

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

-----



**MÔN HỌC: THỰC TẬP CƠ SỞ**  
**BÁO CÁO BÀI THỰC HÀNH SỐ 8**

**Giảng viên hướng dẫn : PGS.TS Hoàng Xuân Dậu**  
**Sinh viên thực hiện : Nguyễn Nhật Minh**  
**Mã sinh viên : B21DCAT132**

**Hà Nội, tháng 3 năm 2024**

# Môn học Thực tập cơ sở

## Bài 8: Bắt dữ liệu mạng

### 1. Mục đích

Sử dụng tcpdump để bắt gói tin mạng  
Sử dụng wireshark để bắt và phân tích gói tin mạng  
Sử dụng Network Miner để bắt và phân tích gói tin mạng

### 2. Tìm hiểu lý thuyết

#### 2.1. Tcpdump

Tcpdump là công cụ hữu ích được ra đời và phát triển để phục vụ cho mục đích hỗ trợ phân tích các gói dữ liệu mạng theo dòng lệnh đồng thời cho phép khách hàng thực hiện việc chặn, lọc và hiển thị các gói tin TCP/IP được truyền đi hoặc nhận trên một mạng có sự tham gia của máy tính.

Một số lợi ích của tcpdump:

- Hỗ trợ xem các bản tin dump trên terminal
- Capture các bản tin và lưu dưới dạng .pcap(hỗ trợ đọc bởi wireshark)
- Hỗ trợ xem trực tiếp các bản tin điều khiển hệ thống linux thông qua wireshark

Tcpdump là công cụ có khả năng capturing packets mạnh mẽ. Hoạt động trên network layer, tcpdump có thể capture tất cả các gói ra vào máy tính.

Tcpdump sẽ xuất ra màn hình nội dung các gói tin chạy trên card nhà mạng mà máy chủ đang lắng nghe sao cho phù hợp với biểu thức logic chọn lọc mà khách hàng đã sử dụng và nhập vào máy tính. Khách hàng có thể xuất ra các mô tả về gói tin thành một file pcap để phân tích sau dựa trên từng loại tùy chọn khác nhau. Để đọc được nội dung của file pcap này, bạn chỉ cần sử dụng các phần mềm khác như Wireshark hay với option -r của tcpdump

Trong các trường hợp không có tùy chọn nào, tcpdump sẽ vẫn tiếp tục chạy cho đến khi nó nhận được tín hiệu ngắt từ phía khách hàng. Sau khi việc bắt các gói tin kết thúc, tcpdump sẽ đưa ra các báo cáo sau:

Packet capture: Số lượng các gói tin đã bắt được và tiến hành xử lý.

Packet received by filter: Số lượng các gói tin mà bộ lọc nhận được.

Packet dropped by kernel: số lượng packet bị dropped do cơ chế bắt gói tin.

#### 2.2. Wireshark

Wireshark là một ứng dụng dùng để bắt (capture), phân tích và xác định các vấn đề

liên quan đến network như: rớt gói tin, kết nối chậm, hoặc các truy cập bất thường. Phần mềm này cho phép quản trị viên hiểu sâu hơn các Network Packets đang chạy trên hệ thống, qua đó dễ dàng xác định các nguyên nhân chính xác gây ra lỗi

Sử dụng Wireshark có thể capture các packet trong thời gian thực (real time), lưu trữ chúng lại và phân tích chúng offline. Ngoài ra, nó cũng bao gồm các filter, color coding và nhiều tính năng khác, cho phép người dùng tìm hiểu sâu hơn về lưu lượng mạng cũng như inspect (kiểm tra) các packets.

Ứng dụng được viết bằng ngôn ngữ C và hệ điều hành Cross-platform, ngoài ra hiện nay gồm có các bản phân phối Linux, Windows, OS X, FreeBSD, NetBSD và OpenBSD. Đây là một phần mềm mã nguồn mở, được cấp phép GPL, và do đó miễn phí sử dụng, tự do chia sẻ, sửa đổi.

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.

