ABOUT WIRESHARK

Wireshark is a **network packet analyzer**. A network packet analyzer will try to capture network packets and try to display that packet data as detailed as possible. We could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course).

In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Wireshark, all that has changed. Wireshark is perhaps one of the best open source packet analyzers available today.

WIRESHARK EXPERIMENTS LIST

1. To demonstrate how to sniff for router traffic by using Wireshark

Sniffing refers to the process of capturing and analyzing network traffic. The packet contents on a network are analyzed. The tools that attackers use for sniffing are called sniffers or more correctly, protocol analyzers. While protocol analyzers are really network troubleshooting tools, hackers also use them for malicious purposes.

A **sniffer** is an application or device that can read, monitor, and capture network data exchanges and read network packets. If the packets are not encrypted, a sniffer provides a full view of the data inside the packet. Even encapsulated packets can be broken open and read unless they are encrypted and the attacker does not have access to the key. Sniffers monitor, capture, and obtain network information such as passwords and valuable customer information.

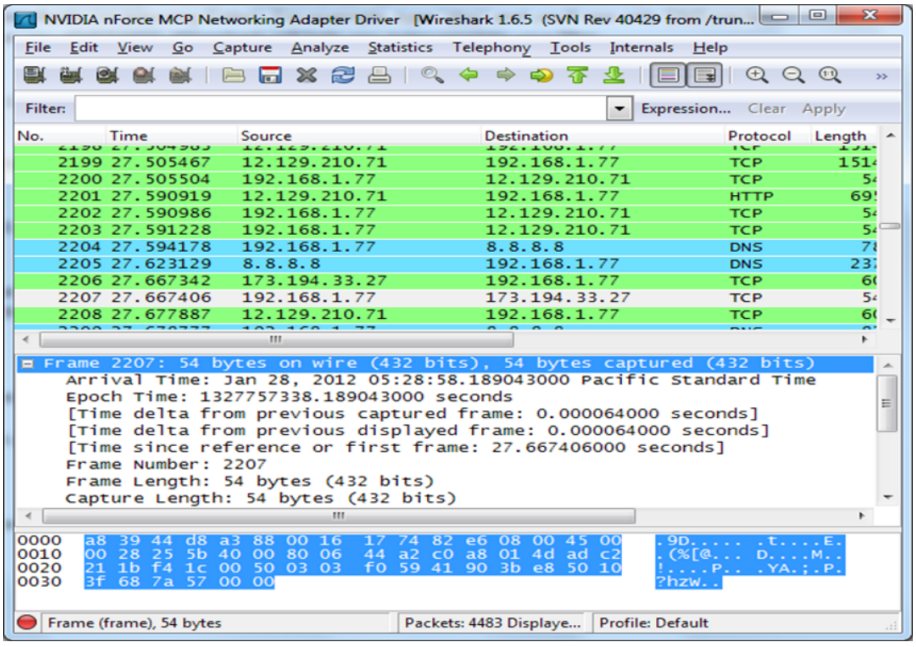
A **packet sniffer**, sometimes referred to as a network monitor or network analyzer, can be used by a network or system administrator to monitor and troubleshoot network traffic. Using the information captured by the packet sniffer an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help maintain efficient

network data transmission.

In its simple form a packet sniffer simply captures all of the packets of data that pass through a given network interface. By placing a packet sniffer on a network in promiscuous mode, a malicious intruder can capture and analyze all of the network traffic.

Wireshark is used to capture and examine encrypted and unencrypted wireless traffic. Use the Wireshark program that is preinstalled in Backtrack, or we can download the Windows version from www.wireshark.org.

