**CSC 302 Computer Security**

**Examining the Network Security with Wireshark**

**1. Objectives**

The goal of this lab is to investigate the network security using network protocol analyzer Wireshark.

**2. Introduction and Background**

The Wireshark network protocol analyzer (former Wireshark) is a tool for capturing, displaying, and analyzing the frames, packets, and messages that are exchanged in a network. The Wireshark package can be downloaded from *http://www.wireshark.org/download.html*. Download the latest version. Note that in some computing environments, such as MS Windows, it is necessary to install a separate file capture utility (WinPcap for MS Windows). This utility is included to the latest version of the Wireshark installation package.

One’s understanding of network protocols can often be greatly deepened by “seeing protocols in action” and by “playing around with protocols” – observing the sequence of messages exchanged between two protocol entities, investigating the details of protocol operation, and causing protocols to perform certain actions and then observing these actions and their consequences. In this lab, you’ll be running various network applications in different scenarios using a computer on your desk, at home, or in a lab. You’ll observe the network protocols in your computer “in action,” interacting and exchanging messages with protocol entities executing elsewhere in the Internet.

In this lab, we will investigate the following protocols:

Telnet ()

SSH (Secure Shell)

HTTP (Hypertext Transfer Protocol)

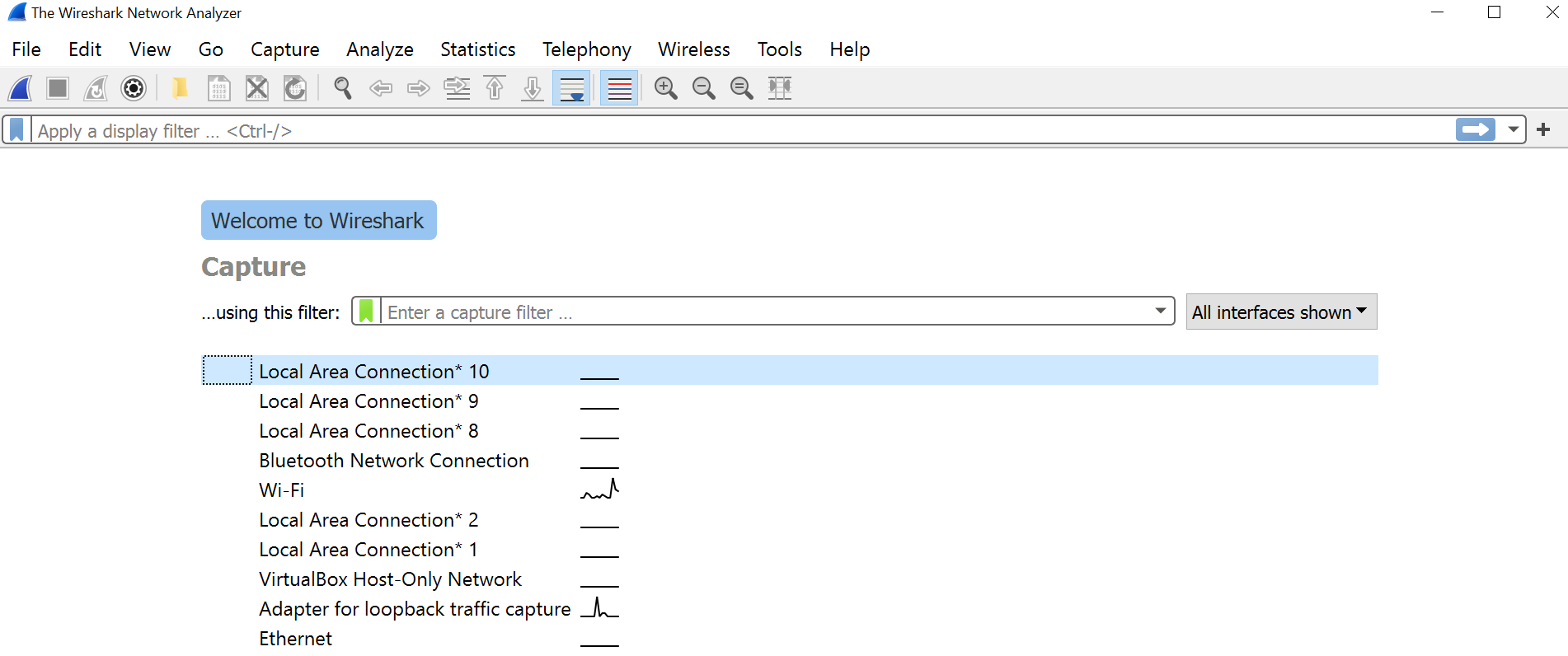
HTTPS (Hypertext Transfer Protocol Secure)