### 2.3. Network Miner

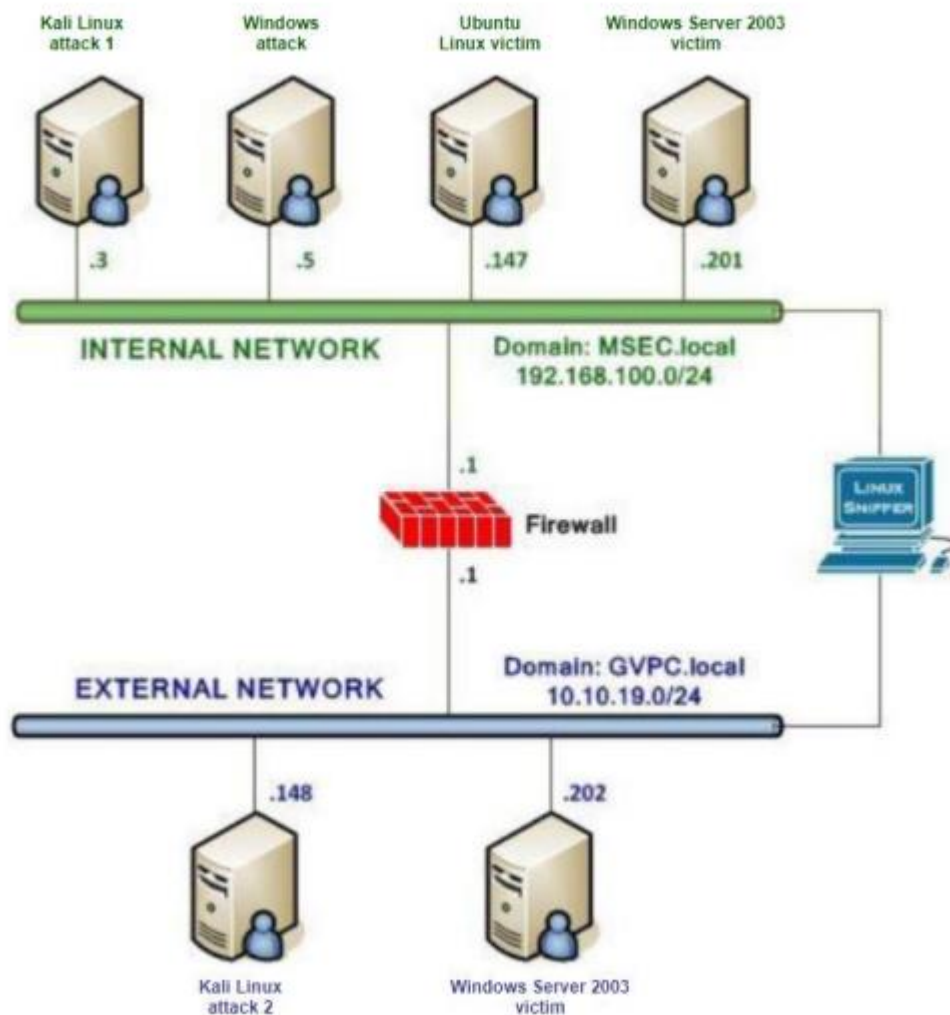
NetworkMiner là công cụ giám sát mạng mã nguồn mở dành cho hệ điều hành Window. Công cụ này cũng được hỗ trợ để cài đặt trên Linux, Mac OS X và FreeBSD. Hiện nay có rất nhiều công cụ giám sát mạng khác nhau, tuy nhiên NetworkMiner vẫn được sử dụng khá phổ biến. Những điểm nổi bật của NetworkMiner phải 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 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.

Nếu bạn đang lo lắng rằng máy tính của mình đang bị kẻ xấu thu thập thông tin từ xa qua các phần mềm gián điệp, ... hãy thử tải và sử dụng NetworkMiner, mọi thiết bị và trang web vừa kết nối thông tin với máy tính của bạn đều sẽ nhanh chóng bị phát hiện.

### 3. Chuẩn bị môi trường



Cấu hình topo mạng đã dùng ở bài thực hành 5

#### 4. Thực hành

##### 4.1. Sử dụng tcpdump bắt gói tin

- Máy linux sniffer có 2 card mạng:  
Eth0 thuộc dải 192.168.100.0/24  
Eth1 thuộc dải 10.10.19.0/24

```
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.100.10 netmask 255.255.255.0 broadcast 192.168.100.255
    inet6 fe80::23d6:1ae:4b3e:c490 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:8e:21:c4 txqueuelen 1000 (Ethernet)
    RX packets 8 bytes 719 (719.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 37 bytes 3954 (3.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.10.19.10 netmask 255.255.255.0 broadcast 10.10.19.255
    inet6 fe80::e3cf:6e18:f533:45dd prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:8e:21:ce txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 2890 (2.8 KiB)
    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 20 bytes 1584 (1.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 20 bytes 1584 (1.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Sử dụng lệnh ifconfig eth0/eth1 promisc để kích hoạt interfaces hoạt động ở chế độ hỗn hợp

```
(kali㉿kali)-[~]
$ sudo ifconfig eth0 promisc
[sudo] password for kali:

(kali㉿kali)-[~]
$ sudo ifconfig eth1 promisc

(kali㉿kali)-[~]
$ nguyen nhat minh b21dcat132
```

- Sử dụng lệnh tcpdump -I eth0 icmp để bật tcpdump bắt gói tin trên dải mạng 192.168.100.0/24
- Trên máy windows server 2016 ping tới các mạng trong internal

```
Ping statistics for 192.168.100.5:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 3ms, Average = 1ms
```

```
C:\Users\Administrator>ping 192.168.100.10
```

```
Pinging 192.168.100.10 with 32 bytes of data:  
Reply from 192.168.100.10: bytes=32 time=1ms TTL=64  
Reply from 192.168.100.10: bytes=32 time<1ms TTL=64  
Reply from 192.168.100.10: bytes=32 time=2ms TTL=64  
Reply from 192.168.100.10: bytes=32 time=1ms TTL=64
```

```
Ping statistics for 192.168.100.10:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 2ms, Average = 1ms
```

```
C:\Users\Administrator>ping 192.168.100.147
```

```
Pinging 192.168.100.147 with 32 bytes of data:  
Reply from 192.168.100.147: bytes=32 time<1ms TTL=64  
Reply from 192.168.100.147: bytes=32 time<1ms TTL=64  
Reply from 192.168.100.147: bytes=32 time<1ms TTL=64  
Reply from 192.168.100.147: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 192.168.100.147:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

```
C:\Users\Administrator>b21dcat132 nguyen nhat minh_
```

File Actions Edit View Help

└─\$ sudo tcpdump -i eth0 icmp

[sudo] password for kali:

Sorry, try again.