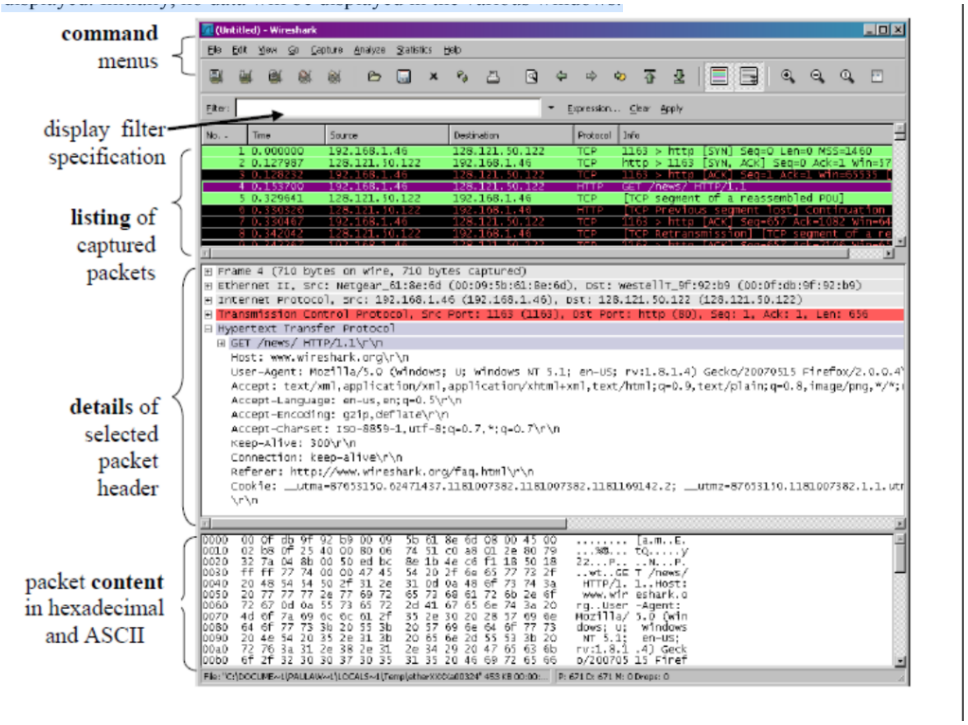
1. After loading Wireshark, we will see several options across the top of the program. Select Capture -> Options to configure the program. Make sure to choose the correct interface (NIC) adapter and set the program to update packets in real time and for automatic scrolling.
2. Choose the Start Capture option.



3. After a few packets have been captured, stop Wireshark. We will see information displayed in three different views. The top window shows all packets that were captured. Clicking one of these will display that frame‘s contents in the middle frame; we may also note that the bottom frame displays the actual hex dump. While reading hex is not mandatory, notice the first 16 bytes of the frame. The first 8 bytes are the destination MAC and the second 8 bytes are the source MAC.

4. Now use Wireshark to capture and analyze some wireless traffic with and without encryption. Note that the MAC addresses will be visible in both.

   When we run the Wireshark program, the Wireshark graphical user interface shown below will be displayed. Initially, no data will be displayed in the various windows.



The Wireshark interface has five major components:

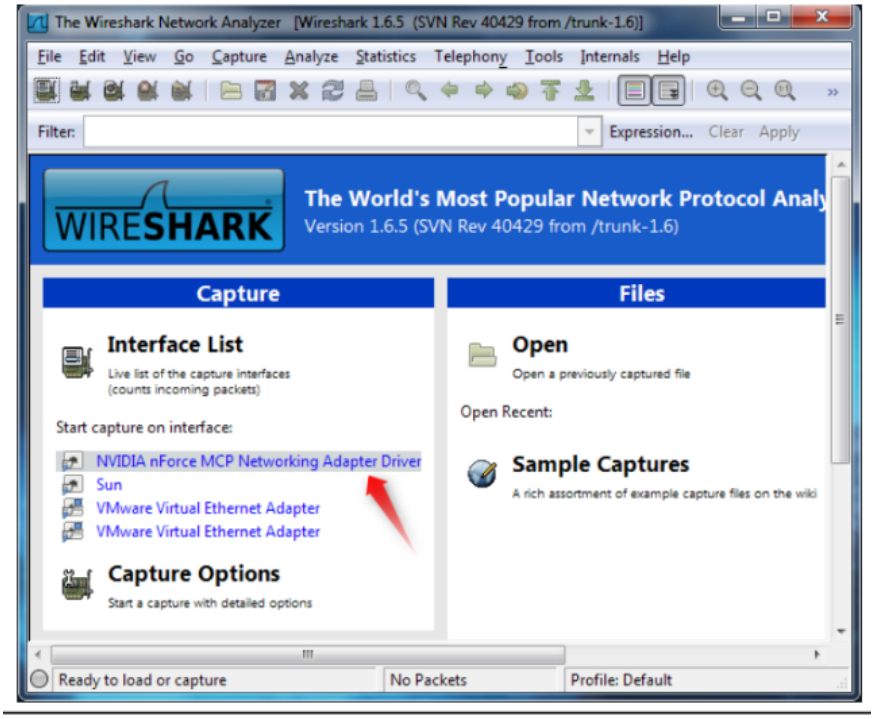
- The *command menus* are standard pull-down menus located at the top of the window. Of interest to us now is the File and Capture menus. The File menu allows us to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows us to begin packet capture.
- The *packet-listing window* displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these

categories by clicking on a column name. The protocol type field lists the highest level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