**3. Getting familiar with Wireshark**

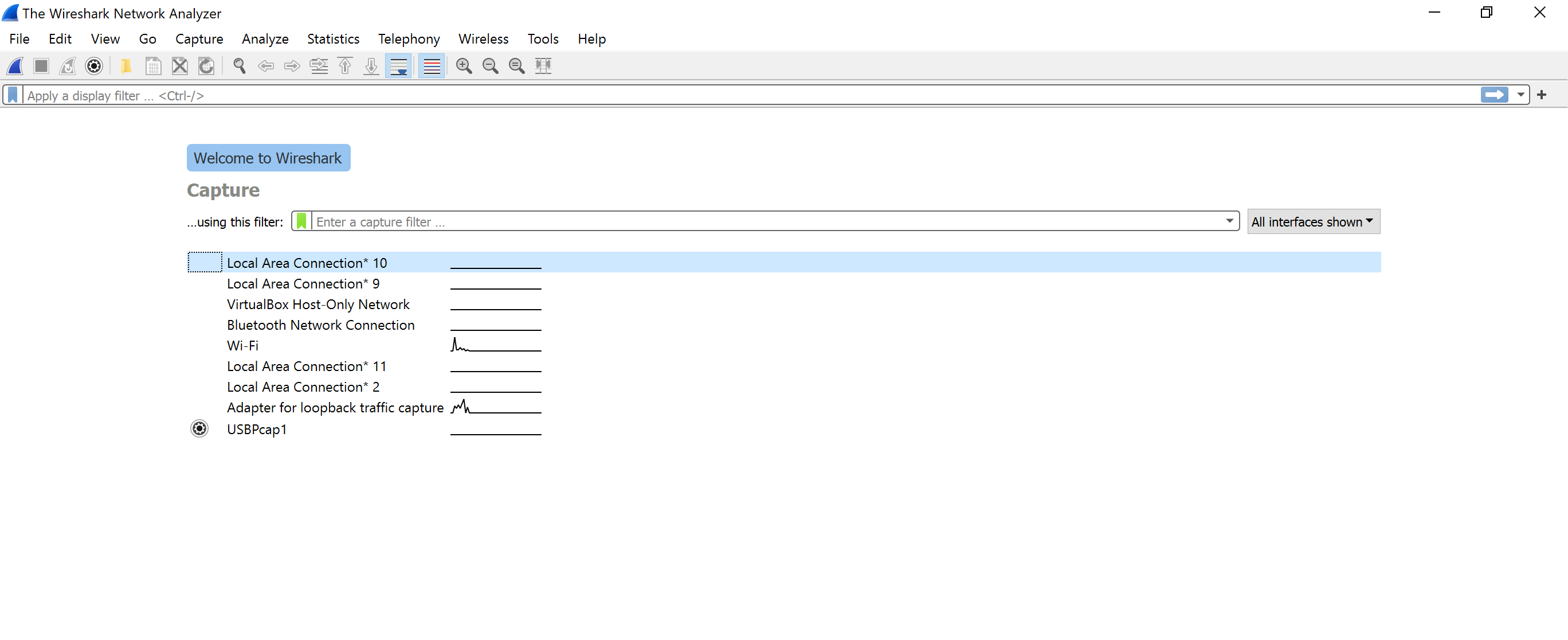
Wireshark allows you to capture real time packets as well as analyze already captured network traffic (pcap file). In this step, let’s try both ways.

PCAP (Packet Capture) is an application programming interface (API) that captures live network packet data from OSI and TCP/IP model. Wireshark creates .pcap files to collect and record packet data from a network.

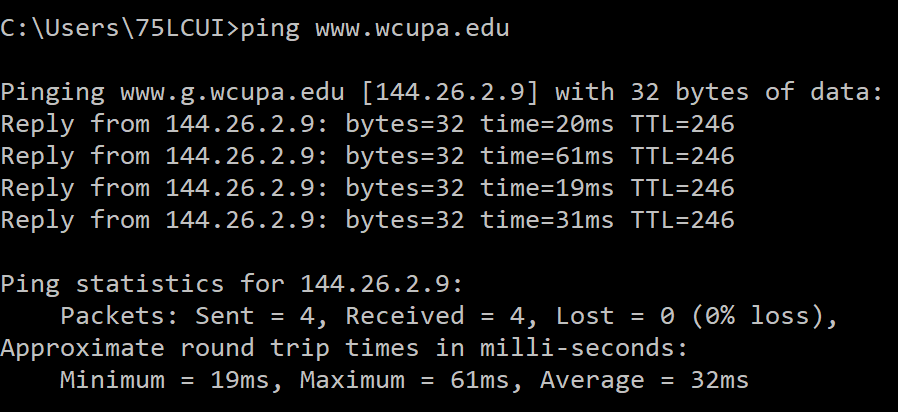
Here is the homepage of wireshark. After a successful installation, you can analyzing and capture packets from this interface.



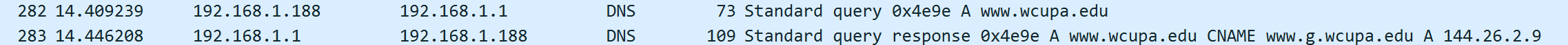
**3.1 Capture real time network traffic**



Now, let’s capture the real time network traffic by double clicking the interface “Wi-Fi”. How many packets do you have right now? Several hundreds of them? We may want to only analyze some of them. That’s is why the filter is so important. We could filter the traffic by the Protocol (TCP, HTTP, etc), or by port number (port:23). Let’s try to find some “dns” traffic by typing “dns” in the filter. Maybe you already have some dns traffic, is not, please open the command prompt, and say “ping [www.wcupa.edu](http://www.wcupa.edu)”.



Now, let’s go back to the Wireshark, and you may have packets like this, where the first one is a query of [www.wcupa.edu](http://www.wcupa.edu), and the second one is a response from [www.wcupa.edu’s](http://www.wcupa.edu's) ip address 144.26.2.9.



**3.2 Pcap files**

After you capture desired packets, please save it as a .pcap file, so you will analyze it later. Let’s try open a Pcap file. You could open a .pcap file that you captured before or download the one I provided on D2L. Then, click File->Open->(find the location and the name of the file)

To close the file, click File->Close

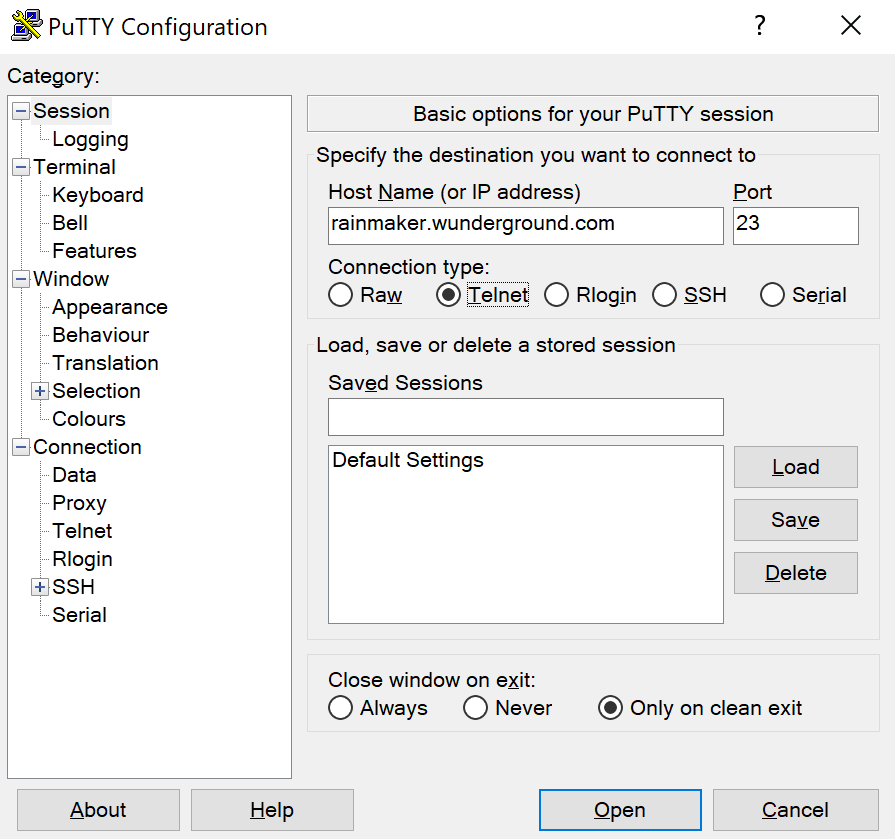
**4. Remote Connection (Unsecure vs Secure)**

