OLD DOMINION UNIVERSITY

CYSE 301: CYBERSECURITY TECHNIQUES AND OPERATIONS

FALL 2017

Module I
Traffic Tracing and Analysis

Topic 3 Use Wireshark to Capture Network Traffic

1. Introduction

In this module, we are going to learn about the basic network structures and the way of simple network defense and countermeasures. As a network administrator, if we want to set up and maintain a simple functionality and security network, we are not only need to master the fundamental knowledge of the network but also need to operate and configure the network facility such as the switch, router and firewall in the field and track the trace of the package through the network to deeply understand how to do cyber defense from the very beginning. Also, as a network administrator, one essential ability is to capture and analyze network traffic. This can be important to identify the cause of bottleneck, determining who is responsible for certain intrusion.

2. Objective

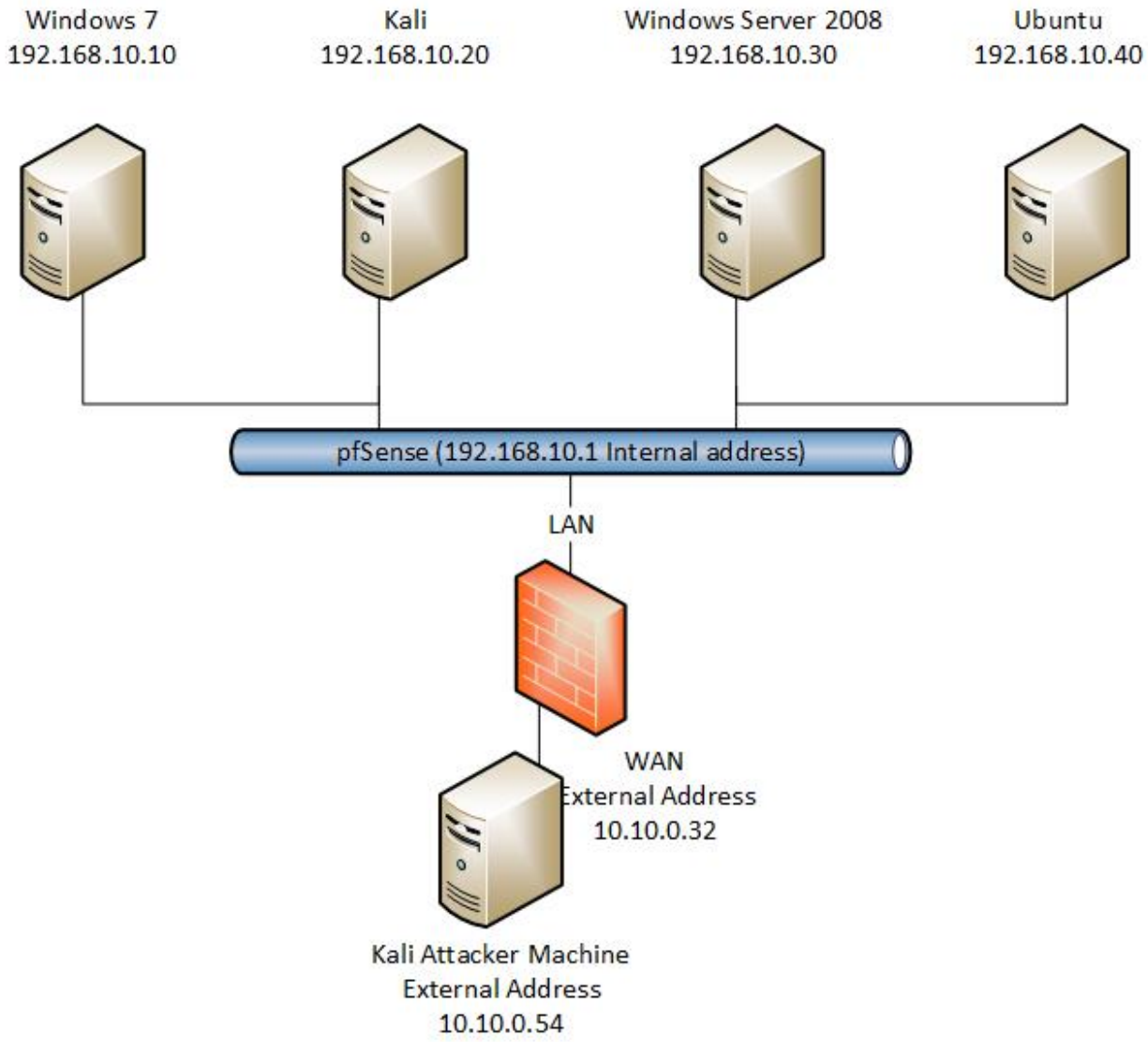
The objective of this topic is to master the most widely used traffic sniffing tool Wireshark to monitor the network traffic. Network traffic analysis is primarily done to get in-depth insight into what type of traffic/network packets or data is flowing through a network. Typically, network traffic analysis is done through a network monitoring or network bandwidth monitoring software/application, in this topic is Wireshark.

WARNING:

Listening, sniffing, eavesdropping on networks to which you do not have legal access is unethical and may even constitute a crime in your area.

NETWORK TOPOLOGY

In this lab, you can log into either Windows 7 VM, Kali VM, or Ubuntu VM to complete the assignment.