[sudo] password for kali:

tcpdump: verbose output suppressed, use -v[v]... for full protocol decode

listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes

03:12:17.045154 IP 192.168.100.201 > 192.168.100.147: ICMP echo request, id 1, seq 22, length 40

03:12:17.045731 IP 192.168.100.147 > 192.168.100.201: ICMP echo reply, id 1, seq 22, length 40

03:12:18.060811 IP 192.168.100.201 > 192.168.100.147: ICMP echo request, id 1, seq 23, length 40

03:12:18.061248 IP 192.168.100.147 > 192.168.100.201: ICMP echo reply, id 1, seq 23, length 40

03:12:19.074317 IP 192.168.100.201 > 192.168.100.147: ICMP echo request, id 1, seq 24, length 40

03:12:19.074572 IP 192.168.100.147 > 192.168.100.201: ICMP echo reply, id 1, seq 24, length 40

03:12:20.091522 IP 192.168.100.201 > 192.168.100.147: ICMP echo request, id 1, seq 25, length 40

03:12:20.092118 IP 192.168.100.147 > 192.168.100.201: ICMP echo reply, id 1, seq 25, length 40

03:13:14.600668 IP 192.168.100.201 > 192.168.100.10: ICMP echo request, id 1, seq 26, length 40

03:13:14.601458 IP 192.168.100.10 > 192.168.100.201: ICMP echo reply, id 1, seq 26, length 40

03:13:15.618669 IP 192.168.100.201 > 192.168.100.10: ICMP echo request, id 1, seq 27, length 40

03:13:15.618715 IP 192.168.100.10 > 192.168.100.201: ICMP echo reply, id 1, seq 27, length 40

03:13:16.632733 IP 192.168.100.201 > 192.168.100.10: ICMP echo request, id 1, seq 28, length 40

03:13:16.632755 IP 192.168.100.10 > 192.168.100.201: ICMP echo reply, id 1, seq 28, length 40

03:13:17.675393 IP 192.168.100.201 > 192.168.100.10: ICMP echo request, id 1, seq 29, length 40

03:13:17.675419 IP 192.168.100.10 > 192.168.100.201: ICMP echo reply, id 1, seq 29, length 40

03:19:08.823231 IP 192.168.100.201 > 192.168.100.5: ICMP echo request, id 1, seq 30, length 40

03:19:08.823518 IP 192.168.100.5 > 192.168.100.201: ICMP echo reply, id 1, seq 30, length 40

03:19:09.851676 IP 192.168.100.201 > 192.168.100.5: ICMP echo request, id 1, seq 31, length 40

03:19:09.852013 IP 192.168.100.5 > 192.168.100.201: ICMP echo reply, id 1, seq 31, length 40

03:19:10.881446 IP 192.168.100.201 > 192.168.100.5: ICMP echo request, id 1, seq 32, length 40

03:19:10.881623 IP 192.168.100.5 > 192.168.100.201: ICMP echo reply, id 1, seq 32, length 40

03:19:11.903614 IP 192.168.100.201 > 192.168.100.5: ICMP echo request, id 1, seq 33, length 40

03:19:11.903784 IP 192.168.100.5 > 192.168.100.201: ICMP echo reply, id 1, seq 33, length 40

^C

24 packets captured

24 packets received by filter

0 packets dropped by kernel

- Tương tự, phía bên external, ping từ windows server và dùng lệnh  
Tcpdump -i eth1 icmp để bắt gói tin

```
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<1ms TTL=128
Reply from 10.10.19.148: bytes=32 time<1ms TTL=128
Reply from 10.10.19.148: bytes=32 time<1ms TTL=128
Reply from 10.10.19.148: bytes=32 time=1ms TTL=128

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 = 1ms, Average = 0ms

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<1ms TTL=128
Reply from 10.10.19.148: bytes=32 time<1ms TTL=128
Reply from 10.10.19.148: bytes=32 time=2ms TTL=128
Reply from 10.10.19.148: bytes=32 time<1ms TTL=128

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 = 0ms

C:\Users\Administrator>nguyen nhath b21dcat132_
```



```
(kali㉿kali)-[~]
└─$ tcpdump -i eth1 icmp
tcpdump: eth1: You don't have permission to perform this capture on that device
(socket: Operation not permitted)

(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth1 icmp
[sudo] password for kali:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
04:32:49.963028 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 5, length 40
04:32:49.963215 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 5, length 40
04:32:50.978533 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 6, length 40
04:32:50.978692 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 6, length 40
04:32:51.993984 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 7, length 40
04:32:51.994976 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 7, length 40
04:32:53.010560 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 8, length 40
04:32:53.010822 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 8, length 40
04:32:57.825333 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 9, length 40
04:32:57.825334 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 9, length 40
04:32:58.837853 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 10, length 40
04:32:58.838132 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 10, length 40
04:32:59.886468 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 11, length 40
04:32:59.886470 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 11, length 40
04:33:00.913748 IP 10.10.19.202 > 10.10.19.148: ICMP echo request, id 1, seq 12, length 40
04:33:00.914043 IP 10.10.19.148 > 10.10.19.202: ICMP echo reply, id 1, seq 12, length 40
```

