TECH3812
Intro to Networking
Lab #7

Name: _____

While connected to the network (internet):

Ipconfig is a windows console application that shows the current TCP/IP configuration of your network interfaces (usually just one for desktop computers). To use ipconfig, go to the START menu, type into the search bar **cmd**. This brings up a command prompt from which you can do command line functions, like ipconfig.

Type **ipconfig**

You should get an output which includes your IPv4 Address, Subnet Mask and Default Gateway. Record those below:

IPv4 Address	
Subnet Mask	
Default Gateway	

Another useful command is **ping** which is another console application that tests the connection between two computers on the network. Using the same cmd window already opened (or open a new one if your closed it)

Type **ping google.com**

Ping sends a packet to the host given and then the host sends the same packet back, it measures the time it takes for the round trip. If you get a "timeout" the message was lost, indicating a connection problem between the two computers.

It also gives you the IP Address of any domain name (an IP address is the xxx.xxx.xxx.xxx formatted address while domain names are text identifiers for that are assigned to one (or more) ip addresses)

Given the output you received determine the following:

Domain name	
IP Address of the domain name	
Average round trip time	

Q. Some servers turn off the ping response. Why do you think they might do this?

Another useful console application is **tracert** (short for "trace route") which shows the path the packets take from Point A to Point B. When a packet travels, it usually has to go through a number of devices called routers. Trace Route reports each time the request packet hits one of these devices (called a hop) and reports the device IP and/or domain name back to you.

Now type **tracert google.com**

How many hops did it take to get to google.com?	
What was the first ip/domain name that was NOT on the 192.xxx.xxx.xxx network?	

Yet another useful console command is nslookup. This utility looks up a domain name and returns the IP address of that domain name.

Now type **nslookup tech-uofm.info**

What DNS Server responded (name and address)?	
What is the IP Address of the tech-uofm.info server?	

Now download Wireshark from http://www.Wireshark.org/ and install the program and all components (including WinPcap which is required to run Wireshark).

Wireshark is a network protocol analyzer that snoops packets on the network and records them. You can go in an analyze packets to see the various parts that make up the packet and the encapsulation.

Create an Isolated Network:

Before proceeding, we are going to create our own isolated network. Disconnect the ethernet port on the back of the computer and connect it to the 10/100 HUB provided by the instructor.

Since we are no longer connected to the network, we will have to give each PC it's own static IP address. To do this, type in "ethernet settings" into windows search. Now click on "Change Adapter Options" and double click on the Ethernet icon. On the new window, click on Properties. Now click on the line that says "Internet Protocol Version 4 (TCP/IPv4) and hit the properties button. Currently this is set up to Obtain an IP address Automatically, but since we will have not DHCP server on our network, we will change this to "Use the following IP address" and we will use 192.168.0.x for the IP address (were x is a UNIQUE number for your machine – no other computer can have that number or the network will not work). The subnet mask should automatically fill in with 255.255.255.0 and the others we do not need for this experiment.

So now the network will look like (note that one of the PCs will be an instructor configured Raspberry Pi Server):

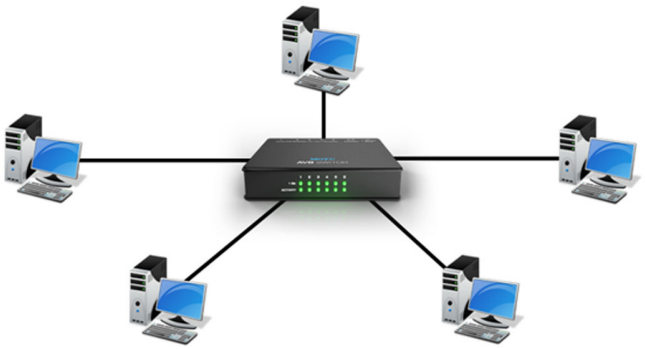


Figure 1- Isolated Network

To ensure that the PC's are configured properly, do a ping to each of the computers on the network.