3. Getting Started with Wireshark

Before you start Wireshark, you will want to clear the cache (or "recent history") on your web-browser (Internet Explorer, Firefox, Safari) and close any open programs.

3.1 Open Wireshark

Wireshark can run on multiple systems, Windows, Linux and MacOS. In Windows and Mac OS system, you can just double click the Wireshark shortcut (Figure 1) on your desktop and start the program.

Figure 1 Wireshark shortcut

In Linux system like Kali, Ubuntu and Debian, open Wireshark as a standard user may not be able to work properly due to the permission issue (Figure 2).

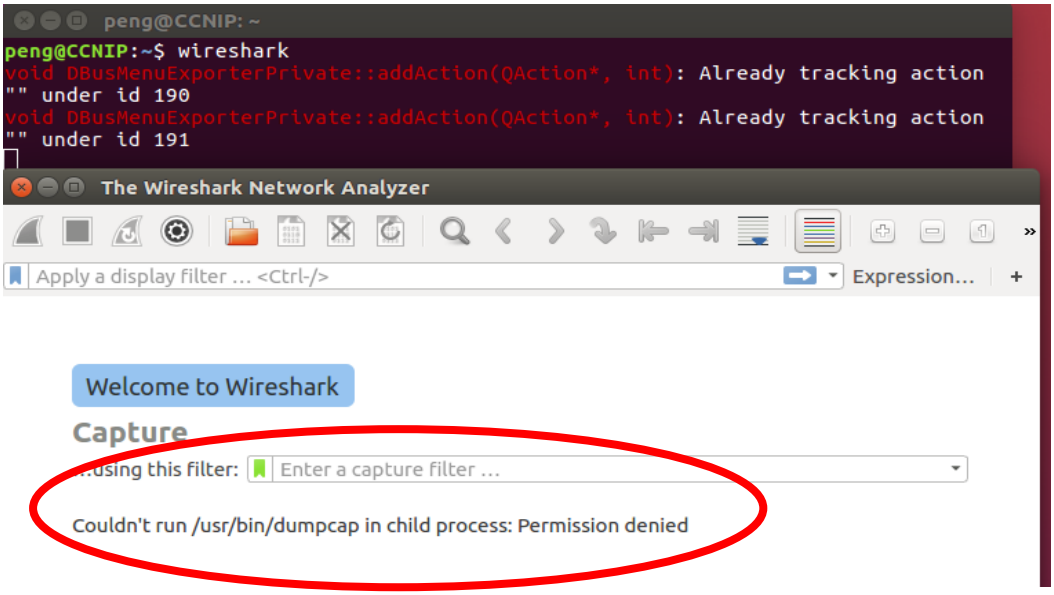


Figure 2

In this case, we need to use **sudo** command to give Wireshark the root privilege (Figure 3). If the list of devices is shown on the welcome page (red rectangular), that means your Wireshark is ready to use.

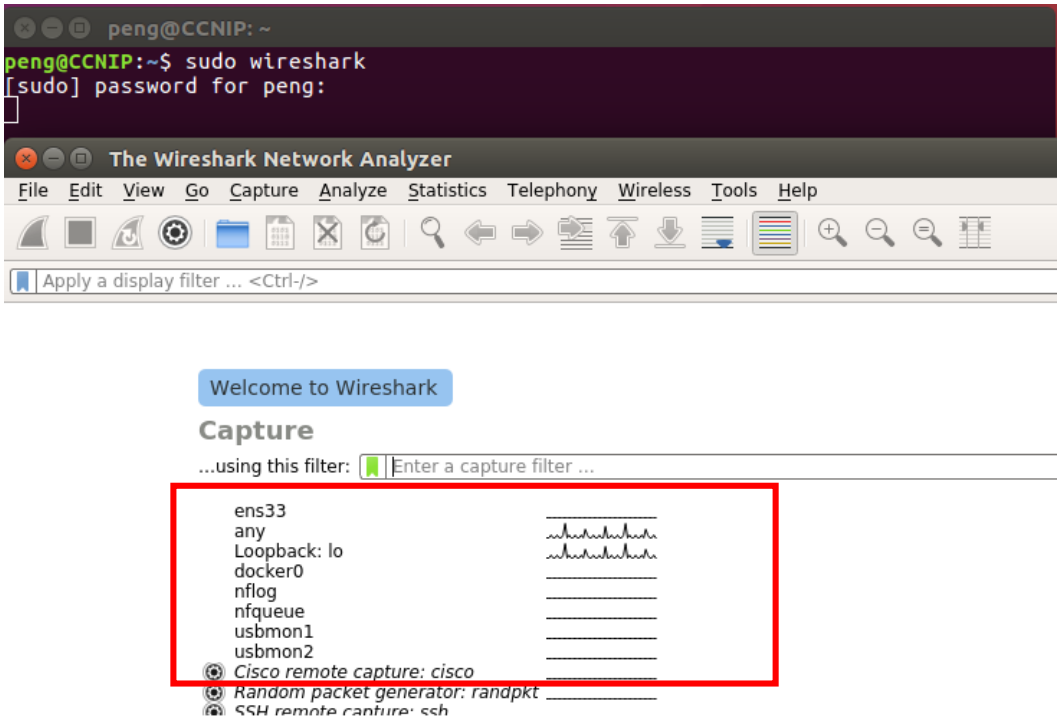


Figure 3

3.2 Select a Network Interface

When you first start Wireshark, you will be brought to the default startup screen (different for Windows version and Linux version). To start tracing the traffic you will need to choose a network device to use. There are several ways to do this: for Linux system (Figure 4), you can simply select the "List the Available Capture Devices" button and for Windows system, click “Capture options” (Figure 4)to see the list of interfaces.