- Để bắt gói tin và lưu vào file pcap sử dụng lệnh:

`tcpdump -i eth0 -w b21dcat132.pcap`

```
(kali㉿kali)-[~]
└─$ sudo tcpdump -i eth0 -w b21dcat132.pcap
[sudo] password for kali:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C438 packets captured
438 packets received by filter
0 packets dropped by kernel

(kali㉿kali)-[~]
└─$ ls
b21dcat132.pcap  Desktop  Downloads  Music  Public  ret  Templates  Videos
chall.S         Documents  minh.txt  Pictures  resolv.conf  svchost.exe  tunn3l_v1s10n

(kali㉿kali)-[~]
└─$ date
Fri Mar 15 03:33:14 AM EDT 2024

(kali㉿kali)-[~]
└─$
```

- Câu lệnh sẽ bắt tất cả các gói tin trên giao diện mạng
- Để lựa chọn các gói tin icmp thì ta thêm tùy chọn icmp vào lệnh

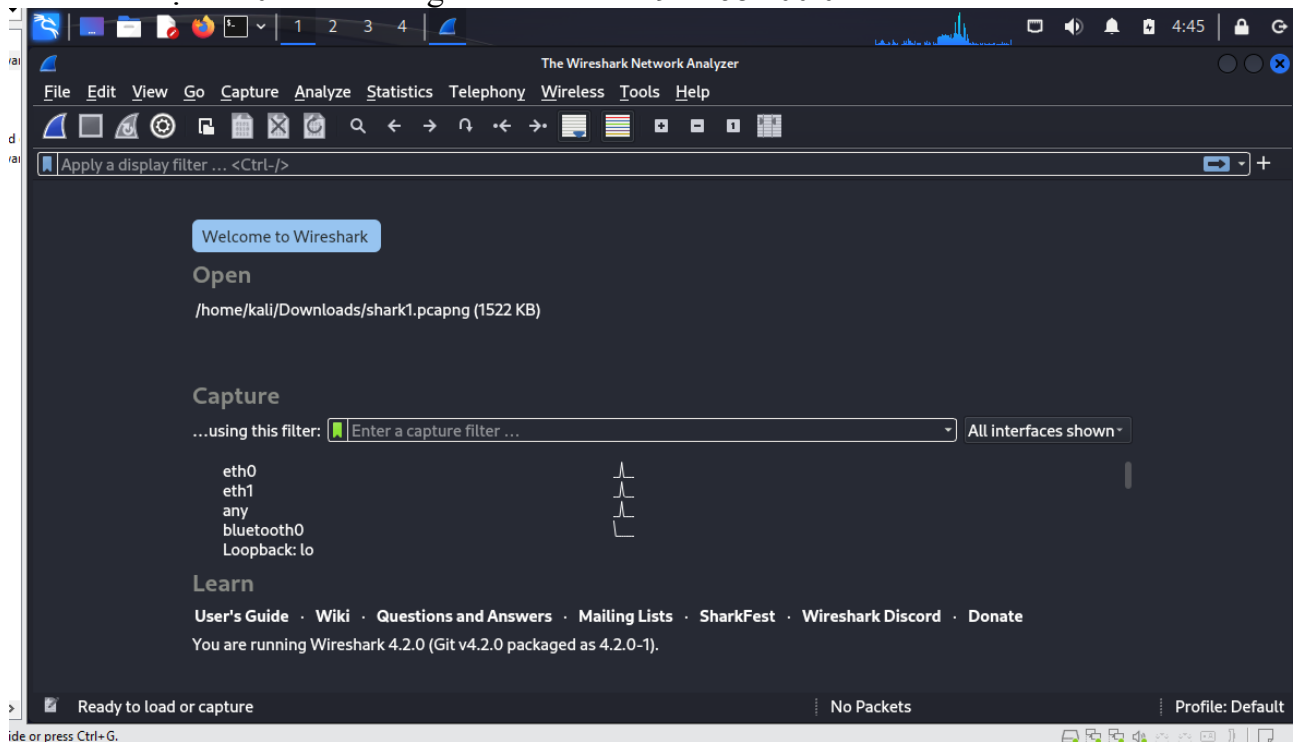
```
(kali㉿kali)-[~]  
$ sudo tcpdump -i eth0 icmp -w minh_b21dcat132.pcap  
  
[sudo] password for kali:  
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes  
^C24 packets captured  
24 packets received by filter  
0 packets dropped by kernel
```