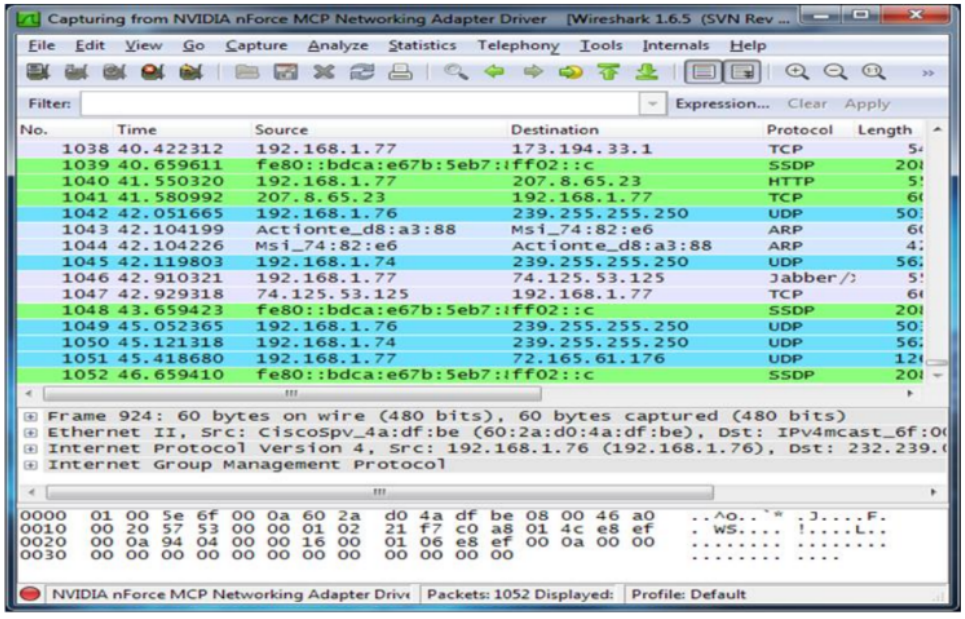
- The *packet-header details* window provides details about the packet selected (highlighted) in the packet listing window. (To select a packet in the packet listing window, place the cursor over the packet‘s one-line summary in the packet listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the plus-or-minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest level protocol that sent or received this packet are also provided.
- The *packet-contents* window displays the entire contents of the captured frame, in both ASCII and hexadecimal format.
- Towards the top of the Wireshark graphical user interface, is the packet *display filter field*, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we‘ll use the packet-display filter field to have Wireshark

Capturing Packets

After downloading and installing Wireshark, we can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if we want to capture traffic on the wireless network, click our wireless interface. We can configure advanced features by clicking Capture Options.