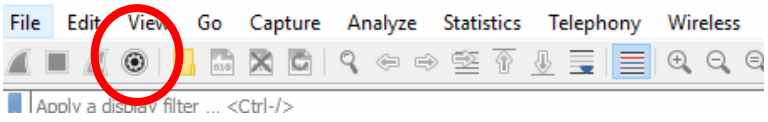


Figure 4

- Notice: In this lab tutorial, we run the Wireshark under Windows system for example, so the interface may have minor difference with Wireshark under Linux, but the content and option are the same.

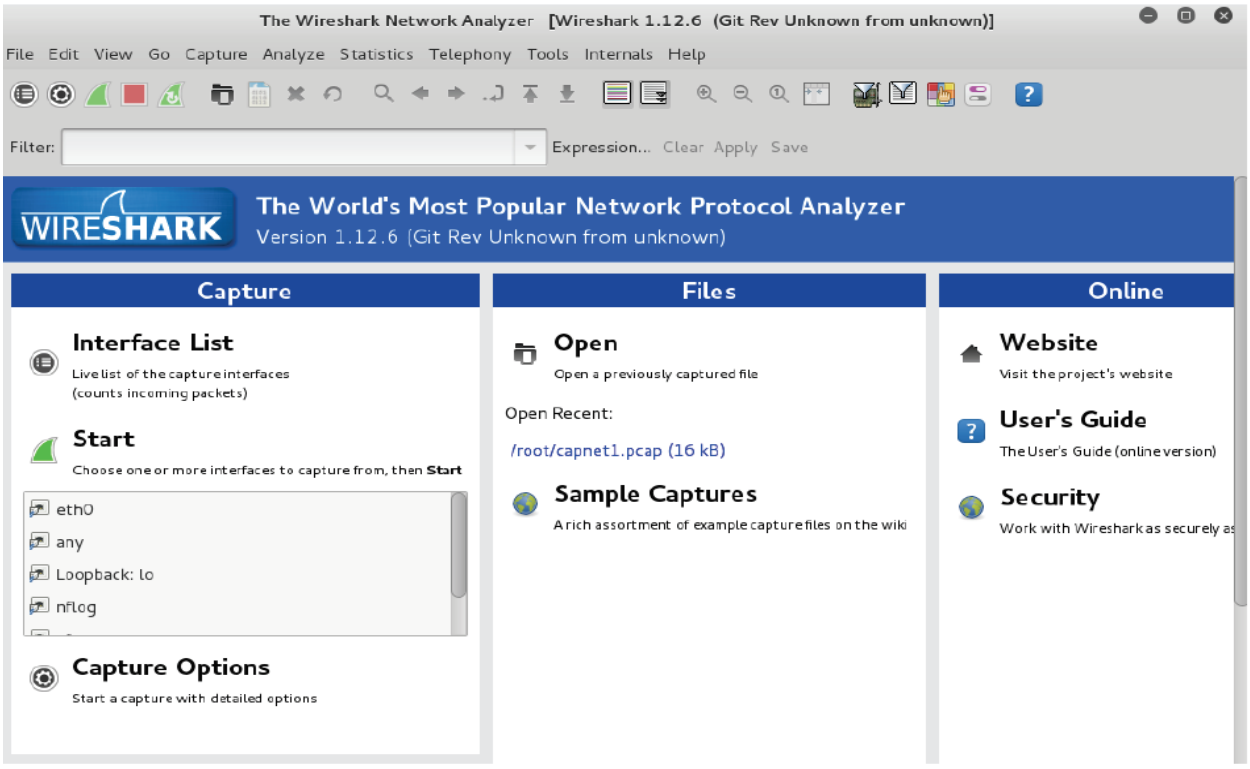


Figure 5 Startup screen of Linux system

In the pop-up box that appears you will need to select one of the devices available on your machine. Most machines will only have one working network interface available. If you have more than one device available, you need to decide which one you want to choose, the Ethernet interface