- Sau khi ping lại 1 lần nữa, ta mở file và check được kết quả đã lưu thành công

```
(kali㉿kali)-[~]  
$ tcpdump -r minh_b21dcat132.pcap  
reading from file minh_b21dcat132.pcap, link-type EN10MB (Ethernet), snapshot length 262144  
04:17:44.916755 IP 192.168.100.201 > 192.168.100.5: ICMP echo request, id 1, seq 46, length 40  
04:17:44.917644 IP 192.168.100.5 > 192.168.100.201: ICMP echo reply, id 1, seq 46, length 40  
04:17:45.954987 IP 192.168.100.201 > 192.168.100.5: ICMP echo request, id 1, seq 47, length 40  
04:17:45.954988 IP 192.168.100.5 > 192.168.100.201: ICMP echo reply, id 1, seq 47, length 40  
04:17:46.965181 IP 192.168.100.201 > 192.168.100.5: ICMP echo request, id 1, seq 48, length 40  
04:17:46.965182 IP 192.168.100.5 > 192.168.100.201: ICMP echo reply, id 1, seq 48, length 40  
04:17:47.974281 IP 192.168.100.201 > 192.168.100.5: ICMP echo request, id 1, seq 49, length 40  
04:17:47.977530 IP 192.168.100.5 > 192.168.100.201: ICMP echo reply, id 1, seq 49, length 40  
04:17:52.533795 IP 192.168.100.201 > 192.168.100.10: ICMP echo request, id 1, seq 50, length 40  
04:17:52.535026 IP 192.168.100.10 > 192.168.100.201: ICMP echo reply, id 1, seq 50, length 40  
04:17:53.555118 IP 192.168.100.201 > 192.168.100.10: ICMP echo request, id 1, seq 51, length 40  
04:17:53.555145 IP 192.168.100.10 > 192.168.100.201: ICMP echo reply, id 1, seq 51, length 40  
04:17:54.571847 IP 192.168.100.201 > 192.168.100.10: ICMP echo request, id 1, seq 52, length 40  
04:17:54.571871 IP 192.168.100.10 > 192.168.100.201: ICMP echo reply, id 1, seq 52, length 40  
04:17:55.604655 IP 192.168.100.201 > 192.168.100.10: ICMP echo request, id 1, seq 53, length 40  
04:17:55.604714 IP 192.168.100.10 > 192.168.100.201: ICMP echo reply, id 1, seq 53, length 40  
04:18:16.407998 IP 192.168.100.201 > 192.168.100.147: ICMP echo request, id 1, seq 54, length 40  
04:18:16.409062 IP 192.168.100.147 > 192.168.100.201: ICMP echo reply, id 1, seq 54, length 40  
04:18:17.425769 IP 192.168.100.201 > 192.168.100.147: ICMP echo request, id 1, seq 55, length 40  
04:18:17.425979 IP 192.168.100.147 > 192.168.100.201: ICMP echo reply, id 1, seq 55, length 40  
04:18:18.444128 IP 192.168.100.201 > 192.168.100.147: ICMP echo request, id 1, seq 56, length 40  
04:18:18.444299 IP 192.168.100.147 > 192.168.100.201: ICMP echo reply, id 1, seq 56, length 40  
04:18:19.458008 IP 192.168.100.201 > 192.168.100.147: ICMP echo request, id 1, seq 57, length 40  
04:18:19.458247 IP 192.168.100.147 > 192.168.100.201: ICMP echo reply, id 1, seq 57, length 40  
  
(kali㉿kali)-[~]
```

## 4.2. Sử dụng wireshark bắt gói tin

- Trên máy linux sniffer mở Wireshark, giao diện hiển thị 2 interfaces eth0 và eth1
- Chọn eth0 để bắt các gói tin trên dải 192.168.100.0/24



- Trên windows 7 tiến hành truy cập ftp tới windows server 2016 qua ip 192.168.100.201

```
C:\Windows\system32\cmd.exe - ftp 192.168.100.201

Pinging 192.168.100.201 with 32 bytes of data:
Reply from 192.168.100.201: bytes=32 time<1ms TTL=128
Reply from 192.168.100.201: bytes=32 time<1ms TTL=128
Reply from 192.168.100.201: bytes=32 time<1ms TTL=128
Reply from 192.168.100.201: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.100.201:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\b21dcat132>ftp 192.168.100.201
Connected to 192.168.100.201.
220 Microsoft FTP Service
User (192.168.100.201:(none)): user administrator
331 Password required
Password:
530 User cannot log in.
Login failed.
ftp> user Administrator
331 Password required
Password:
230 User logged in.
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp> whoami
Invalid command.
ftp> nguyen nhath minh b21dcat132_
```

