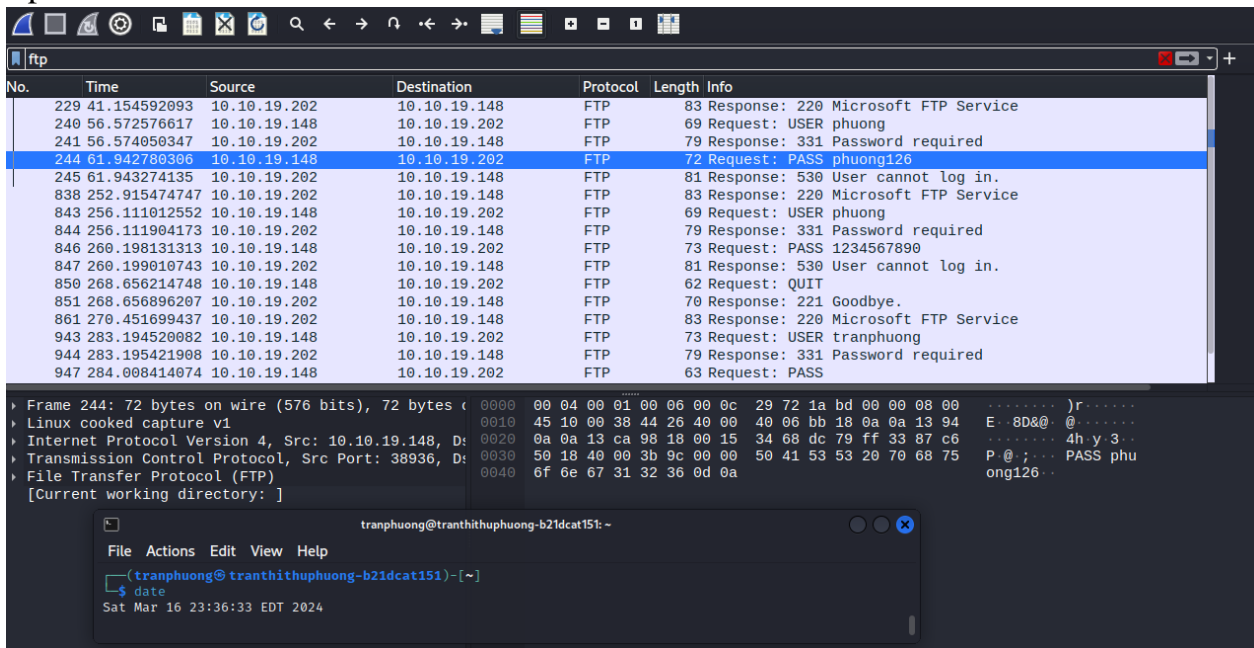
## Bài 8: Bắt dữ liệu mạng

- Trên máy Kali Linux Attack External, kết nối với ftp server(root@bt:~#ftp 10.10.19.202)

```
(tranhithuphuong@tranhithuphuong-b21dcat151)-[~] 19.148
$ ftp 10.10.19.202
Connected to 10.10.19.202.
220 Microsoft FTP Service
Name (10.10.19.202:tranhithuphuong): phuong
331 Password required
Password:
530 User cannot log in.
ftp: Login failed
ftp> get
(remote-file) tranthithuphuong151
(local-file) C:\Users\Administrator\Documents
local: C:\Users\Administrator\Documents remote: tranthithuphuong151
421 Service not available, remote server has closed connection.
530 User cannot log in.
ftp> bye
```

*Trên máy Kali Linux External kết nối ftp đến máy Ftp Server*

- Trên Linux Sniffer dùng quá trình bắt gói tin và tiến hành lọc gói tin theo giao thức ftp