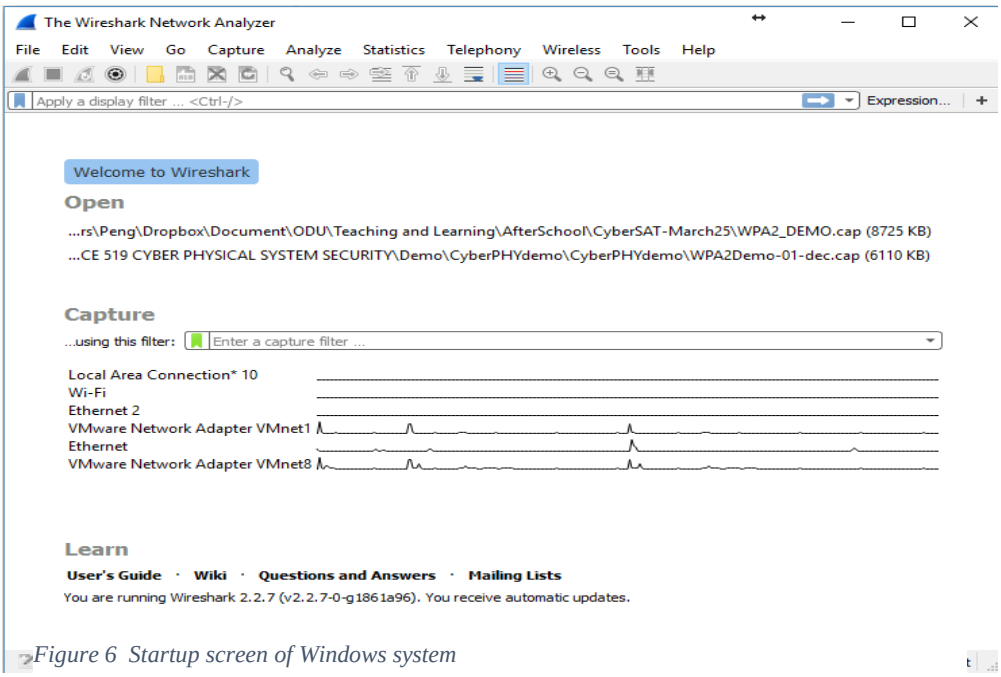


Figure 6 Startup screen of Windows system

(cable) or the wireless interface. And the representation of device name varies from Windows system and Linux system.

As you can see in the Figure 6 and Figure 5, there are multiple options in both Windows and Linux system. In Windows, there are Wi-Fi, Ethernet 2, Ethernet… and in the corresponding Linux system, there are shown as eth0, wlan0…

Generally, **eth0**, **lo**, **wlan0, Ethernet, Wi-Fi** are the names of the active network interfaces on the system.

- **eth0** (Linux) or **Ethernet** (Windows) are the first ethernet interface. (Additional ethernet interfaces would be named eth1(Ethernet 2), eth2(Ethernet 3), etc.) This type of interface is usually a NIC connected to the network by a category 5 cable.
- **lo** is the loopback interface. This is a special network interface that the system uses to communicate with itself.
- **wlan0** (Linux) or **Wi-Fi** (Windows) are the name of the first wireless network interface on the system. Additional wireless interfaces would be named wlan1 (Wi-Fi 2), wlan2 (Wi-Fi 3), etc.

Once you have decided what device to use, click "Start" to capture the traffic (Figure 7) on this interface. (In this case, we choose interface "Ethernet")/

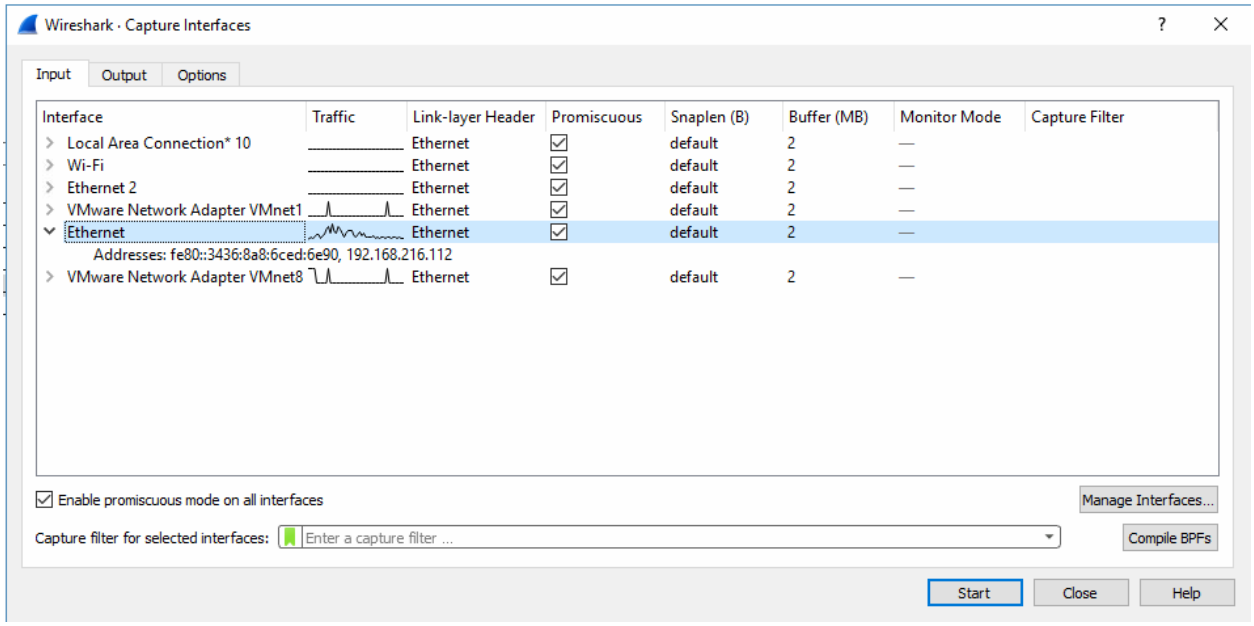


Figure 7 Choose network interfaces

3.3 Capturing Packets

If your Ethernet card is working you should very quickly start to see lines appearing in your packet capture window (Figure 8). Each line represents a packet (a unit of data) that was sent over the network that Wireshark detected. Even though you may not be actively requesting any information (such as asking your browser to get you a webpage) your computer and the other devices on your network are constantly sending messages back and forth to each other to keep each other updated about their status, and programs on your computer may also be sending information over the network on their own (software update requests, would be one example.)

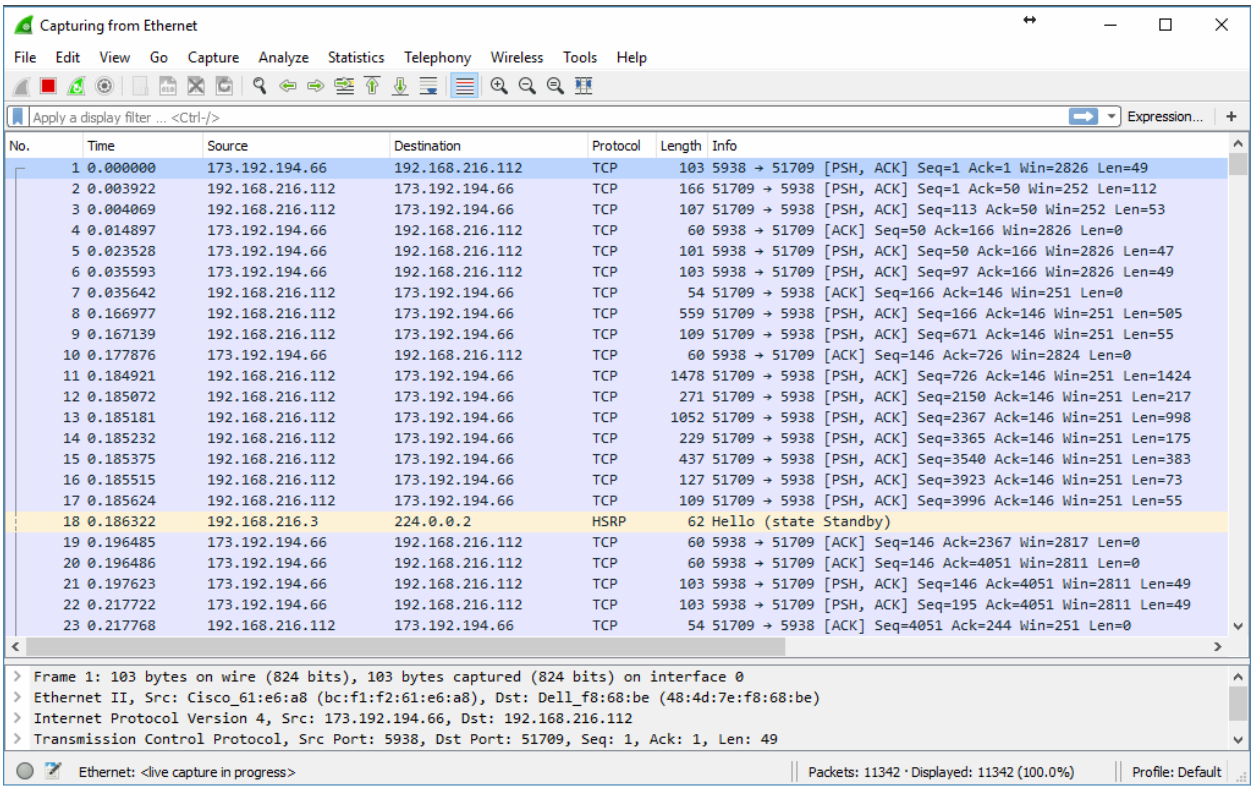


Figure 8

The different columns in the display window detail the number and time of packets that were captured, the source and destination for the packets, as well as protocol type and general information about the packet. You can double click on a packet to get more information (Figure 9).