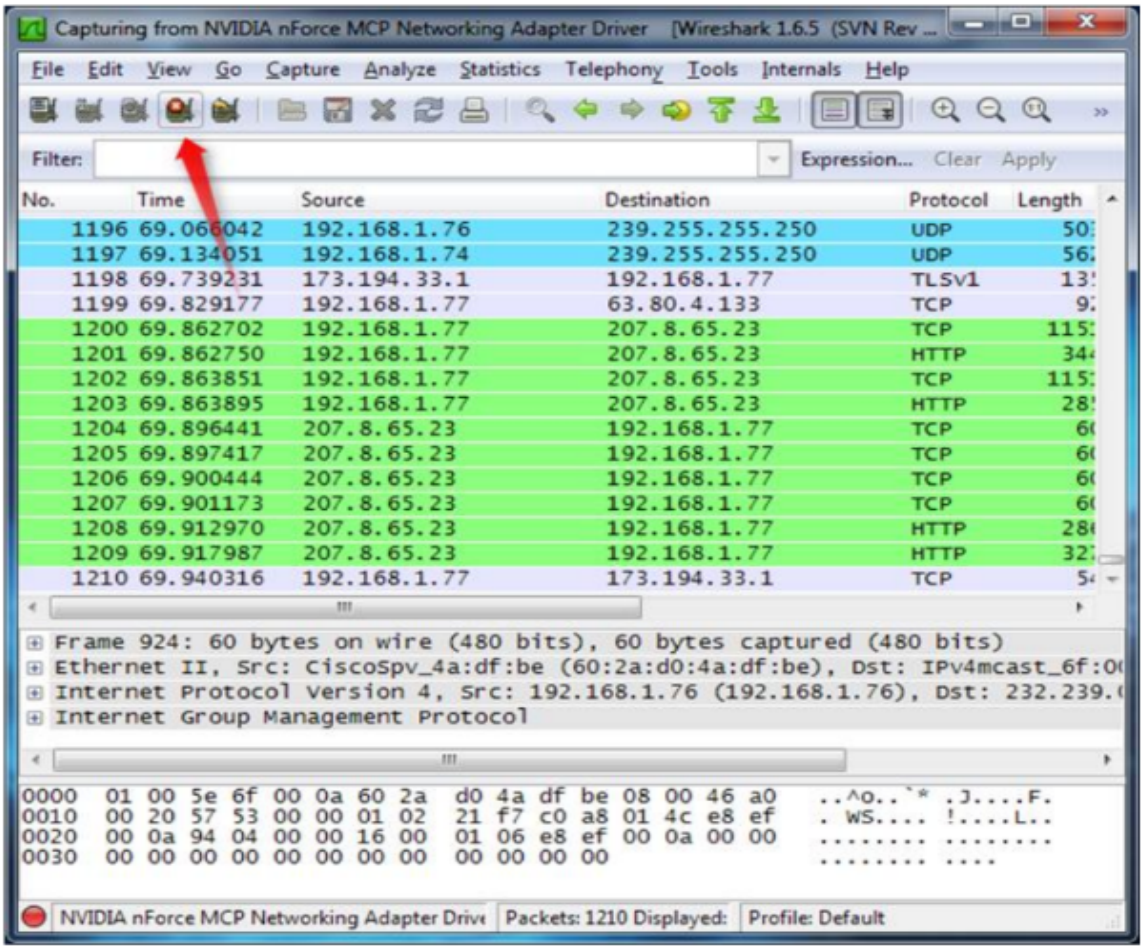


As soon as we click the interface‘s name, we will see the packets start to appear in real time. Wireshark captures each packet sent to or from our system. If we are capturing on a wireless interface and have promiscuous mode enabled in our capture options, we will also see the other packets on the network.



Click the stop capture button near the top left corner of the window when we want to stop

capturing traffic.

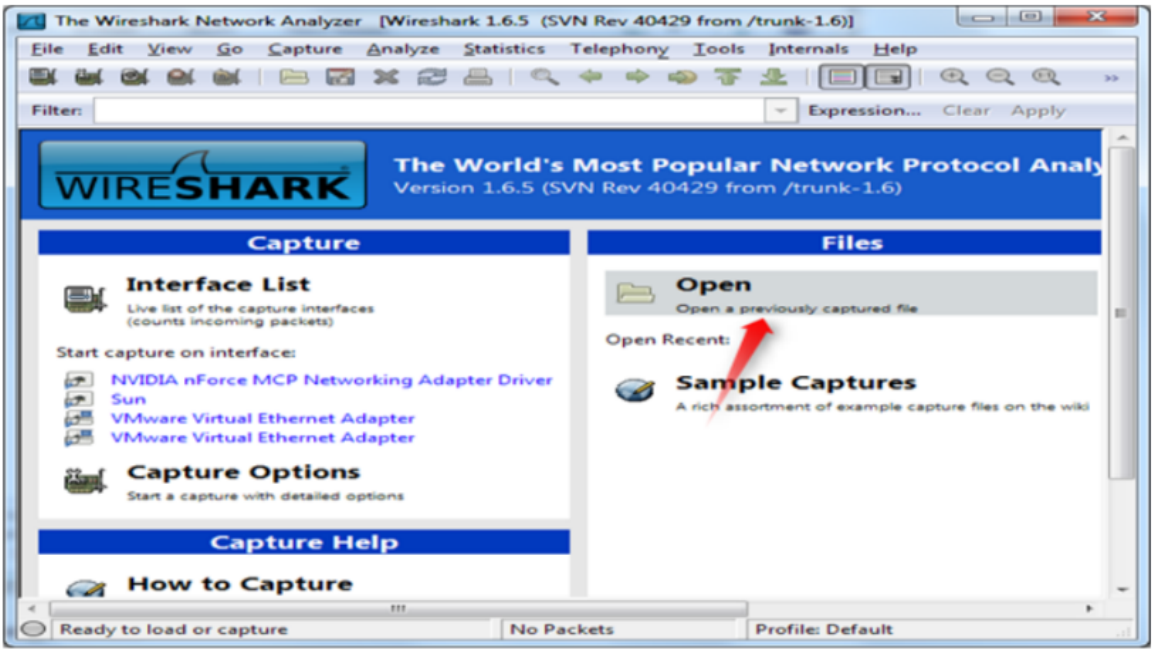


Color Coding

Observe the packets highlighted in green, blue, and black. Wireshark uses colors to help us to identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems.

Sample Captures

If there‘s nothing interesting on our own network to inspect, Wireshark‘s wiki has we covered. The wiki contains a page of sample capture files that we can load and inspect. Opening a capture file is easy; just click Open on the main screen and browse for a file. We can also save our own captures in Wireshark and open them later.



Filtering Packets

If we are trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so we can narrow down the traffic. Still, we will likely have a large amount of packets to sift through. That‘s where Wireshark‘s filters come in.

The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type ―dns‖ and we will see only DNS packets.

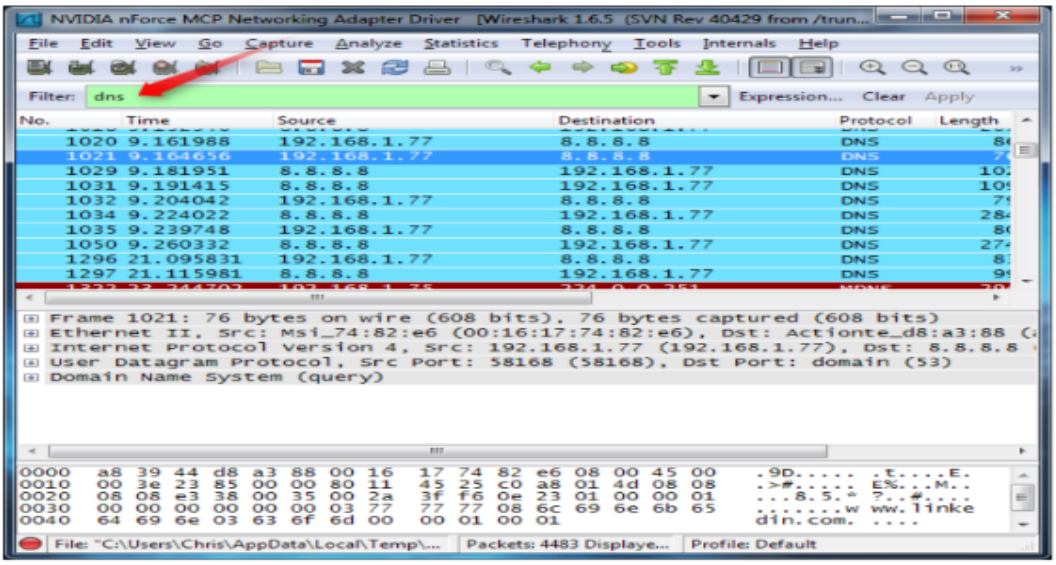
When we start typing, Wireshark will help us autocomplete our filter.