Network traffic are either encrypted or unencrypted. Unencrypted traffic allows the eavesdroppers read the plain text message that you send through the network, while encrypted traffic only display the ciphertext.

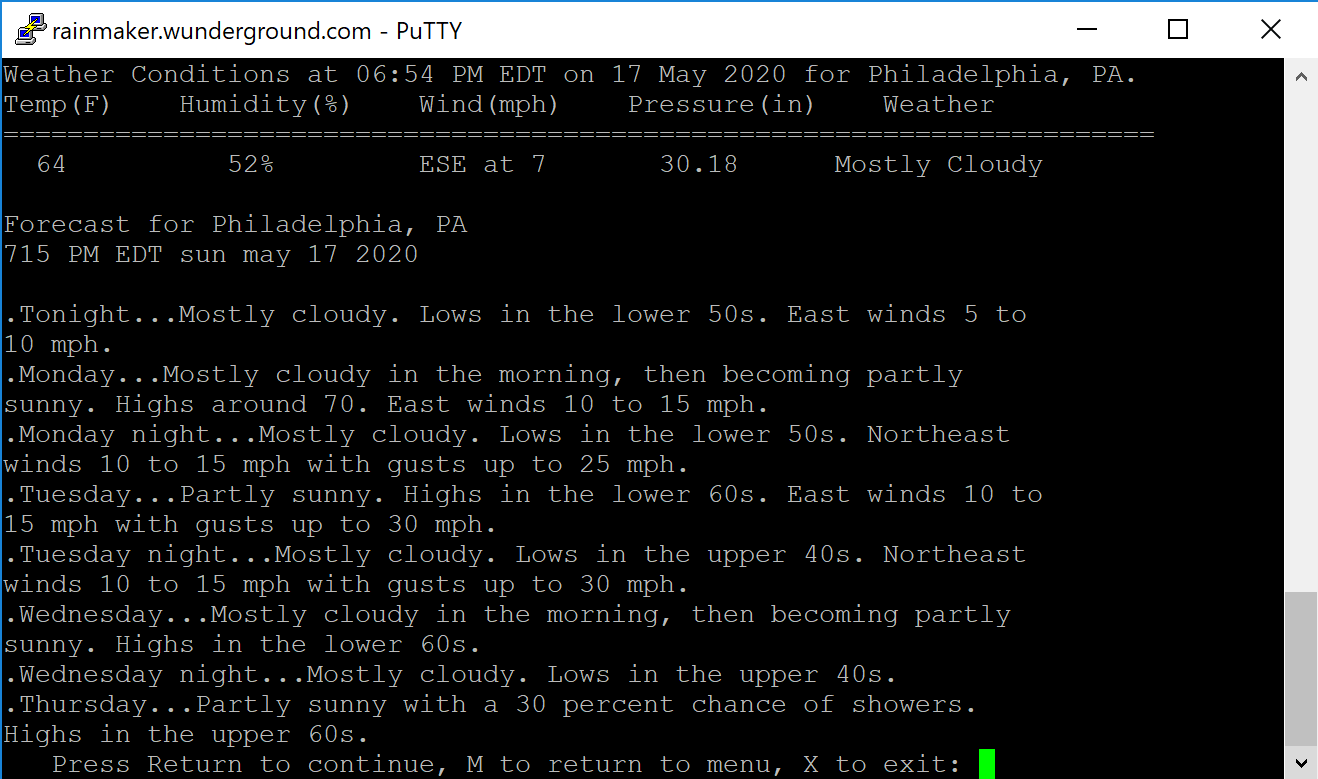
**4.1 Telnet (Unsecure)**

Telnet is a network protocol that allows a user on one computer to log into another computer. Telnet does not provide encryption, meaning all communications between two ends are transmitted as plain text message, which may leak information to the eavesdropper.