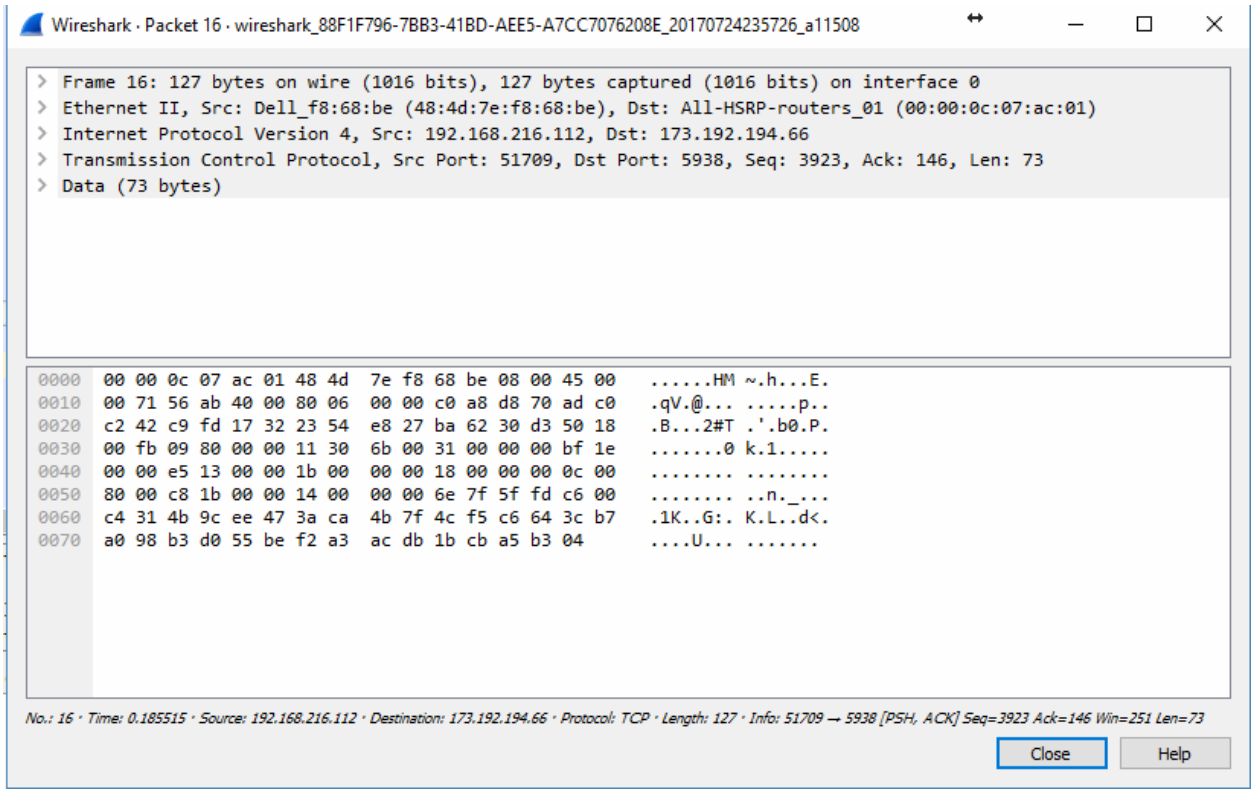


Figure 9 Packet detail

3.4 Understand a packet

To understand a caputered data packet, we first need to understand what are the MAC address, IP address and Port number.

- **MAC address**

Every network device (network card) has a globally unique MAC (Machine Access Code) Address. Mac Addresses (in theory) can never me changed, so even if a machine is moved to a new network, its Mac Address will remain the same. Mac Addresses are used for communication between network devices that reside on the same physical network segment. The standard (IEEE 802) format for printing MAC-48 addresses in human-friendly form is six groups of two hexadecimal digits, separated by hyphens (-) or colons (:), in transmission order. Examples:

01-23-45-67-89-ab

01:23:45:67:89:ab

To look in your packet capture window for a packet that is using the TCP protocol, we double click on the packet line to obtain more details about that specific packet (Figure 9).

In the window above you can see that the frame that I captured and opened to examine was an Ethernet II packet, which contained within it two other packets. The packet selected above is a nesting of 3 different protocol packets, like the picture below (Figure 10).

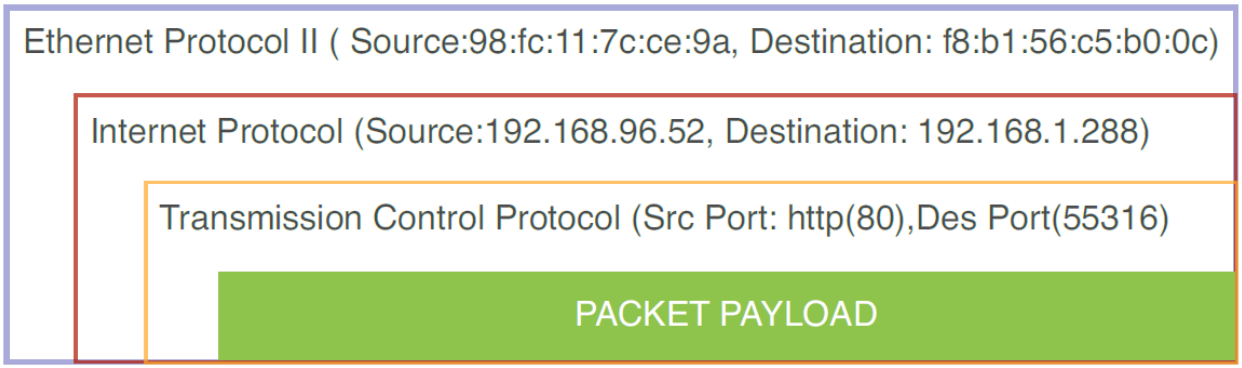


Figure 10 Packet stack

- **IP address**

The Ethernet Protocol uses MAC addresses to pass messages between machines that are on the same network.

Since not all machines of course are on the same physical network. Two machines that are using the Internet to communicate may be on opposite sides of the world and therefore will need to use **IP (Internet Protocol)** Addresses to communicate. An IP address is a unique number that can be assigned to a network card. The Internet (using routing tables) keeps track of where particular sets of IP addresses are located and how to route messages to get to those IP addresses.

You can think of it like this, if you and I are not in the same room and I want to get a message to you, I could simply send you a letter. In order for the letter to reach you, I need to have your address (street, state, country, zip). With your address, I can put my letter into any mailbox anywhere in the world and it will eventually get to you. Just as you can move and change your physical address, a computer can also move and change its IP address.

There are two general formats for IP addresses: IPv4 and IPv6. Most computers in the world currently use IPv4 addresses. IPv6 is the successor to the Internet Protocol version 4 (IPv4).