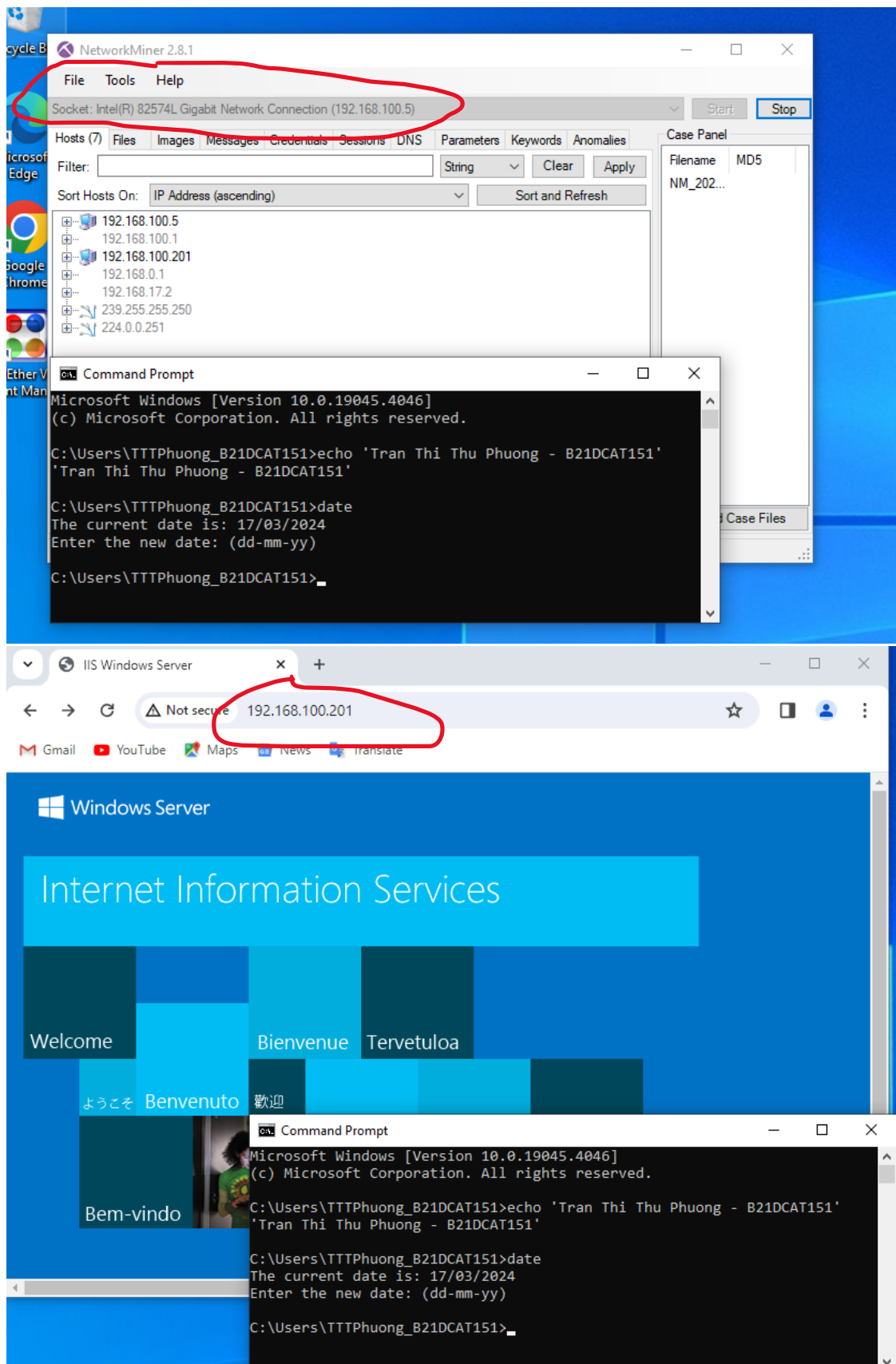
The screenshot shows a network sniffer interface with a table of captured packets. The selected packet (No. 244) is an FTP request for the file 'phuong126'. Below the table, the packet details are shown, including the FTP protocol structure and the file name 'phuong126'.

*Bắt gói tin trên dải 10.10.19.0/24 và lọc theo giao thức ftp*

### 2.2.4. Sử dụng Network Miner để bắt và phân tích các gói tin

- Trên máy Windows 10 Internal Attack khởi động Network Miner và chọn Socket: Intel® PRO/1000MT Network Connection(192.168.100.5) và bắt đầu bắt gói tin. Sử dụng Internet Explorer để kết nối đến trang web của Windows 2003 Server Internal Victim: <http://192.168.100.201/>. Sau đó dùng quá trình bắt gói tin.

## Bài 8: Bắt dữ liệu mạng

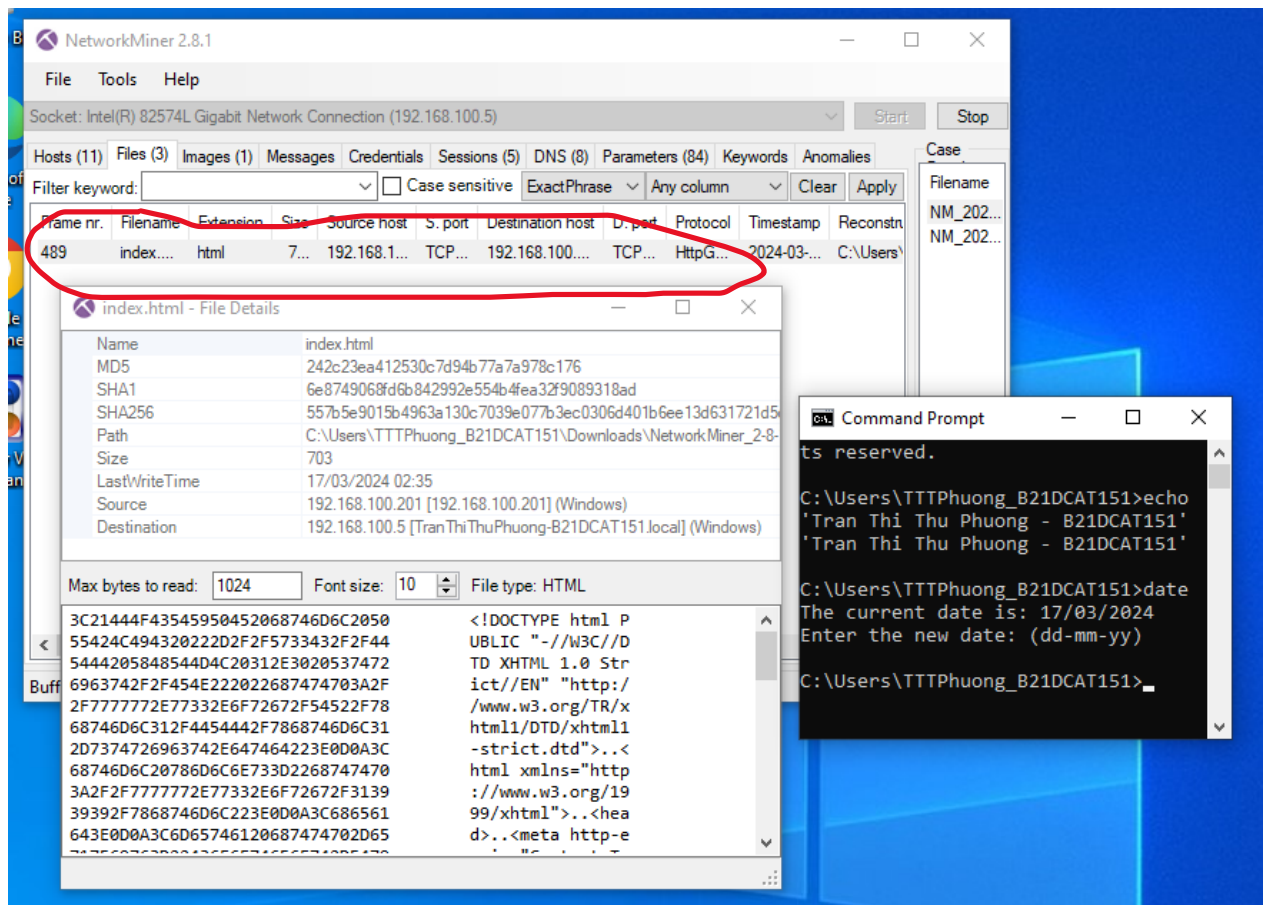


*Kết nối đến trang web của Windows 2003 Server Internal*



## Bài 8: Bắt dữ liệu mạng

- Trong Network Miner, chọn File/ index.html để xem dữ liệu gói tin vừa bắt được.



*Chọn File/ index.html để xem dữ liệu gói tin vừa bắt được.*

### 3. Kết luận

- Thu được kết quả bắt gói tin và các file pcap thông qua tcpdump
- Sử dụng Wireshark để bắt và lọc ra được các gói tin ftp, các file pcap tương ứng
- Bắt được các dữ liệu trong file index.html.

### 4. Tài liệu tham khảo

- [1]. Chương 4, Bài giảng Kỹ thuật theo dõi giám sát an toàn mạng, HVCN BCVT 2021
- [2]. <https://www.tcpdump.org/index.html#documentation>
- [3]. [https://www.wireshark.org/docs/wsug\\_html/](https://www.wireshark.org/docs/wsug_html/)
- [4]. <https://docs.securityonion.net/en/2.3/networkminer.html#>