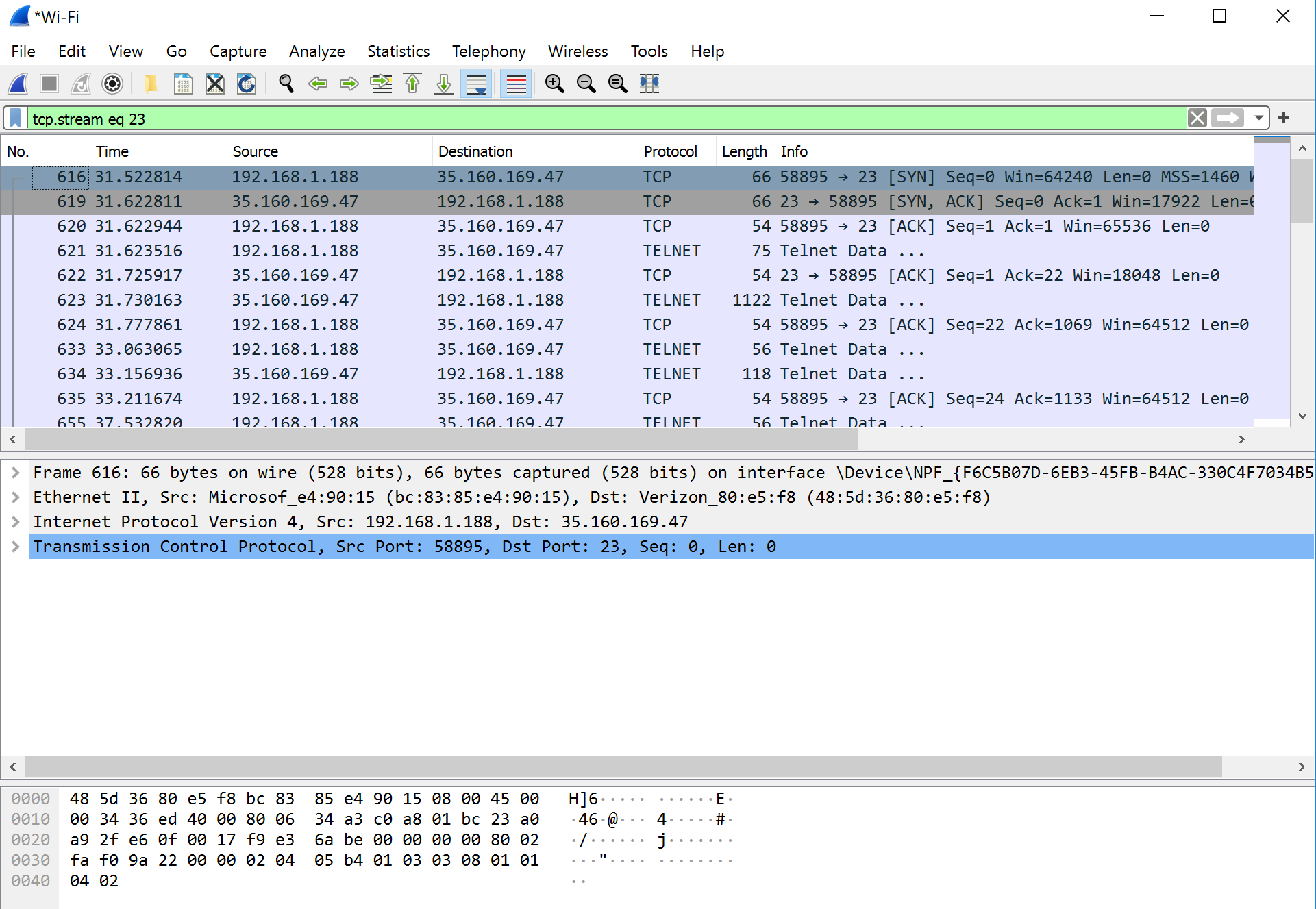
Let’s telnet to a webpage that gives weather report: rainmaker.wunderground.com. If you are a Windows user, please use Putty for telnet as bellow. For Mac users, you may need to install Telnet first, since Mac removed Telnet services from modern versions of system software, including macOS Big Sur, Catalina, Mojava and macOS High Sierra. Here is how to install Telnet and Telnet another device: https://osxdaily.com/2018/07/18/get-telnet-macos/



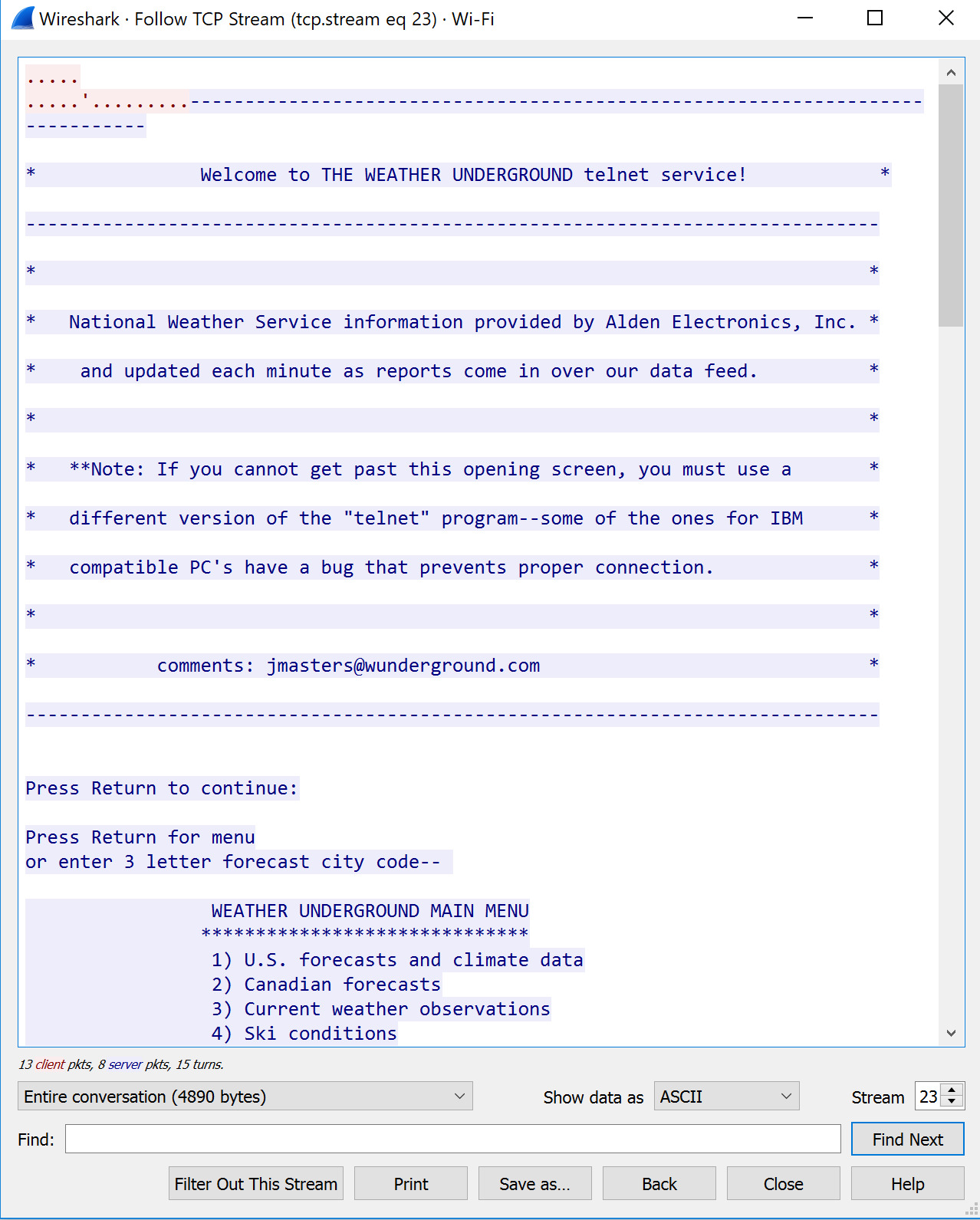
After you successfully connect to the server using Telnet, please press “enter” again for menu, enter 1 for U.S. forecasts and climate data, enter 3 to display 3-letter city codes for a selected state, enter “PA” for Pennsylvania, and then enter m to go back to the main menu, enter PHL for Philadelphia. Now, you should see the weather report. Apparently, the weather forecast is not up to date.