In contrast to IPv4, which defined an IP address as a 32-bit number, IPv6 addresses have a size of 128 bits, vastly expanding the addressing capability of the Internet Protocol. IPv4 addresses are most often written in dot-decimal notation, which consists of the four octets of the address expressed separately in decimal and separated by periods.

Example:

192.168.1.1

IPv6 address is represented as eight groups of four hexadecimal digits, each group representing 16 bits (two octets). The groups are separated by colons (:) Leading 0's are sometime omitted. An example of both formats is shown below:

2001:0db8:85a3:0000:0000:8a2e:0370:7334 or 2001:db8:85a3:0:0:8a2e:370:7334.

- **Port Number**

A single computer may at any given time be running dozens of different programs (applications). Consequently, when a network packet arrives at a machine, we would like to know what program should receive the information in the packet (if it's a webpage it needs to go to the web-browser, if it's a anti-virus update, it needs to go to the antivirus program). We can make sure that data packets reach the proper programs by using Ports. Many protocols such as Transmission Control Protocol (TCP) and Universal Datagram Protocol (UDP) allow packets to be addressed to unique numbers called ports. These ports, in turn are linked (by convention, and by the operating system) to certain programs. Examples:

Port 80: Used for web-servers

Port 21: Used by FTP Servers

Port 3724: Used by "World of Warcraft" game

3.5 Stop and save your traffic file

By now your capture (which we left running) should have acquired a lot of data. Let's stop our capture and then save it so that we can use it later. We do this by hitting the "stop capture" button in red circle and then hitting the “Save this capture file” in blue circle. And then you can save the file in the name and folder you want.

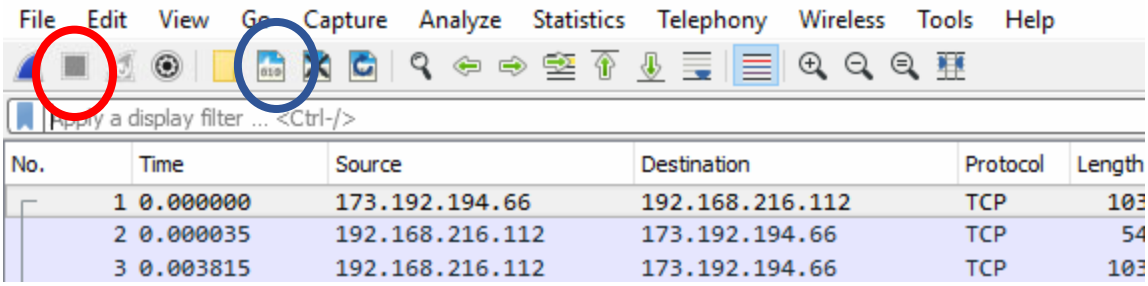


Figure 11 Stop and save the traffic

3.6 Analyzing the Saved Capture

Wireshark has many different tools to help users filter and analyze the contents of a capture. Open a previous capture file (you may have one already open) and then in the menu bar click on **Statistics** and then **Protocol Hierarchy**. You should get a window that looks something like this (Figure 12 and Figure 13)