Example: ip.addr == 192.168.1.77

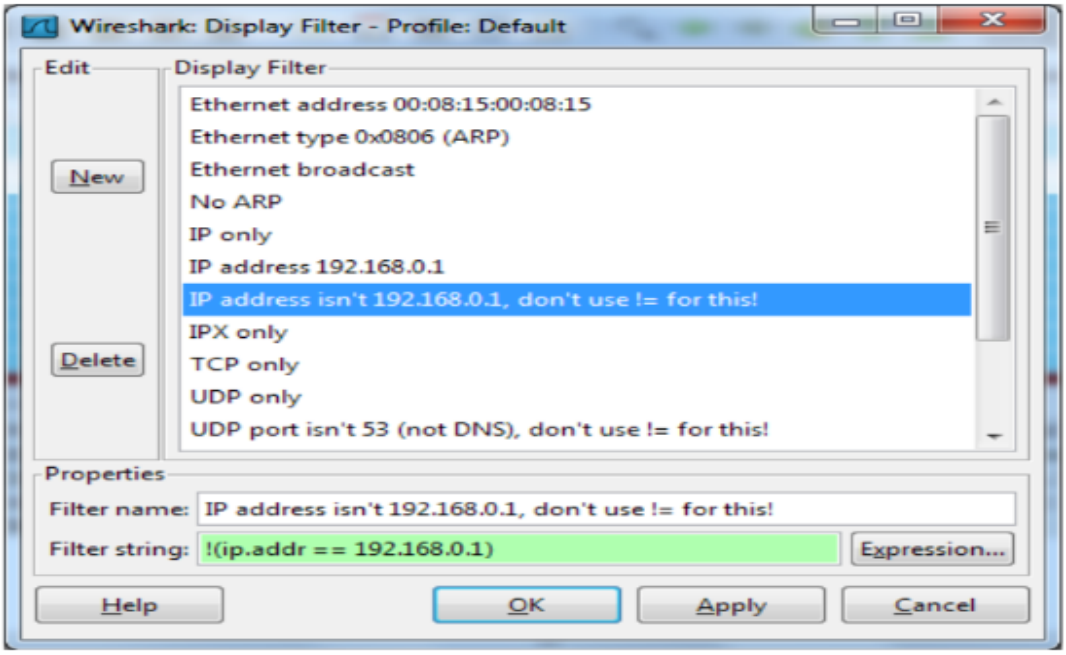
ip.src == 192.168.1.77

ip.dst == 192.168.1.77

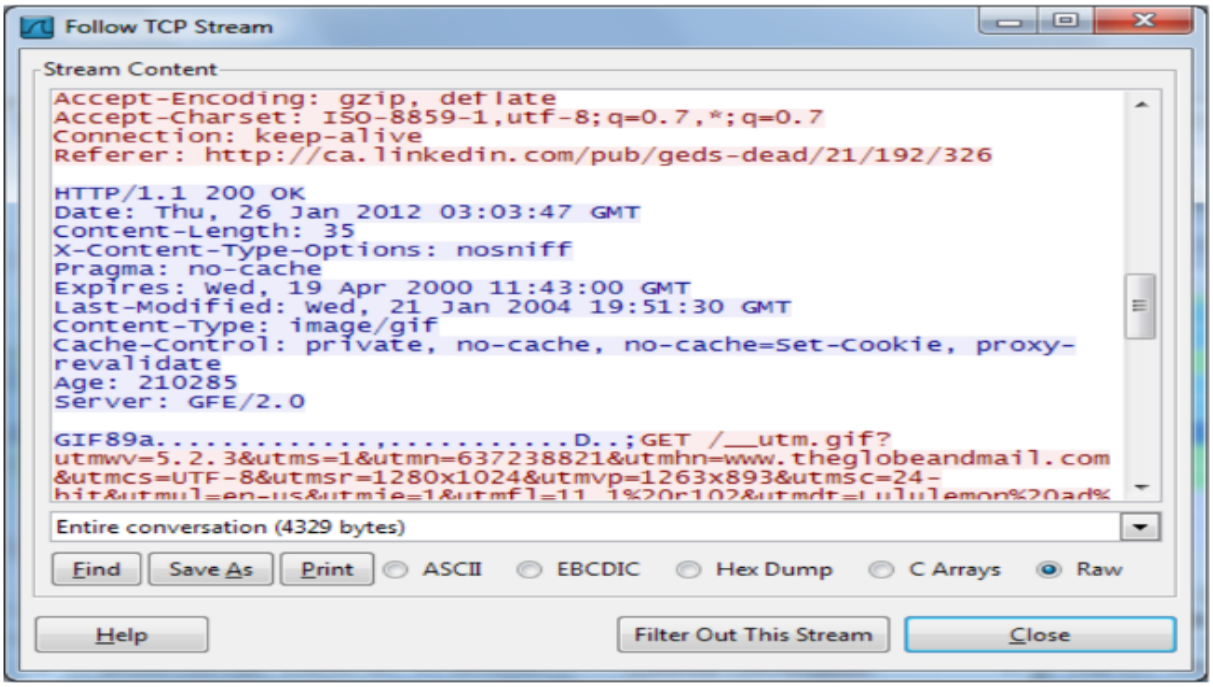
TCP.Port == 80



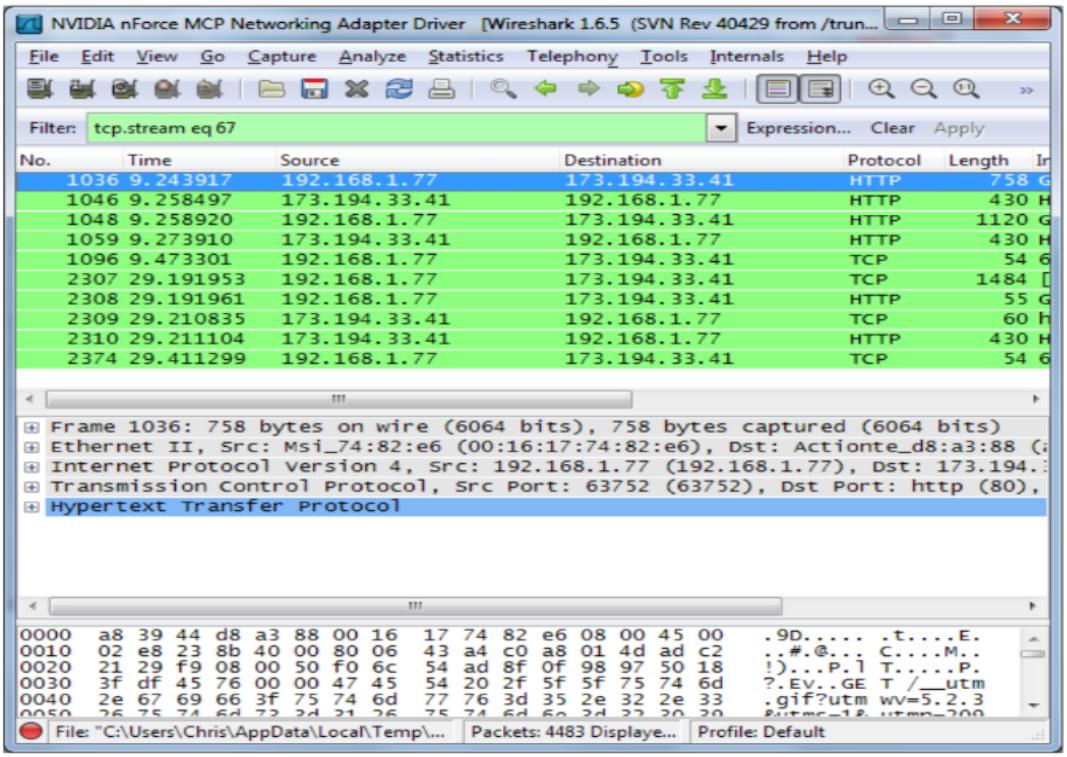
You can also click the Analyze menu and select Display Filters to create a new filter.



You'll see the full conversation between the client and the server.



Close the window and we will find a filter has been applied automatically — Wireshark is showing us the packets that make up the conversation.

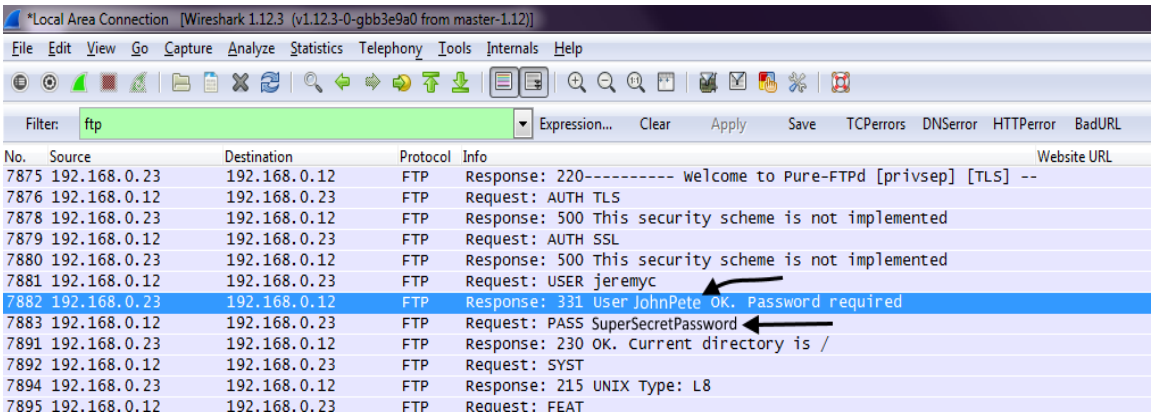


2. To capture the FTP password using Wireshark.

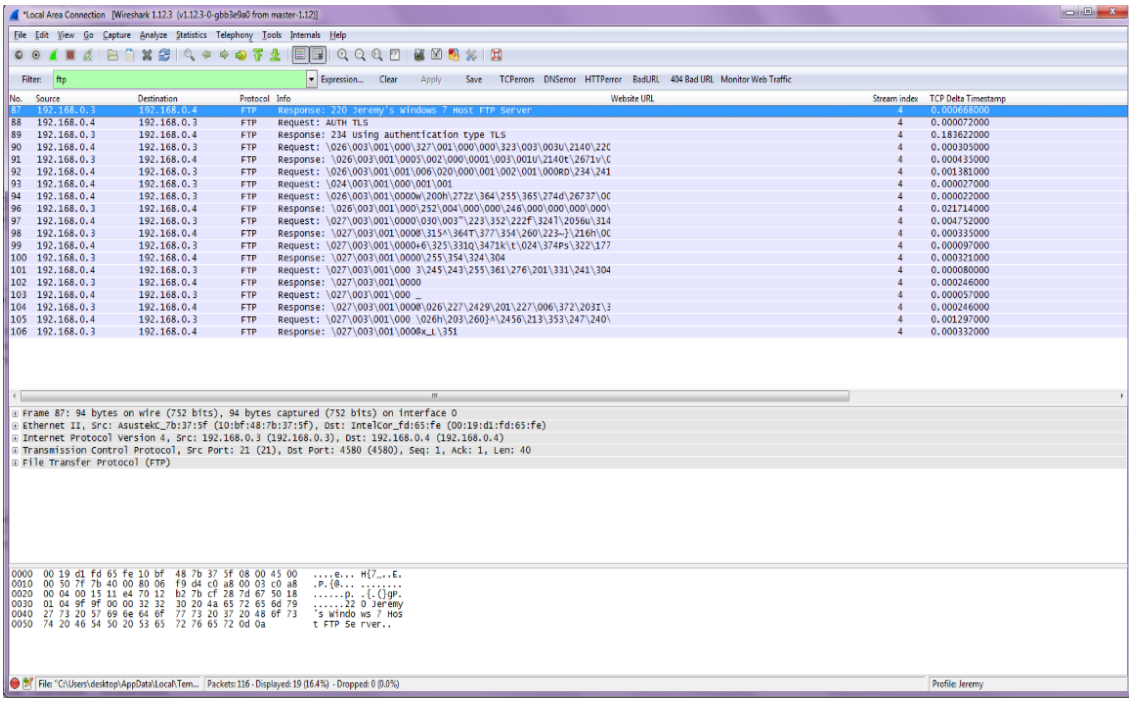
There is a FTP server installed on the Kali Linux VM. You need to use a terminal to log into the server and use Wireshark to capture the password. The username for the FTP server is csc5991-student, and the password is [WSU-csc5991.] without the brackets. You will use the username and password to login the FTP server while Wireshark is running. Note that the FTP server is installed on the localhost, make sure you select the right interface for the capturing.

Alternatively ,

1. Start a capture in Wireshark
2. Connect to an FTP Server. Include your username and password in connection.
3. Stop the capture in Wireshark.
4. In the Wireshark filter, enter FTP.
5. In the list of packets, the unencrypted username and password should be displayed.



On the other hand, if the connection between the client and FTP server is encrypted with a SSL/TLS certficate, Wireshark will not show the username and password.



3. To demonstrate username and password sniffing using WireShark

Upon entering the area of “filters block” enter “frame contains <--*your website that has username and password* → and some traffic can be seen. Right-click on the particular network and select 'Follow', and then 'TCP Stream.' You can see that all the data is secured in the encrypted form. After that the show and save data can have many choices