Let’s go back to the Wireshark. You may have multiple TCP and Telnet packet when filter Telnet or port 23 traffic like this. Telnet is an application layer protocol which uses TCP as the transport layer protocol (TCP).



Let’s right click the first Telnet packet, then choose “follow”, and then “TCP stream”. You will see a pop up window that display everything you see from this Telnet connection. It is what an eavesdropper sees when it listens to the communication between two end devices using Telnet. As you can see, the eavesdropper reads all the communication between you and the server as plaintext messages.



Since Telnet is not secure, not many services use this protocol anymore. Here are two lists of telnet servers for you to try out: <https://store.chipkin.com/articles/telnet-list-of-telnet-servers>, https://www.telnet.org/htm/places.htm

Please start a wireshark capture and make a telnet connection to any of the telnet server that you are interested in. Then, show the plaintext message in the wireshark by right click the packet, then “follow”, then “TCP Stream”. Please attach an screenshot of the plaintext message here.

**The weather wasn’t working, so professor sent me a list of Telnet servers, the one I used was freechess.org**

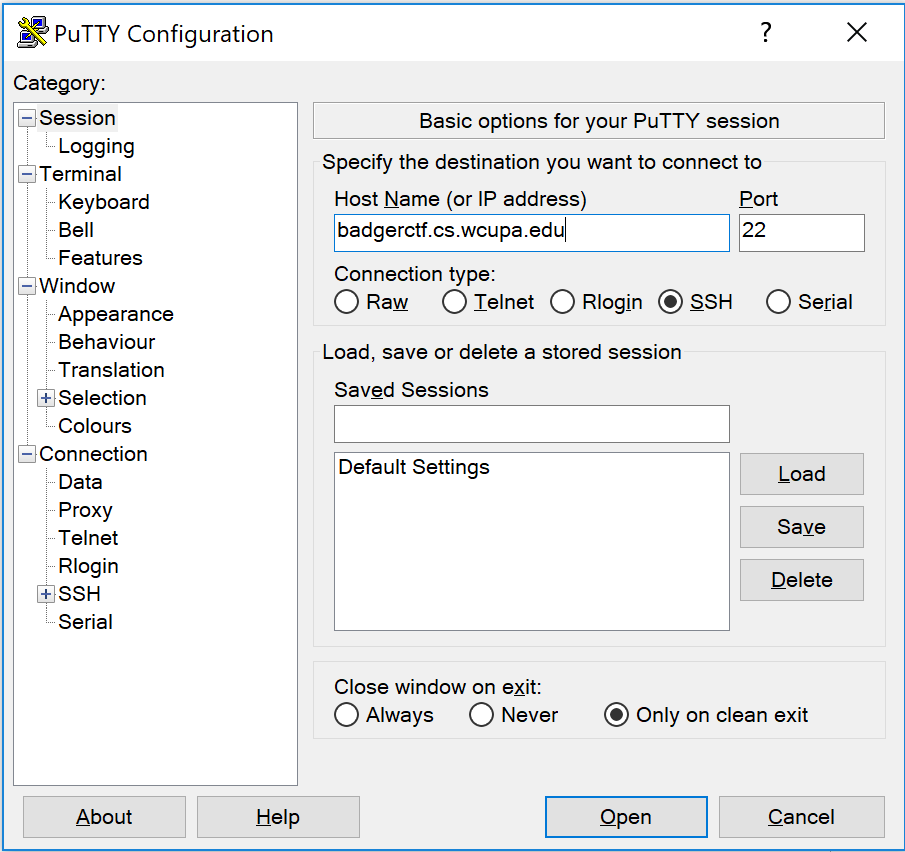
**A screenshot of a computer

Description automatically generated**

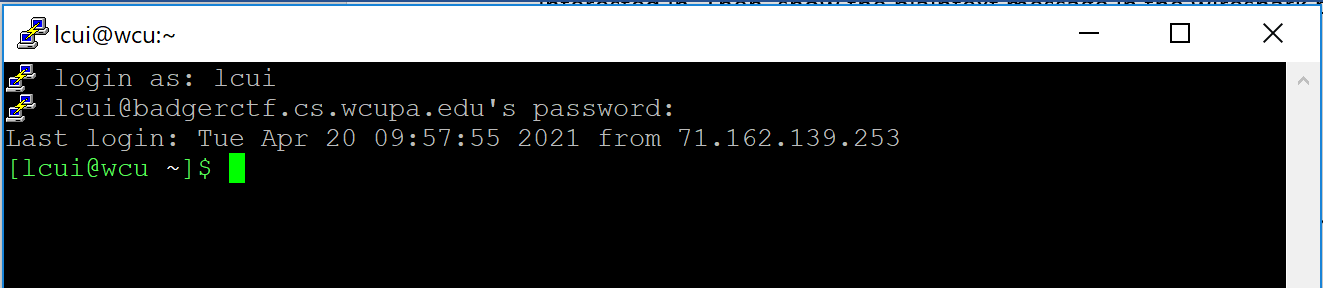
**4.2 SSH (secure remote connection)**

SSH (Secure Shell) is another way for remote log in. It is a cryptographic network protocol for operating network service securely over an unsecured network.