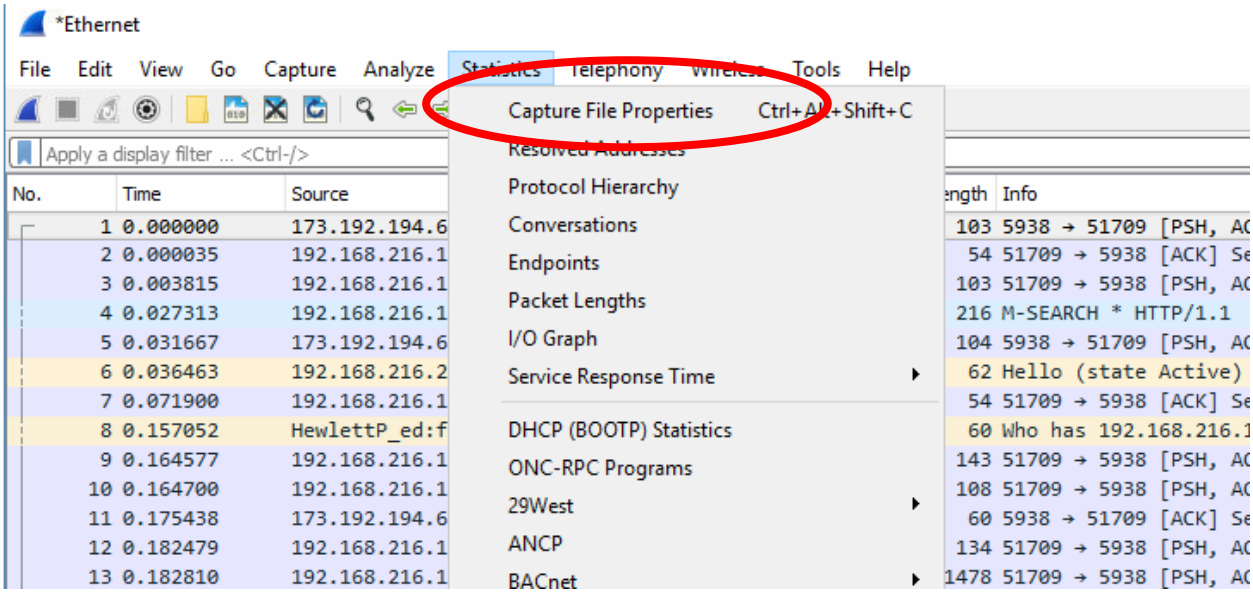
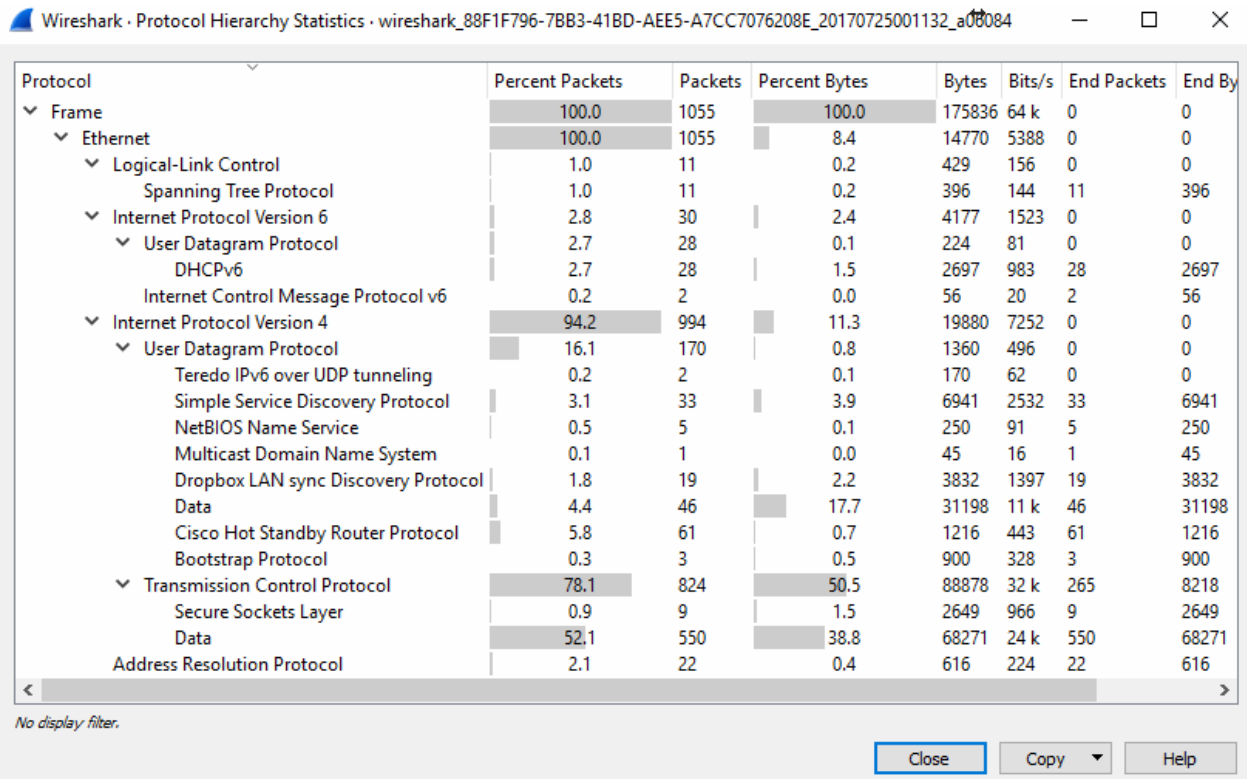


Figure 12



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End By
Frame	100.0	1055	100.0	175836	64 k	0	0
Ethernet	100.0	1055	8.4	14770	5388	0	0
Logical-Link Control	1.0	11	0.2	429	156	0	0
Spanning Tree Protocol	1.0	11	0.2	396	144	11	396
Internet Protocol Version 6	2.8	30	2.4	4177	1523	0	0
User Datagram Protocol	2.7	28	0.1	224	81	0	0
DHCPv6	2.7	28	1.5	2697	983	28	2697
Internet Control Message Protocol v6	0.2	2	0.0	56	20	2	56
Internet Protocol Version 4	94.2	994	11.3	19880	7252	0	0
User Datagram Protocol	16.1	170	0.8	1360	496	0	0
Teredo IPv6 over UDP tunneling	0.2	2	0.1	170	62	0	0
Simple Service Discovery Protocol	3.1	33	3.9	6941	2532	33	6941
NetBIOS Name Service	0.5	5	0.1	250	91	5	250
Multicast Domain Name System	0.1	1	0.0	45	16	1	45
Dropbox LAN sync Discovery Protocol	1.8	19	2.2	3832	1397	19	3832
Data	4.4	46	17.7	31198	11 k	46	31198
Cisco Hot Standby Router Protocol	5.8	61	0.7	1216	443	61	1216
Bootstrap Protocol	0.3	3	0.5	900	328	3	900
Transmission Control Protocol	78.1	824	50.5	88878	32 k	265	8218
Secure Sockets Layer	0.9	9	1.5	2649	966	9	2649
Data	52.1	550	38.8	68271	24 k	550	68271
Address Resolution Protocol	2.1	22	0.4	616	224	22	616

Figure 13

In this window, you can see that the types of all protocol are listed which were used during the capture file.