NOTE: Win 10 and above has ping response turned off…to turn it back on do the following: In the Windows search bar type in " Advanced Security " (you really want "Windows Defender Firewall with Advanced Security"). Off to the left, click on "Inbound Rules". Scroll down the list until you find "File and Print Sharing (Echo Request – ICMPv4-In)". There should be two lines (one for Public and one for Domain). Click on both lines until you see a green checkmark by the line. This will allow your pc to respond to a ping request.

Once everyone in the class has verified they can ping everyone else, Open **Wireshark**. You should see something similar to this on your screen:

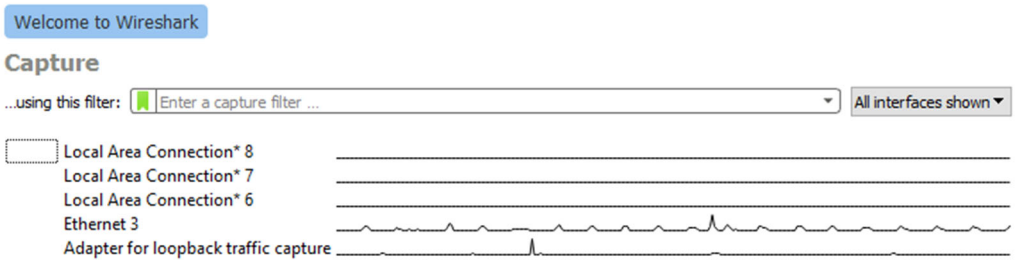


Figure 2- Wireshark available interfaces

This is showing the network interfaces that wireshark has detected (you might have more or fewer, but you should have at least one that says "Ethernet x". Click on that one to select that interface.

On the menu bar click on **Capture** then **the start**. This will start Wireshark capturing packets on the network.

Since we are on an isolated network, you should not see a lot of traffic (although it is still possible that one or more of the computers will be sending packets for various reasons).

Ask ONE student to do a ping to another computer. Since we are connected to a HUB, everyone should see the packets generated.

Once the ping command has finished, Stop the capture by using the menu option **Capture** | **Stop**.

The ping packets will be identified by the Protocol of ICMP as shown below:

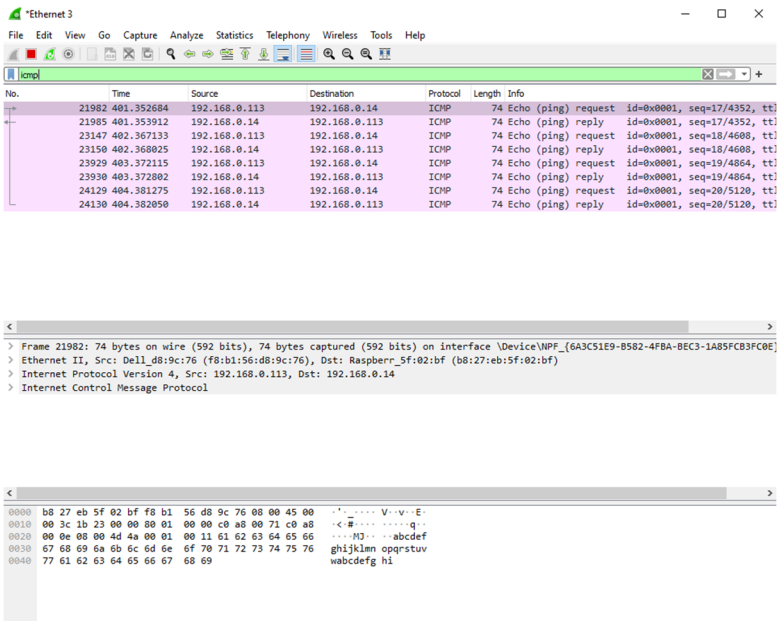


Figure 3- Typical Ping capture

As you can see if Figure 3, you should see three windows:

The first (top) window is a list of packets including the source, destination and protocol. In the example, 192.168.0.113 is the computer doing the ping command and 192.168.0.14 is the responding computer.