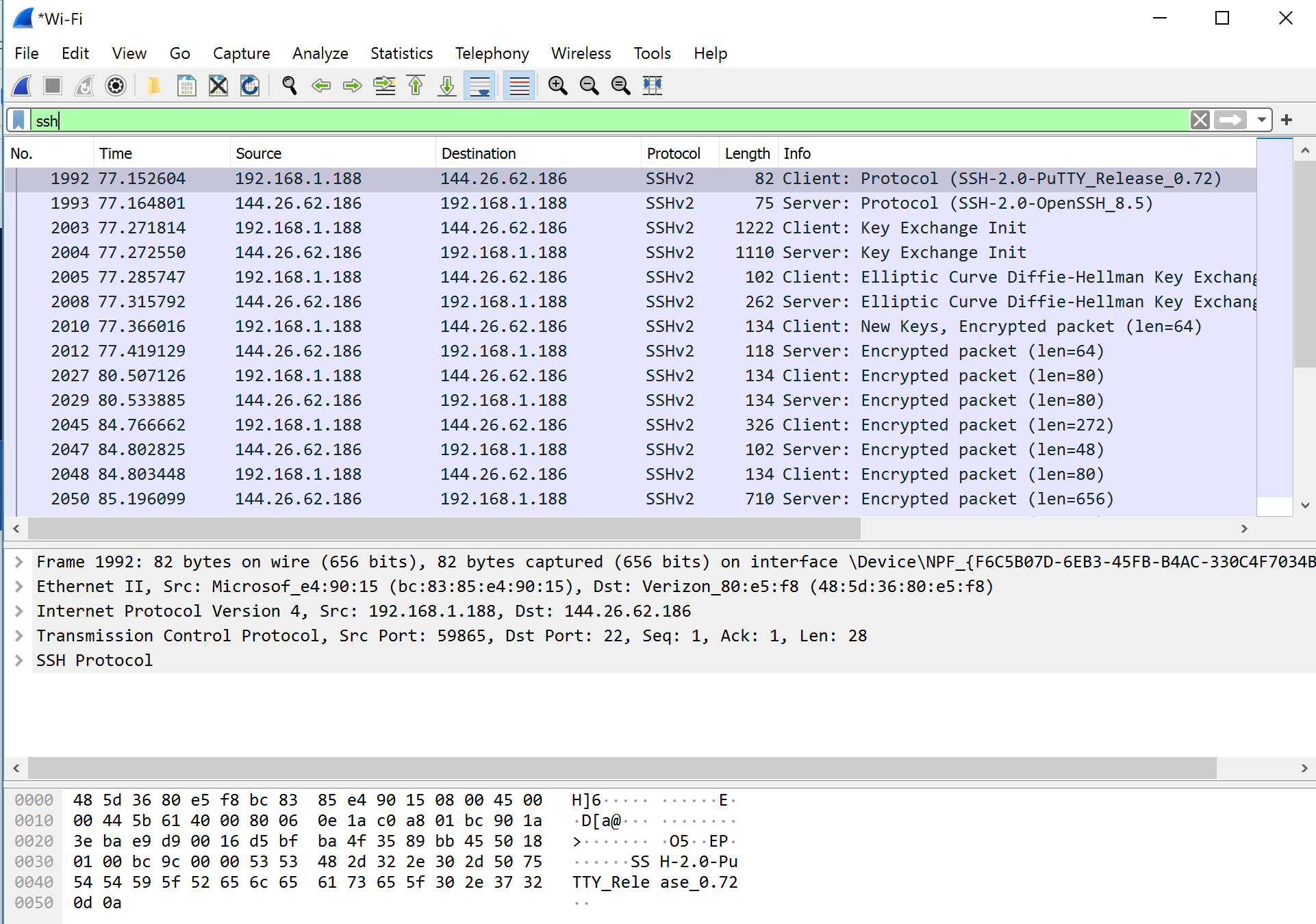
Let’s start wireshark first then filter ssh. After that, open Putty to remote login to the CS services, like badgerctf.cs.wcupa.edu server. Here is a list of servers in the department you could try to connect: <https://www.wcupa.edu/sciences-mathematics/computerScience/labs_facilities.aspx>. Please use the same structure as the host name: *servername.cs.wcupa.edu*. For example, molly.cs.wcupa.edu; taz.cs.wcupa.edu



After you successfully login to the session as, go to the Wireshark.