- Trên máy linux sniffer lọc và thu được các gói tin ftp

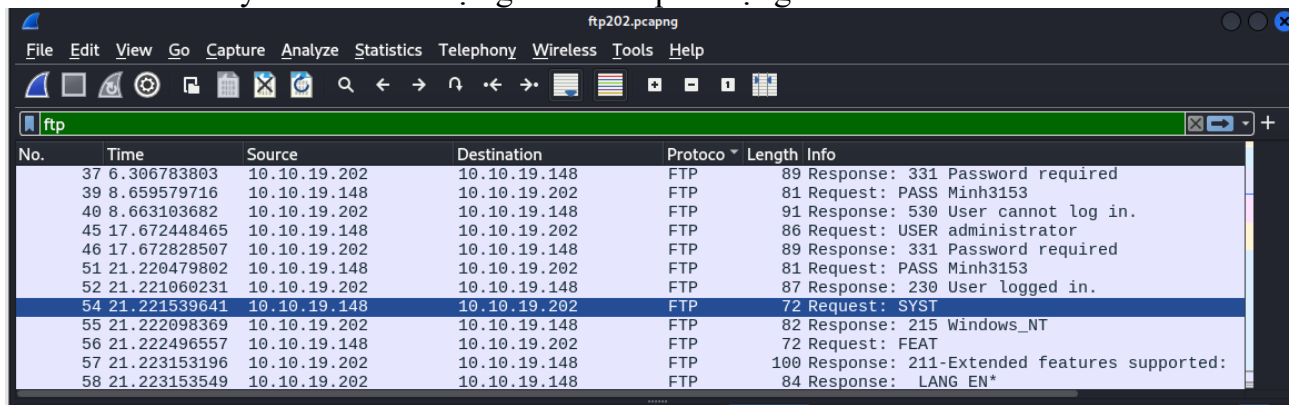
in 10 and  
t.1-vmwa  
er 2016

No.	Time	Source	Destination	Protocol	Length	Info
25	16.003005972	192.168.100.201	192.168.100.5	FTP	81	Response: 220 Microsoft FTP Service
57	23.431974321	192.168.100.5	192.168.100.201	FTP	79	Request: USER user administrator
58	23.433440102	192.168.100.201	192.168.100.5	FTP	77	Response: 331 Password required
61	26.055916004	192.168.100.5	192.168.100.201	FTP	69	Request: PASS Minh3153
62	26.061056407	192.168.100.201	192.168.100.5	FTP	79	Response: 530 User cannot log in.
69	39.348074682	192.168.100.5	192.168.100.201	FTP	74	Request: USER Administrator
70	39.348450381	192.168.100.201	192.168.100.5	FTP	77	Response: 331 Password required
75	42.267641565	192.168.100.5	192.168.100.201	FTP	69	Request: PASS Minh3153
76	42.271651799	192.168.100.201	192.168.100.5	FTP	75	Response: 230 User logged in.
81	44.187229354	192.168.100.5	192.168.100.201	FTP	81	Request: PORT 192,168,100,5,192,15
83	44.189936281	192.168.100.201	192.168.100.5	FTP	84	Response: 200 PORT command successful.
86	44.192835359	192.168.100.5	192.168.100.201	FTP	60	Request: NLST
87	44.193331937	192.168.100.201	192.168.100.5	FTP	108	Response: 125 Data connection already open; Tr
90	44.193823377	192.168.100.201	192.168.100.5	FTP	78	Response: 226 Transfer complete.

- Trên máy kali external kết nối ftp tới windows server qua ip 10.10.19.202

```
(kali@minhb21dcat132)-[~]
$ ftp 10.10.19.202
Connected to 10.10.19.202.
220 Microsoft FTP Service
Name (10.10.19.202:kali): user Administrator
331 Password required
Password:
530 User cannot log in.
ftp: Login failed
ftp> user administrator
331 Password required
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> systeminfo
?Invalid command.
ftp> ls
229 Entering Extended Passive Mode (|||49672|)
125 Data connection already open; Transfer starting.
226 Transfer complete.
ftp> bye
221 Goodbye.
```

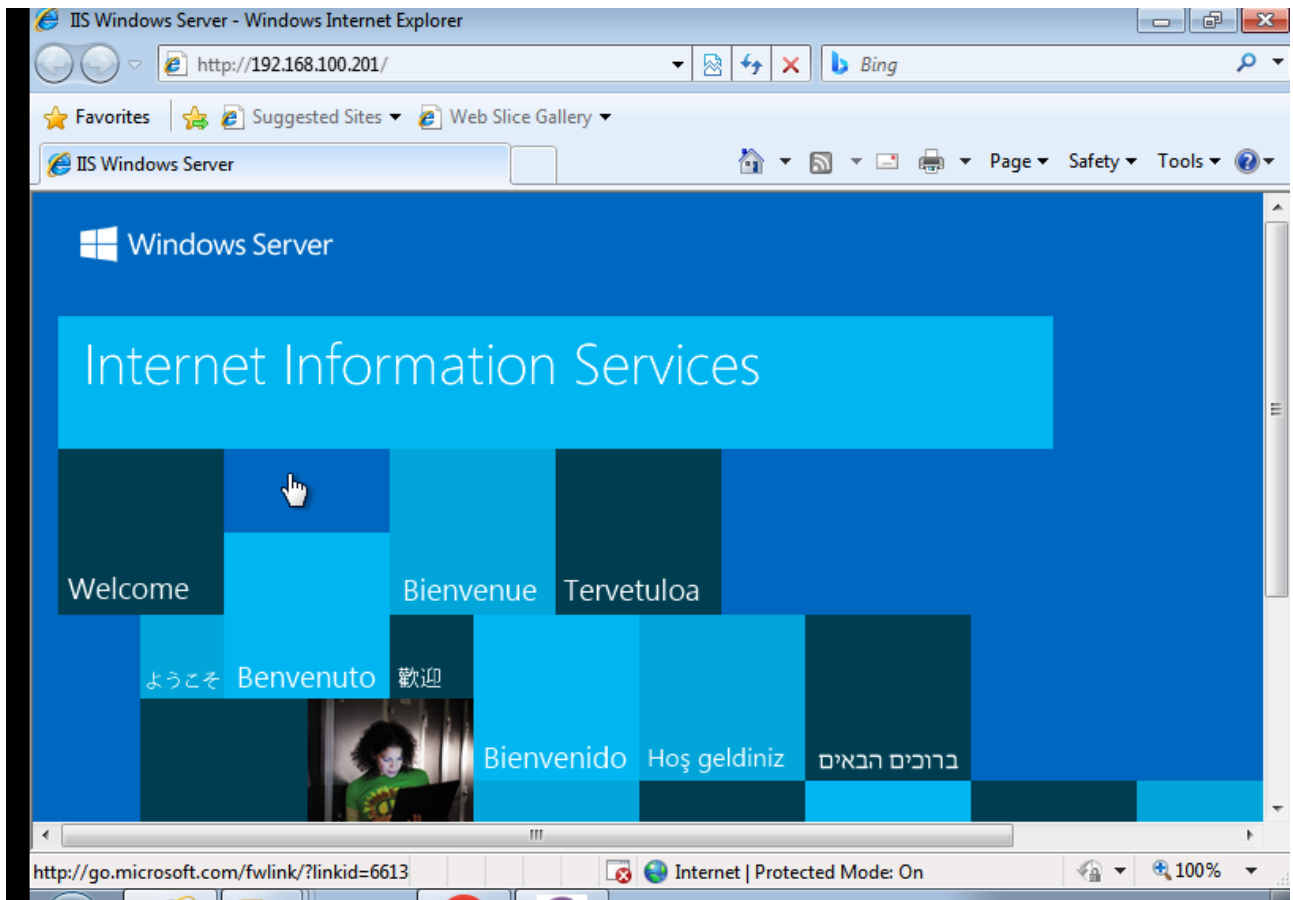
- Trên máy linux sniffer lọc giao thức ftp sử dụng wireshark



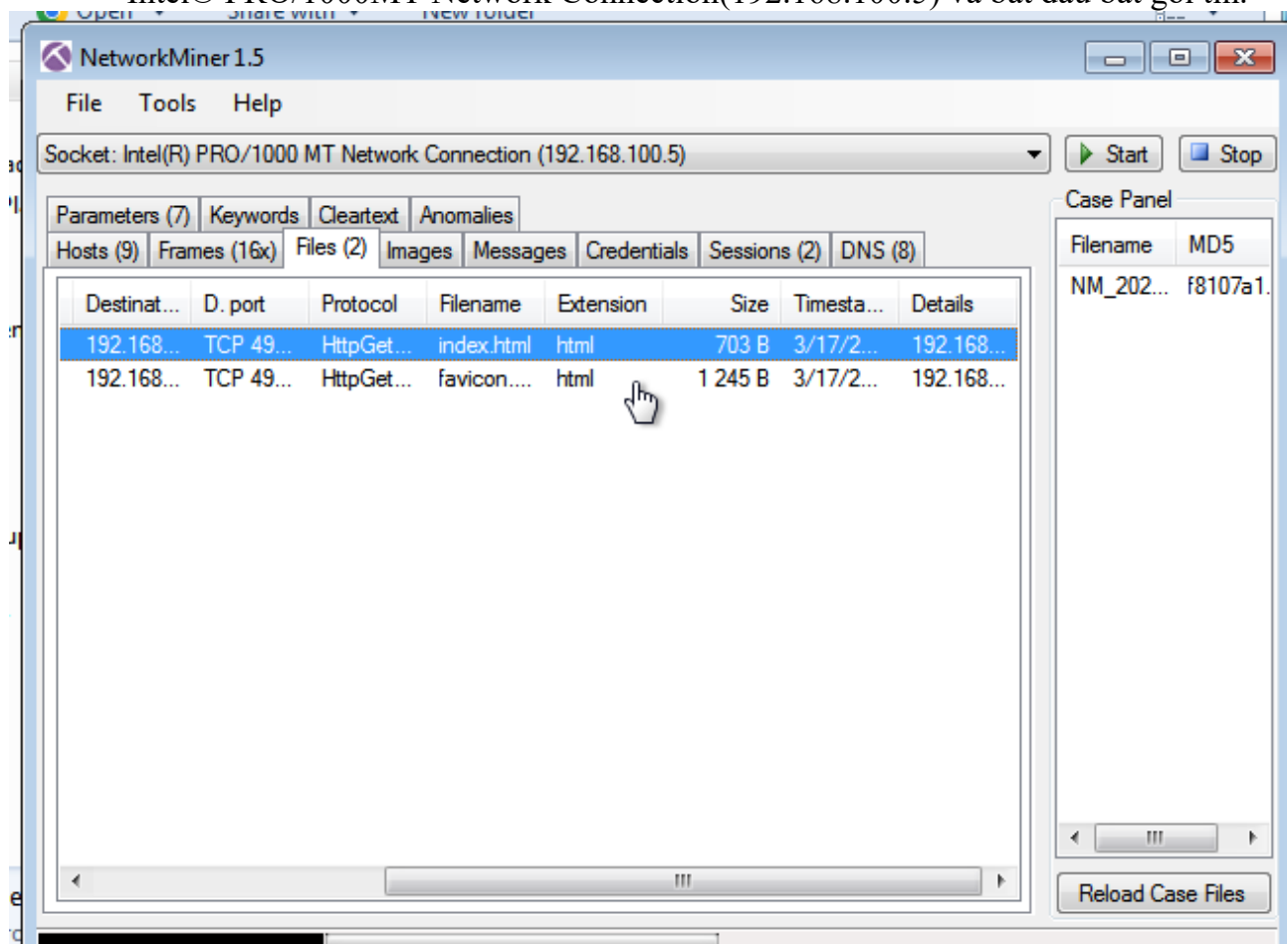
No.	Time	Source	Destination	Protocol	Length	Info
37	6.306783803	10.10.19.202	10.10.19.148	FTP	89	Response: 331 Password required
39	8.659579716	10.10.19.148	10.10.19.202	FTP	81	Request: PASS Minh3153
40	8.663103682	10.10.19.202	10.10.19.148	FTP	91	Response: 530 User cannot log in.
45	17.672448465	10.10.19.148	10.10.19.202	FTP	86	Request: USER administrator
46	17.672828507	10.10.19.202	10.10.19.148	FTP	89	Response: 331 Password required
51	21.220479802	10.10.19.148	10.10.19.202	FTP	81	Request: PASS Minh3153
52	21.221060231	10.10.19.202	10.10.19.148	FTP	87	Response: 230 User logged in.
54	21.221539641	10.10.19.148	10.10.19.202	FTP	72	Request: SYST
55	21.222098369	10.10.19.202	10.10.19.148	FTP	82	Response: 215 Windows_NT
56	21.222496557	10.10.19.148	10.10.19.202	FTP	72	Request: FEAT
57	21.223153196	10.10.19.202	10.10.19.148	FTP	100	Response: 211-Extended features supported:
58	21.223153549	10.10.19.202	10.10.19.148	FTP	84	Response: LANG EN*