The second window shows more detailed information about the packet that has been selected. You can click on the ">" at the start of each line to dive deeper into the packet.

The last (bottom) window shows the hex values of the packet itself. Note that if you select a item in the 2nd window, this window will show the bytes of info in the packet the contains that information.

It should be noted that the textbox above the first window is a Filter. In this case a filter has been applied to show only ICMP packets. Wireshark has MANY filters that allow you to find just the packet(s) you are looking. Remember we are on an isolated network for this lab, when connected to the internet it is possible to get thousands of packets in a very short period of time, depending on network traffic and how your pc is connected to the network.

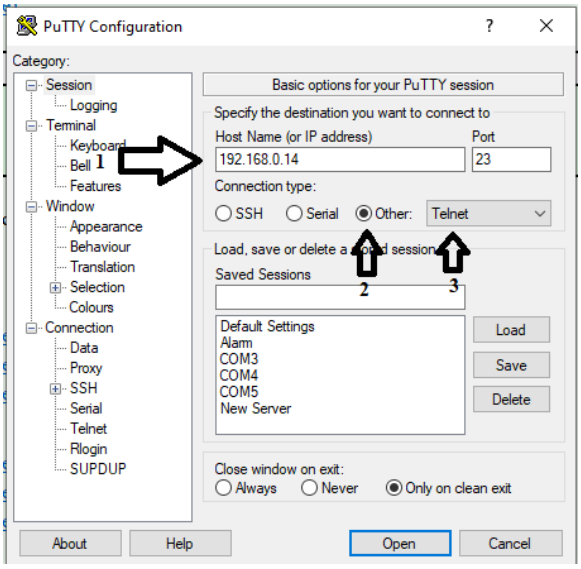
Select one of the ping packets in the first window then capture the wireshark screen and paste it below:

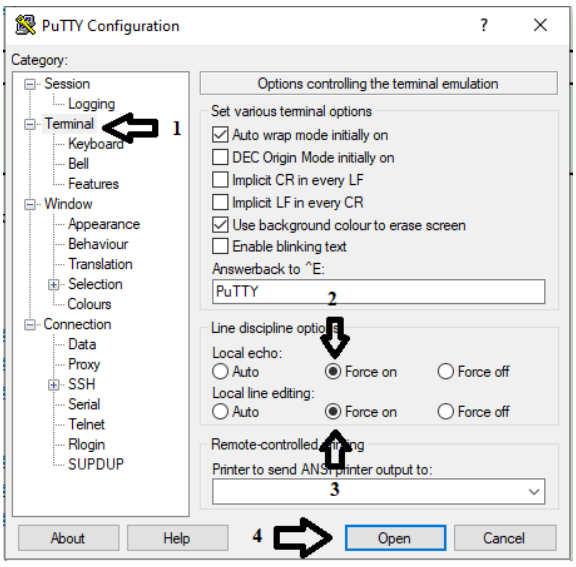
Insert Capture Here

Fill in the following information:

What was the source computers address?	
What was the source computers mac address?	
What was the destination computers address?	
What was the destination computers mac address?	
What is the **data** within the ICMP Message?	
What is the checksum?	

Now open putty. Set up putty in the following manner:





(The left hand image sets up a telnet session to 192.168.0.14 (the Raspberry Pi) and the right hand image sets it up so it only transmits a packet when the enter key is hit).

Before hitting "open", start a new wireshark capture, then hit the "open" button

Now Log into the PI using the following:

Login: pi
Password: tech3821

Stop the wireshark capture.

Add the filter "ip.addr==192.168.0.14 && ip.addr==192.168.0.xxx" (no quotes and xxx is replaced with your PC's IP address) in wireshark to filter all but the traffic to and from that IP address.

Starting from the 1st packet, look for a TELNET packet that contains "raspberrypi login:" in the data. It should be from your PC to the PI.

Now look for the next Telnet packet going from your PC to the PI.

Do a screen capture of the data window

Insert Capture Here

Starting from the 1st packet, look for a TELNET packet that contains "Password:" in the data. It should be from your PC to the PI.