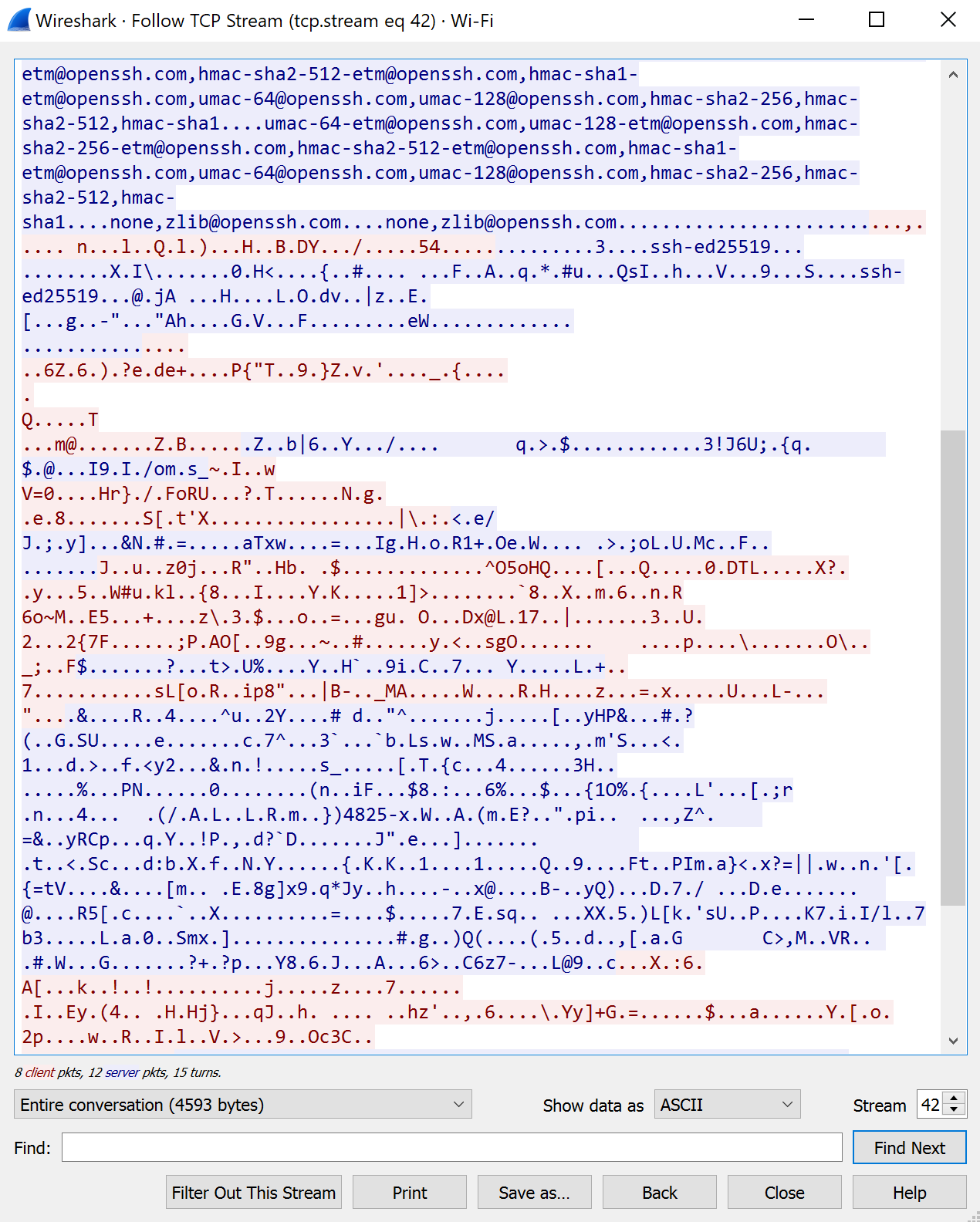
In the Wireshark, please stop the capture. Let’s first look at the source and destination part. Clearly, the communication is between my machine (192.168.1.188) and the badgerctf server (144.26.62.186). The protocol used is SSHv2 (SSH version 2). The Info column provides a lot information, such as which message comes from Client, and which message comes from Server. The purpose of the packets, such as Key Exchange Init, Diffie-Hellman Key exchange, New Keys, then everything is encrypted.



Now, let’s go to statistics above the tool bar, then select conversations. Since both Telnet and SSH use TCP. Let’s click TCP. Within multiple TCP packets you may have, one is for the ssh connection between your machine and the badgerctf server as:



Click it and click “follow stream” button below. You will see a pop-up window like this with meaningless characters inside. Those meaningless characters are ciphertext message, which the eavesdroppers see.



Please start a wireshark capture and make a ssh connection to any of the server that you are interested in. Then, show the plaintext message in the wireshark by right click the packet, then “follow”, then “TCP Stream”. Please attach a screenshot of the cypher text message here.

A screenshot of a computer

Description automatically generated

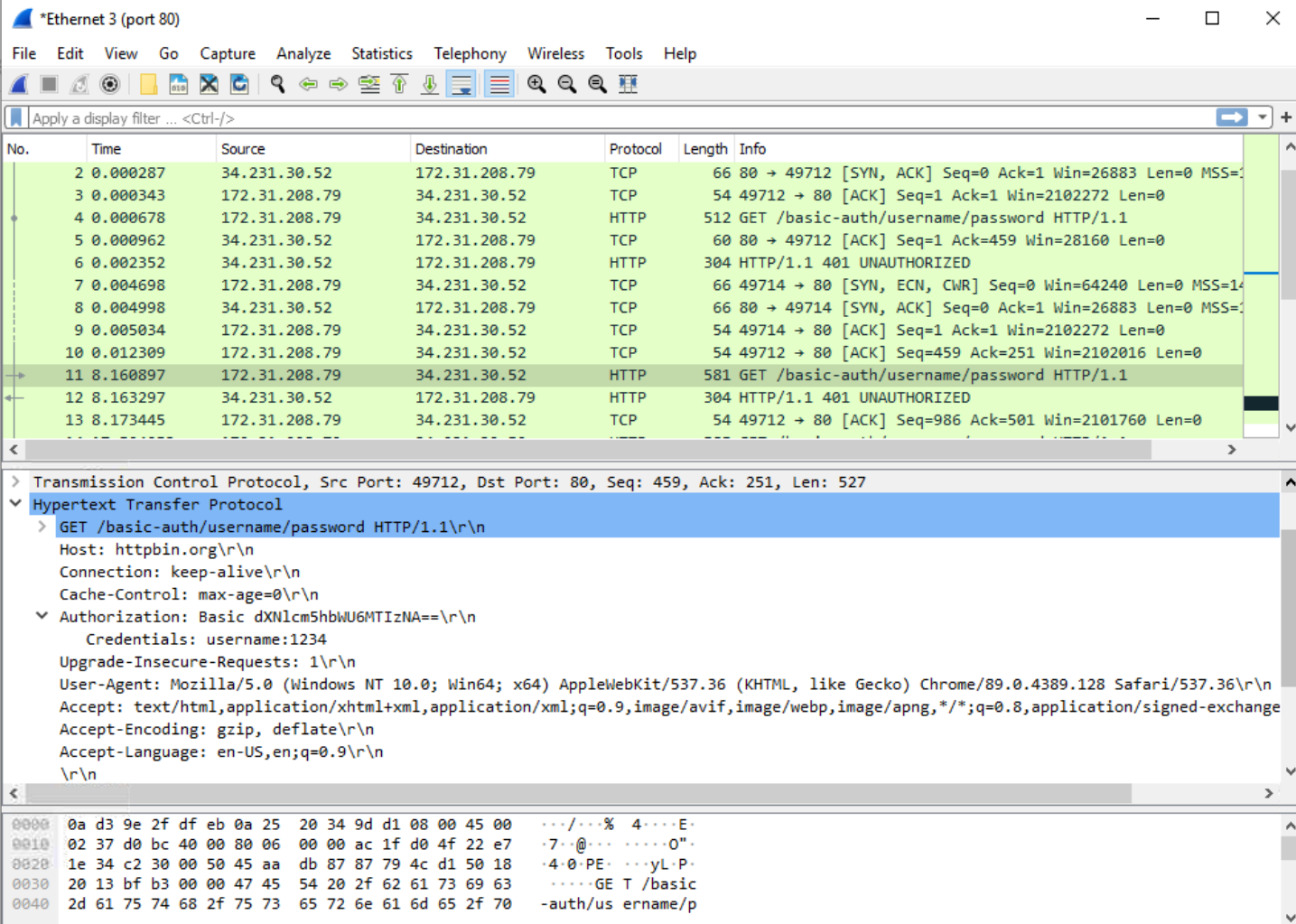
**5. Web traffic (unsecure vs. secure)**

**5.1 HTTP traffic[[1]](#footnote-1)**

First, go to this webpage: <http://httpbin.org/basic-auth/username/password>

The correct credential is username: username, password: password. Let’s first try an incorrect password: 1234, then enter the correct one. As you can see here, the username and password you entered in the text box are clearly shown in the “Authorization” part.

If the above webpage does not work for you, try this one: http://vbsca.ca/login/login.asp



Please show a screenshot of the username and password you entered. Please use a different password from 1234.

**When I came to your office hours, we discussed this was acceptable I wasn’t getting the GET/basic-auth/ API back**

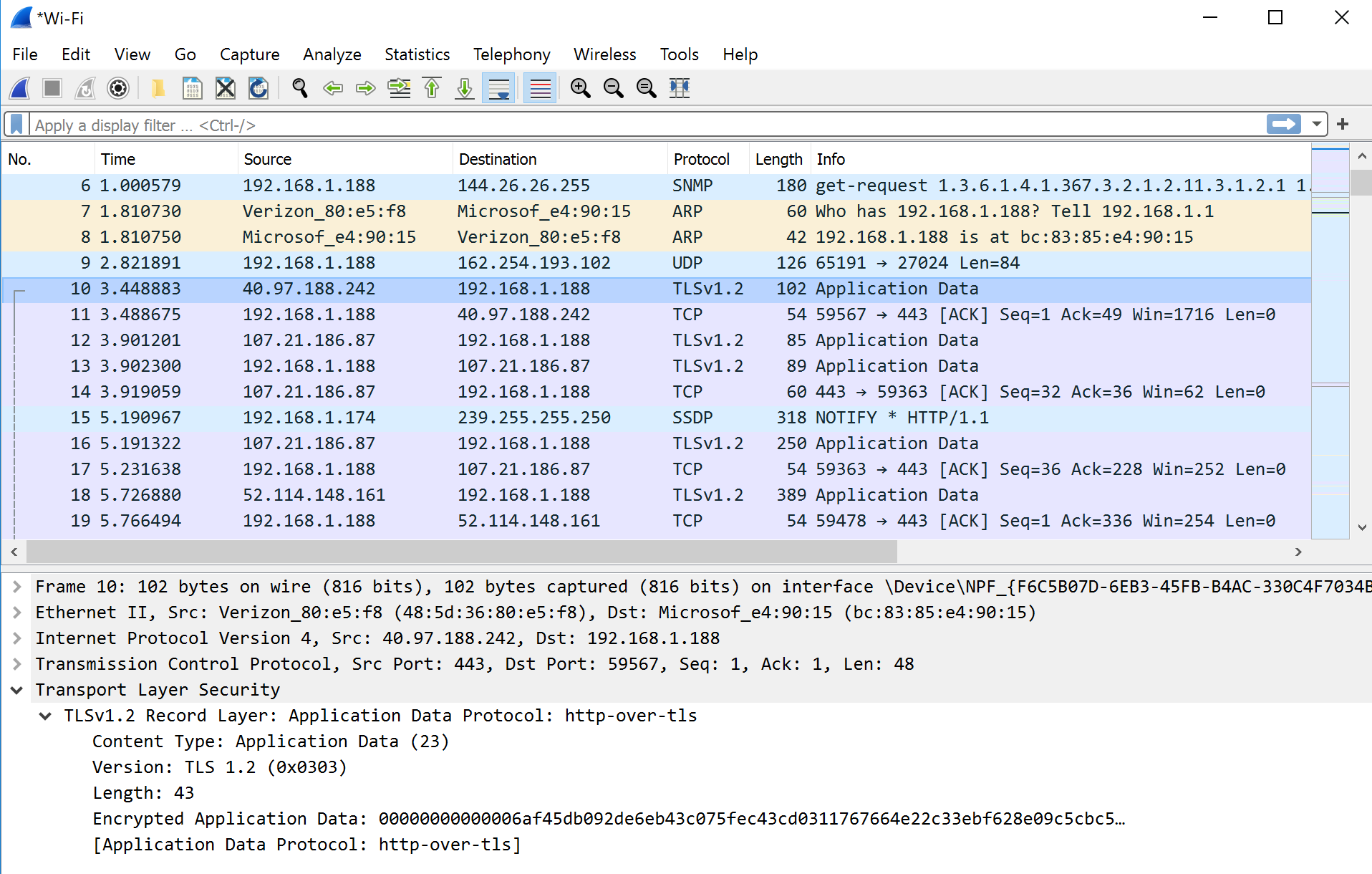
A screenshot of a computer

Description automatically generated

Bonus Project (2 extra points): If you are interested, please find a website that still use http instead of https and “search” a term in the webpage and see whether you can see the plaintext in the wireshark.

**5.2 HTTPS**

Start capture, and open a webpage like [www.google.com](http://www.google.com), [www.amazon.com](http://www.amazon.com), etc. Then filter the port 443 for HTTPS or find protocol “TLSv1.2”. Expand Transport Layer Security and then TLSv1.2, you will see the application data protocol is http-over-tls, which is HTTTS. The application data is Encrypted Application Data. The meaningless characters after the “Encrypted Application Data” is the ciphertext message.



Please attach a screenshot about an encrypted application data in TLS packets.

A screenshot of a computer

Description automatically generated

1. Modified from Coursera Guided Project “Wireshark for Basic Network Security Analysis” [↑](#footnote-ref-1)