#### 4.3. Sử dụng Network Miner bắt gói tin

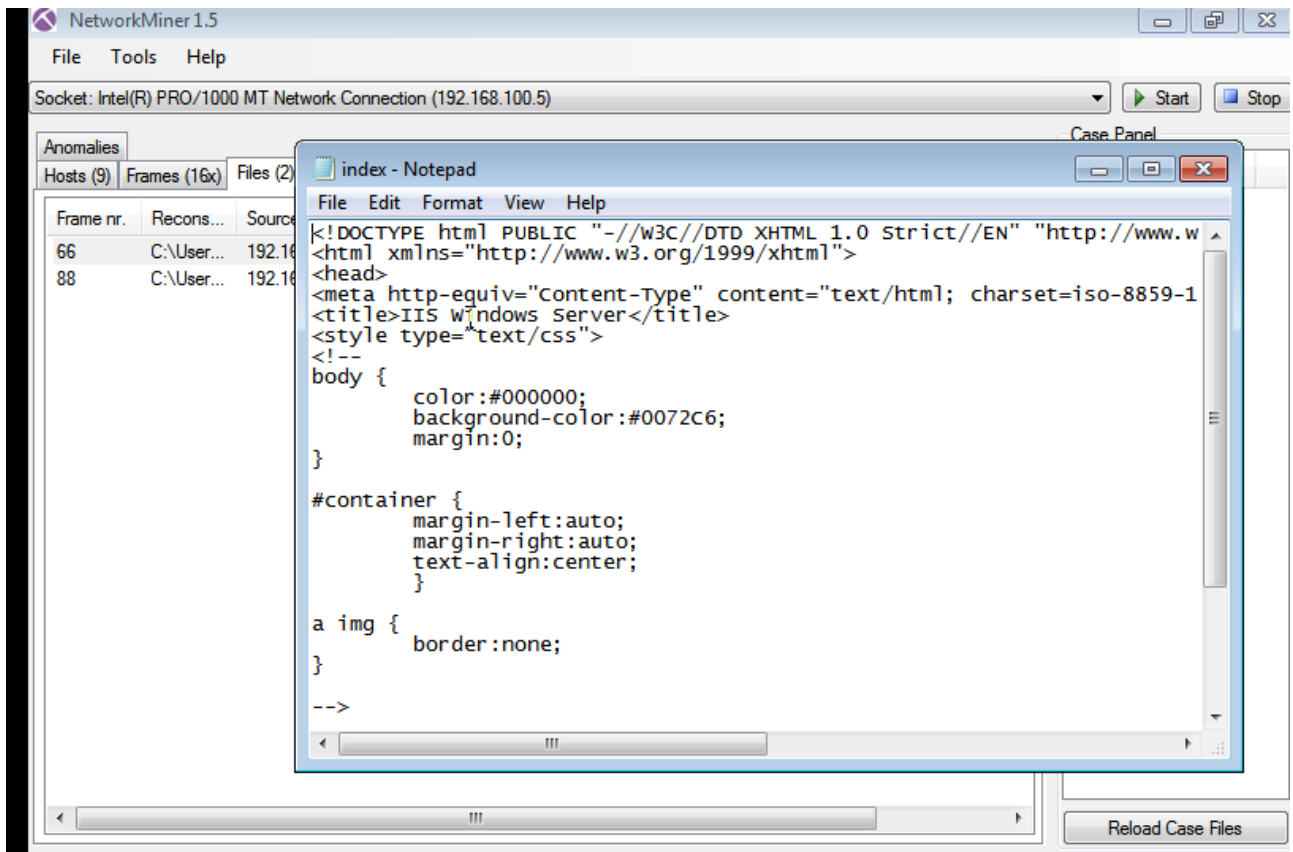
- Trên windows 7 truy cập web với địa chỉ ip 192.168.100.201



- Trên máy Windows 7 Internal Attack khởi động Network Miner và chọn Socket: Intel® PRO/1000MT Network Connection(192.168.100.5) và bắt đầu bắt gói tin.



- Chuột phải và chọn open file để xem dữ liệu gói tin vừa bắt được



## 5. Kết quả

- Bài thực hành chính thức hoàn thành vào ngày 18/03/2024 do em bị vướng mắc cài đặt network miner trên windows 7