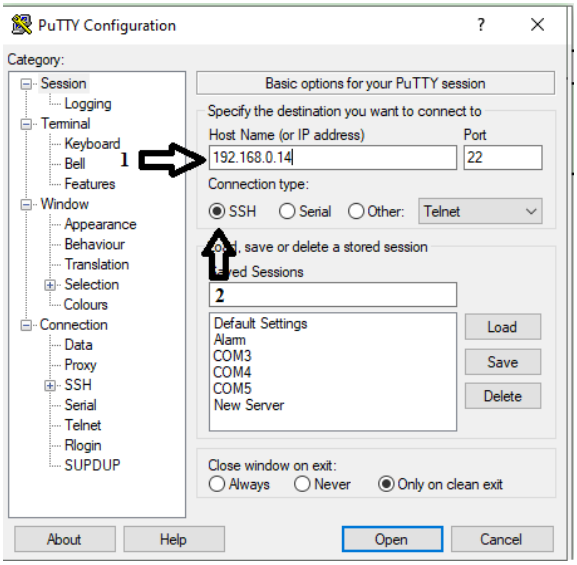
Now look for the next Telnet packet going from your PC to the PI.

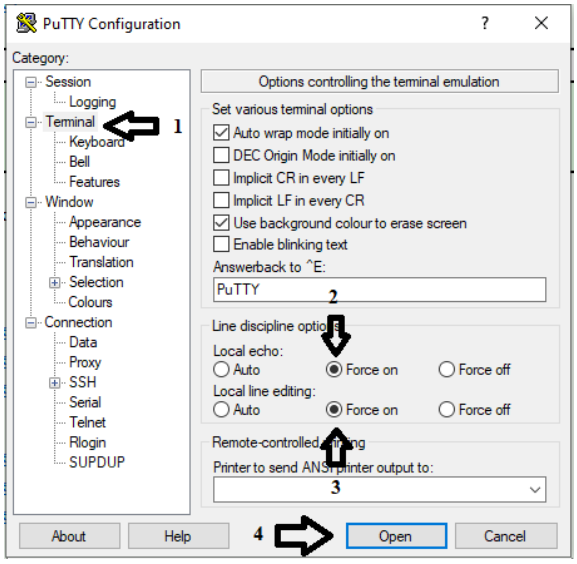
Do a screen capture of the data window:

Insert Capture Here

As you can see, anyone snooping the network can see the name and password in the data of these packets. This is why TELNET is NOT SECURE and has been replaced by SSH TELENT.

Close the putty window and then reopen it using the following setup:





(The left hand image sets up a SSH Telnet session to 192.168.0.14 (the Raspberry Pi) and the right hand image sets it up so it only transmits a packet when the enter key is hit).

Once again, start a wireshark capture, log in and then turn off the capture.

Now look at the wireshark capture and try to find the login info.

Q. Did you find the login info?

Take screen capture(s) the data from the packets that contained any readable text (if the packet data goes beyond 00a0 in size, only capture to that point):

Insert Capture(s) Here

It should be noted that info in the first wireshark window should have said something about "Key Exchange" for two of the packets captured. SSH is an encryption protocol that allows messages to be

passed between machines without the ability to read the packets containing the user transmitted info. This is why you cannot find the name and password (unlike telnet).

Now open a browser and then start a new Wireshark Capture. In the bowser's url line type 192.168.0.14. This should show the following website:

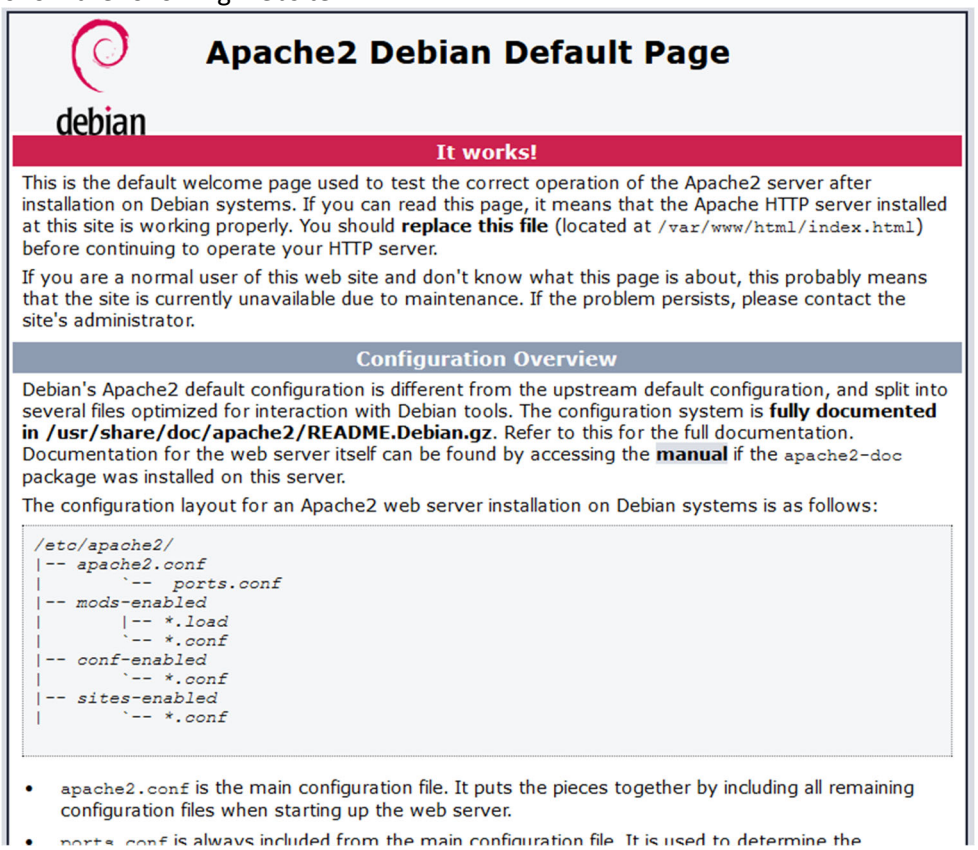


Figure 4- Default Webpage

Now stop the Wireshark capture.

Add the filter "ip.addr==192.168.0.14 && ip.addr==192.168.0.xxx" (no quotes and xxx is replaced with your PC's IP address) in wireshark to filter all but the traffic to and from that IP address.

The first 3 packets should be TCP Packets….these establish a "connection" between your PC and the Raspberry Pi Apache Server.

The next packet should be an HTTP packet with "GET / HTTP/……". Select this packet, go into the 2nd window of wireshark and expand out the Hypertext Transfer Protocol section and do a screen capture from the "Hypertext Transfer Protocol" to the line that starts "[full request url…."

Insert Capture Here

This text is actually what is sent from the browser to the server to retrieve the website (this will come up again in latter labs).

In between the "GET / HTTP…." Packet and the "HTTP/x.x 200 OK" packet is the actual website itself (being sent in multiple packets.

Lets do a search for some of the test found on the served webpage. To do this in wireshark, do an Edit | Find Packet. This will bring up a new menu below the filter that looks like this:

Select "Packet Bytes" from the first pulldown, Narrow & Wide from the second pulldown, String from the third pulldown and put in "fully documented" (no quotes) into the textbox and click find.

This should take you to the packet that contains that exact text.

Do a screenshot of the bottom window (hex / ascii dump):

Insert Capture Here

Looking at Figure 4 and the above screenshot, locate the text you searched for in the packet. **Answer the following:**

1. What are the 3 characters before the words "fully documented"

2. Looking at Figure 4, speculate what these 3 characters do (formatting wise)

3. Continuing with Figure 4 and your screen capture, speculate what ends that special formatting?

4. Using your phone or laptop connected to the internet, do a search for "HTML" and your answer in Q1 and verify your answer in Q3. Give the url of the website: