**Packet Capture and Analysis**

**Homework #2**

**Due Date: 4-16-2015 at 10am (was extended)**

**by David Fullerton**

**Professor: Dr. Omprakash Gnawali**

**COSC 6377: Computer Networks  
Spring 2015**

**MW 1-2:30pm AH 110**

**Packet Capture and Analysis**

**Introduction:**

This assignment focuses on learning about multiple open source tools that are available for capturing packets from the local network and / or wireless WiFi networks. This assignment requires us to use and learn to understand three different software tools. Then, we are required to discuss and evaluate their performance from our testing of the applications.

* 1. **What were your three selected networking tools?**

This assignment requires us to select three different open source tools to capture packets from our network interface. Thus, the applications[[1]](#footnote-1) that I chose are the following:

1. **Wireshark v1.12.4**. It is an industry standard tool that I have heard about many times, and I have never had a chance to work with it. After using a few minutes, I immediately decided to use this application to be one of my primary tools for gathering network information.
2. **NetWalk**. This is a very nice application which has incredible monitoring tools. This application was easy to understand and clearly designed well, so I also selected this application.
3. **Windump**. After striking dead-ends on many applications that I downloaded and tried from CNET.com, I asked my fellow classmates as to what they were using, and they indicated that this was an easy one to use. And, it took a while to understand how to get it working, but I studied the website and was able to adapt to using it rather easily.

However, I would like to discuss the other applications that I tried, but I could not get to work. So, this project might be easy for someone who has done this, but there are a lot of false starts and wasted time for people who do not yet have this skillset or know which applications to use.

The below list provides which applications not to use:

1. **Capsa Packet Sniffer**. This was a highly recommended application on websites, and it was impressive as I was starting to learn it and use it. Yet, it was not able to save packets nor be able to be used in a wireless configuration, so I could not use $999 to buy the industry version. Thus, I had to quit using it and switch to something else.
2. **BitAware Network Monitor**. I tried to use this application, but it was out-of-date and did not have a Windows 7 version (nor recommendation) since 2008. Thus, it gave me run-time errors and would not even run, so I could not use it.
3. **Compass Free**. This looked like a very nice application, and I studied the documentation and it truly can help someone analyze PCAP files. Even though it is highly recommended and lauded by the industry, it does not have a “live capture” capability, so I had to decide against using it but wished that I could have tried some of its many tools and capabilities for creating reports after a network analysis is completed.
4. **IONinja**. I started using this wonderful application, and it was easy-to-use and very designed to be easily understood in through a well-designed graphical interface. Yet, IONinja outputs files that are not read by anything else (a “.NJLOG” format), so I did not have time to figure out how to access the data that was stored in its proprietary format. And, it does not output text nor PCAC files. So, I quit using it after learning how to get four file traces.
5. **EtherDetect Packet Sniffer v1.4**. I worked to install this application, but the installation package aborted because its version does not support Windows 7.
   1. **What are the main differences?**

As I am early in learning this different tools, I find that there are many differences, which are too numerous to list. Each of the tools are designed differently, but all share a custom Windows interface (other than WinDump which is command-line driven). The applications are all pretty easy to understand and adapt too, yet they have so much capability that it is really hard to gain a comprehensive appreciation of them without much time being able to spend using them in a real situation.

With the different tools, I spent time just selecting three different applications that would actually work in Windows 7. Many of the applications were written for earlier versions of Windows, which is shocking that they would still be offered on a popular open-source download website. Yet, I can understand (maybe) because they are free, and the authors are not required to update them with every new version of Windows. And, since they are free or almost free with shareware, the authors are about as likely to get incredible compensation for their work as is Senator Cruz becoming the President of the United States, so they don’t really invest much in them to keep them up with the times, especially in the face of competing with something like Wireshark which has all the capabilities in the world.

My immediate observation is that these are incredible tools, but it would take many networking projects and problems in the corporation network to be able to have opportunities to learn in-depth how each of these work and when is the time to use just one of them, more than one of them, or all of them. In actuality, I found that Wireshark seems to be the most capable of the tools.

* 1. **How do these network tools work?**

From my somewhat limited understanding, I will explain how I see them working. Of course, you install the application, then you run the application in Windows. Therefore, I am assuming that the application is correctly installed and is opened to the opening window / screen. Well, on all of the programs, you need to set up the options for the **Interface Configuration** as to how you are going to capture network packet flows. So, I always selected the one NIC (Network Interface Card) that I have, which is a Intel 82579V Gigabit Network Connection on a Local Area Connected network, and it indicated that this would be the **Driver** that would be monitored. Then, I had to indicate a log file where I would store the recorded network traffic. And, I didn’t specify really anything else, because I didn’t know what else to specify. Then, I just selected “Capture Traffic”, and the screen would scroll as many captured packets would scroll by with properties of each packet that was being communicated across my gigabit network at home.

To answer it more specifically at a more technical level, I assume that the application will watch the network packets that are being transferred through the currently selected network adapter. But, I do not understand yet how anything like a library and / or a kernel within the Windows is relevant.

**2.0. After capturing six packet traces, what is commentary on performance of the tools?**

On my system, it is a pretty-fast rig that is designed to be fast, because I use my computer non-stop most days; therefore, it is hard for me to detect much slowdown on any of the networking tools. Each tool performed exceedingly well in my opinion. Each tool quickly gathered the network packets and stored them in the given log files, and the applications were very speedy in creating the log file sizes that were asked. So, I do not have any metrics as to how to compare, because on my system, each application worked speedily and flawlessly. I have a faster Intel processor, good amount of GB of memory, and an SSD for storage. Thus, any degradation of performance was not nothing that I could perceptibly notice nor could I honestly feel any difference what-so-ever between their various performances.

However, I can make some related observations that are related to this question. At home, I was surprised how much traffic is going over my home network. In just a few moments (say two or three minutes), the files were growing rapidly in size. I probably have more than 500k in no time. I am not sure what is consuming so much bandwidth and what is transferring so many packets. Granted, I have probably 50+ active devices (probably 60 max devices) on my home network, but I don’t think that all of the devices are currently transmitting data. Therefore, when I have time, I will have to analyze with these tools what is sending so much data. Honestly, I was surprised at how fast the screen was scrolling in Wireshark with so many countless packets going across the network. It was amazing to me, and I would jump into a deep analysis, if I were not so SHACKLED by my daily obligations and my GTA duties of grading countless assignments along with re-gradings that I count the moments until summer when I will be free again for doing other things. Hopefully, I won’t get another undergraduate programming course.

If someone did notice a difference in application performance, I would guess that it is the network being analyzed. If they have a network with not much traffic going across, then it might take longer to get enough packets generated to be able to analyze. And, if the computer sputters after a good number of network packets are captured, then they might have older hardware or a slower 5600-rpm hard drive on a laptop that would slow down the storage. And, if the entire application runs slowly, then they might have a slower system that is not able to keep up with the network monitoring. However, I am not certain as to what criteria with which this question is to be answered, so I gave the answer/s that I understood best modelled the motivation for insight behind the question.

**3.0. What are various observations about each network packet trace?**

During this assignment, this student collected 6 different packet traces. The traces were for both a wired and a wireless network on these three applications: 1) Wireshark, 2) NetWalk, and 3) Windump. Thus, the five observations per network trace that I can make are given in the below sections.

To derive the below observations, I wanted to develop a unique program to derive different analyses. However, I found that I could load each trace back into Wireshark, because I selected network monitors that output in the PCAC file format. This PCAC file format is a format that Wireshark can easily load-in, and then Wireshark will allow an individual to export the trace as an “exported packet dissection” in a CSV file, which I could read and work with easier in Microsoft Excel to save time and effort.

So, I wrote some Microsoft Visual Basic for Applications (VBA) code, which I am very fond of utilizing to get stuff done very simply and incredibly quickly. There is a “Read Me.txt” text file to explain how to run the VBA within the Excel application.

**3.1. What are five observations about the Wireshark network packet trace when wired?**

This is a tough question because there is so much that I immediately see when I am using a network sniffer. So, I hope that the below observations show a good analysis of the data[[2]](#footnote-2) for using a Wireshark network packet trace when having a wired connection:

1. **WASTED PACKETS**. There are an incredible number of packets that are just being wasted which is developed more below. Or, there are packets that I don’t think have any real use. There were a total of 11166 total packets captured but 10535 (94.4%) of them were between my computer and a network DVR device for recording television.
2. **PING PACKETS**. As I was testing in a command window during this packet capture, I can see multiple packets between my FileServer (192.168.1.100) and my current computer (192.168.1.108) which are doing echoing and pinging. This is ICMP protocol.
3. **GOOGLE GEO-LOCATION INFORMATION**. I found packets from my machine to this IP address of 74.125.227.207 with protocol of “TLSv1.2”. It looks like this is going to Google, Inc., and it is application data.
4. **MY COMPUTER and CETON NETWORK DEVICE HAVE CONSTANT COMMUNICATION**. I found 4707 (or 42.2% of all packets) going from my current computer to a network device that I have which records television programs. Then, I found the other bulk of packets 5828 (or 52.2% of all packets) going that network device to my computer. This is happening, even though, there is no television program being recorded.
5. **CETON NETWORK DEVICE SENDING FREQUENT NOTIFY EVENTS**. My Ceton device keeps sending a protocol of HTTP/XML with a Notify event for something constantly. Maybe, it is about to record a show. I have Windows Media Center installed on this PC, so maybe that is the reason. It is just surprising that there is this much information going back & forth.

So, I am perplexed by some of this behavior, because I would never have expected so much of this packet information would be just plain wasted communication. And, I didn’t know so much network traffic was occurring without anything really happening at the current moment.

**3.2. What are five observations about the NetWalk network packet trace when wired?**

As the NetWalk network packet trace is analyzed[[3]](#footnote-3), the findings are presented below:

1. **DIFFERENT RESULTS**. Surprisingly, the answers are different in comparison to the Wireshark analysis. As to why this is the case, I am not sure right now as I begin to analyze the data, but it is an interesting initial observation. My numbers are tilted in a totally different direction.
2. **MY COMPUTER and CETON NETWORK DEVICE STILL HAVE CONSTANT COMMUNICATION**. Of 14653 packets tracked, I found 1222 (or 8.3% of all packets) going from my current computer to a Ceton network device. Then, I found the other bulk of packets 13243 (or 90.4% of all packets) going from that network device to my computer. Thus, there are 98.72% of packets in this network trace that are consumed by network packets between my computer and the Ceton network device. Again, this is happening, even though, there is no television program being recorded.
3. **DEVICE PERFORMING SSDP NOTIFICATIONS**. I have a device at 192.168.1.200 which is communicating to destination 239.255.255.250 with a protocol of “SSDP” with a message content of “NOTIFY \* HTTP/1.1” (or something similar). To explain this, I know that the device on my network at 192.168.1.200 is a shared network drive that is housed in a NIC enclosure [an Iomega NAS] which is sending a SSDP (or Simple Service Discovery Protocol) request. Thus, I found information related to explain this on Wikipedia: "The Simple Service Discovery Protocol (SSDP) is a network protocol based on the Internet Protocol Suite for advertisement and discovery of network services and presence information. It accomplishes this without assistance of server-based configuration mechanisms, such as the Dynamic Host Configuration Protocol (DHCP), and without special static configuration of a network host. SSDP is the basis of the discovery protocol of Universal Plug and Play (UPnP) and is intended for use in residential or small office environments." And, continues with "... SSDP uses a NOTIFY HTTP method to announce the establishment or withdrawal of services (presence) information to the multicast group. A client that wishes to discover available services on a network, uses the M-SEARCH method. Responses to such search requests are sent via unicast addressing to the originating address and port number of the multicast request."[[4]](#footnote-4) So, my network drive is constantly trying to figure out what services are on the network. I don’t understand why it would do it so frequently, but another thing that I found and didn’t expect.
4. **INITIAL UNKNOWN WEBSITE INTERACTING WITH MY COMPUTER**. A website at the address of 192.241.168.226 is occasionally interacting with my computer. And, I don’t have a device at that address, so I looked up the reverse IP-address of that site. And, it turns out to be LastPass which is a password storage provider. Yet, again, I didn’t do any password requests during the time frame of the monitoring.
5. **SECOND UNKNOWN WEBSITE INTERACTING WITH MY COMPUTER**. A website at the address of 193.149.72.169 on port 443 is occasionally interacting with my computer. And, I look it up to find that it is Microsoft Limited in United Kingdom, and port 443 is HTTPS. However, I don’t remember (during this time period) doing anything with Microsoft nor needing HTTPS. So, this is another thing to research in the future.

Again, I am baffled with so many unknown source and / or destination addresses interacting with my computer without my knowledge. This is another thing that I learned from this class.

**3.3. What are five observations about the Windump network packet trace when wired?**

After analyzing the Wireshark and NetWalk network packet traces above, I feel that I can produce an analysis much quicker and better than before, which is presented again in the below utilizing my VBA code again[[5]](#footnote-5):

1. **SLOWER PERFORMANCE SO SMALLER SAMPLE SIZE**. This Windump application ran about as long as the other tools, so I thought that it got about the same size of a file because it ran for about 3 minutes or so. However, it only gathered 977kb, thus I only have 4761 total packets to analyze.
2. **CETON AND GIGABYTE COMPUTER HAVE A VAST AMOUNT OF TO/FROM PACKETS**. Of these 4761 packets, the Ceton device and my computer still have significant communications. My computer is sending 1920 packets (or 40.3% of all packets), and Ceton is sending to my computer 2475 packets (or 52% of all packets). So, the total communications between the two of them is 92.3% of these packets. That is amazing, and I would have never believed it to be possible.
3. **UNKNOWN WEBSITE #1 INTERACTING WITH MY COMPUTER**. A website at the address of 54.225.155.188 on port 443 is occasionally interacting with my computer. And, I look it up to find that it is Amazon Web Services, then I look further and find that it is [www.Piazza.com](http://www.Piazza.com) which makes sense because updates from other students were being made as I monitored my network.
4. **UNKNOWN WEBSITE #2 INTERACTING WITH MY COMPUTER**. A website at the address of 216.58.218.206 on multiple ports is occasionally interacting with my computer. And, I look it up to find that it is Google, Inc., and I sort it by Time to discover that it is using multiple protocols and changing ports frequently. So, let me just look up one that I don’t understand. It is protocol of “QUIC” which is[[6]](#footnote-6) “a multiplexed stream transport over UDP.” QUIC is “an experimental protocol aimed at reducing web latency over that of TCP. On the surface, QUIC is very similar to TCP+TLS+SPDY implemented on UDP. Because TCP is implement in operating system kernels, and middlebox firmware, making significant changes to TCP is next to impossible. However, since QUIC is built on top of UDP, it suffers from no such limitations.” So, some key features of QUIC when used over existing TCP+TLS+SPDY are: “1) Dramatically reduced connection establishment time, 2) Improved congestion control, 3) Multiplexing without head of line blocking, 4) Forward error correction, and 5) Connection migration”. So, I understand it better now.
5. **UNKNOWN WEBSITE #3 INTERACTING WITH MY COMPUTER**. A website at the address of 173.194.115.35 on unspecified ports utilizing “QUIC” protocol Again, this is Google, Inc., because it is the QUIC protocol but it doesn’t spell out what is occurring other than just a “CID: 0, Seq: 76” with a sequence from 76 to 99 occurring during this timeframe. So, I believe that I missed some of the start of this transfer because it is just sending packets in a sequence, and I have only caught some of the sequence. So, I am not sure what it is doing. To do some research, a CID is a “Context Session Identifier (CID)” is utilized in a UDP Packet to encode the information transfer to assure that it is ordered. It also seems to be designed to develop a context for a compressed packet for an end-to-end, connection-oriented communication. From multiple websites, it says that these sequence numbers help to discover out-of-order and / or lost packets.

Thus, these are my observations after using Windump.

**3.4. What are five observations about the Wireshark network packet trace when wireless?**

For these next three questions, I borrowed a laptop (because I don’t have one that is modern) to be able to do this monitoring. So, I went up to Chick-fil-A in the evening of Thursday (April 16, 2015) from about 8:30pm until 9:25pm. It was raining, but the place still had traffic and many people using their wireless communication devices. Therefore, I should have good data to analyze. I did not analyze it there, because the screen size of the laptop is not conducive for that. Thus, I waited to do the analysis on a bigger monitor.

The below is my analysis of the network packet trace when using the Wireshark application:

1. **ATTEMPTS TO CONTACT PRIOR NETWORK CONNECTIONS**. This laptop is attempting to do multiple DNS (“Standard query 0x\_\_\_\_”) queries to attempt to find computers that are on my home network like FileServer, TheSith, etc. And, it using a destination port of 198.153.192.60, which is different than on my home network.
2. **ATTEMPTING TO LOCATE PRIOR COMPUTERS**. Also, I see another protocol of “NBNS” with a “Name Query” for THESITH called out to the same subnet of 192.168.10.XXX, but the final octet of the address is 255. It looks like NBNS is the NetBIOS Name Service which is a legacy protocol, superseded by many functions of DNS. From ehow.com[[7]](#footnote-7), NBNS is a Presentation-level protocol that uses UDP/TCP port 137. This is performing NetBIOS over TCP/IP, which is an analogous predecessor of DNS. As stated, the NBNS/WINS will collect and compile information about NetBIOS names on a particular network, so it is collating the network information where it can cross-reference it against the existing workstations. So, I am assuming that, after waking up from hibernation, it is trying to reconnect with the prior network devices.
3. **OTHER DEVICES ON THIS NETWORK**. I can see many more IPaddresses like 192.168.10.16 (which is someone else), because the laptop is 192.168.10.26 on this network. So, I am seeing a lot of other traffic on this network that I do not recognize who it is. So, it is indeed picking up network traffic over the wireless adapter.
4. **EXCHANGING CERTIFICATES AND CHANGING CIPHERS**. My wife’s laptop is interacting with the servers at 15.217.50.50, which is Hewlett Packard’s company. So, after losing contact from her work computers at home, the laptop is attempting to get another secure connection on the restaurant’s network where it can exchange application data in a secure fashion. These interactions occur for quite some time as attempts are made to setup the protected connection.
5. **MULTIPLE COMMUNICATIONS WITH 108.160.166.138**. After analysis of why there are so much traffic between this laptop and the 108.160.166.138 address, I find out that the address is Dropbox where a lot of her immediate files that were being used are backed-up. So, the laptop is probably trying to backup files, which it is instructed to do every evening.

Thus, the above observations are what I can immediately notice when analyzing the Wireshark network trace using the wireless interface.

**3.5. What are five observations about the NetWalk network packet trace when wireless?**

The below is my analysis of the network packet trace when using the NetWalk application:

1. **DROPBOX HAS MULTIPLE IP-ADDRESS**. In the prior analysis, I found that Dropbox is using 108.160.166.138. However, I see many interactions with IP addresses that are similar, such as: 108.160.166.138, 108.160.167.145, 108.160.169.183, and 108.160.173.162. So, maybe, Dropbox is doing load-balancing or has some information stored on one server, but then other information is on another server. So, when the laptop is requesting certain information or making a particular backup, then a main Dropbox server is directing traffic to the exact server. Looks like anything that has the octet of 108.160.XXX.XXX is Dropbox, because it is in the packet trace many, many times.
2. **ANOTHER DEVICE IS ASKING FOR CLIENTS ON THE NETWORK**. After learning on a prior analysis, I see another device (192.168.10.16) sending a SSDP protocol request to destination (239.255.255.250). I remember from the prior reading that this is a Simple Service Discovery Protocol (SSDP) that is trying to see who is on the current subnet.
3. **LAPTOP WANTS TO DO A NAME QUERY REQUEST FOR PRIOR CONNECTIONS**. The laptop tries to do a 192.168.10.255 request for all of these prior connections: FileServer, HQ\_NT, TheSith, etc. So, I see that the laptop then sends a DNS request to 198.153.192.60, asking about each of these (or more than that) domains: FILESERVER.aquent.com, TheSith.aquent.com, client.dropbox.com, block.dropbox.com, d1ihw0800yew8.cloudfront.net, etc. It looks like the laptop is checking these domains with 198.153.192.60 who is Norton DNS name server. So, the laptop has Norton installed and is trying to see that everything is safe before it accesses these website and / or domains.
4. **LAPTOP TALKING WITH 54.200.75.96 MULTIPLE TIMES**. In the tracing history, I can see that the laptop is interacting with 54.200.75.96 countless times. It looks like these are non-default DNS servers for this network, so the filtered servers were being used by Norton, which were returning the IP address of Amazon-AWS servers for any DNS request that is not getting resolved. And, this is a documented problem[[8]](#footnote-8) that has occurred on the web before.
5. **LAPTOP COMMUNICATES FREQUENTLY WITH 54.230.6.15**. Studying this, I find that 54.230.6.15 is Amazon.com or Amazon Technologies. I am not sure why this laptop would need to talk to Amazon, but it could be that Amazon Technologies is hosting some type of application that this laptop is using for her work applications.

Thus, the above observations are what I can immediately notice when analyzing the NetWalk network trace using the wireless interface.

**3.6. What are five observations about the Windump network packet trace when wireless?**

Before this analysis begins, I am not sure that Windump worked correctly on the laptop. Everything else seemed to work well after I figured out the details. However, this Windump prepared a dump of communications much too fast to seemingly be a wireless connection. However, the laptop was not connected through a Gigabit NIC-link nor any type of wired connection. So, I do not have enough experience to exactly provide a good explanation for this behavior.

The below is my analysis of the network packet trace[[9]](#footnote-9) when using the Windump application:

1. **LAPTOP PREDOMINATELY COMMUNICATES WITH 54.230.6.15**. Again, I looked this IP address up to find that it is Amazon. So, I am not sure why the laptop would need to communicate with Amazon so many times. There are 255 of 1009 packets from Laptop to Amazon (or 25.3%), and there are 466 of 1009 packets from Amazon to laptop (46.2%). Or, there are 71.5% of packets used primarily for this purpose which I do not understand why it would be necessary, because the laptop was not utilizing an Amazon webpage.
2. **LAPTOP COMMUNICATING EXTENSIVELY WITH DROPBOX**. For a numerical analysis, the laptop is communicating with Dropbox (i.e., IP address of 108.160.173.162) for a significant number of packets. Of the total of 1009 packets measured, packets from laptop to Dropbox is 108 of 1009 (or 10.7%), and packets from Dropbox to laptop is 170 of 1009 (or 16.9%). So, roughly 27.6% of packets are devoted directly to communicating with Dropbox.
3. **TOTALITY OF PACKETS IS FOR AMAZON OR DROPBOX**. When using Windump, I found that 99.1% of the packets are being used to talk with Amazon servers (71.5%) and / or Dropbox servers (27.6%). So, there are only 10 packets out of 1009 packets that are not talking to either of those services.
4. **ONE PACKET IS TALKING WITH 74.125.227.175**. This is not identified explicitly on the web as being identified, but it is suggested that it is Google. So, one packet out of 1009 is communicating with Google.
5. **REMAINING PACKETS ARE UNCLEAR**. There are three packets with a source of Fortinet\_b3:c6:01 and a destination of IntelCor\_c7:bb:3c, whereas three more packets have these IP addresses switched. But, I searched, and I cannot find anything that explains what this is.

Thus, I refine what I said above, because it appears that Windump was working correctly and did actually perform as it should have on the laptop. But, with such a small screen, I could not test nor verify it before the restaurant closed and before my chicken sandwich & waffle fries got cold.

**4.0. Provide a further analysis of a few IP address from the hosts found earlier.**

Now, I will continue the analysis for this homework by analyzing some of the above IP address in more detail by providing the following information on each:

1. ICMP ping reachability and latency,
2. nmap port scan (use your own computer or your friends computer as target IP) ,
3. nmap TCP/IP fingerprints and IP ID sequence,
4. reverse DNS, and
5. traceroute.

So, I have never really done the above in such detail, so I will do my best to fulfill this requirement. So, I will select some external addresses that I am curious about which are listed below:

|  |  |  |
| --- | --- | --- |
| **(#)** | **IP Address:** | **Destination:** |
| (1) | 108.160.166.138 | Believed to be Dropbox. |
| (2) | 54.230.6.15 | Believed to be Amazon. |
| (3) | 15.217.50.50 | Believed to be Hewlett Packard’s company |

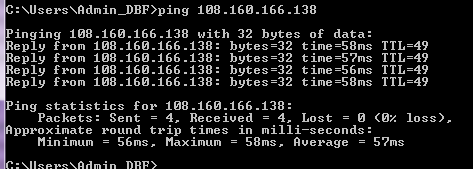
However, this analysis takes far too long per IP address, so I will only do two of them. I want to do Dropbox and my wife’s corporate sign-in site of Hewlett Packard’s VPN connection (I believe).

**4.1. Analysis of 108.160.166.138 which should be Dropbox.**

In this section, I am performing the following analyzes on the given IP address and am providing the following information on each:

1. ICMP ping reachability and latency,
2. nmap port scan[[10]](#footnote-10) (use your own computer or your friends computer as target IP) ,
3. nmap TCP/IP fingerprints and IP ID sequence,
4. reverse DNS, and
5. traceroute[[11]](#footnote-11).

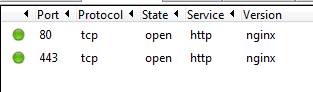
[4.1a] The ping command has the following screenshot:



This screenshot is communicating that it is latency between my computer and this IP address is between 56ms and 58ms (or 0.058s latency), whereas the reachability is possible and is not restricted.

[4.1b] Having never had to do a port scan through a nmap application, I really wished that I would have known about this years ago. However, I learned how to quickly use Zenmap in Windows 7. It is an awesome port scanner. All I had to do was enter a target IP address, and it provided a lot of detail.

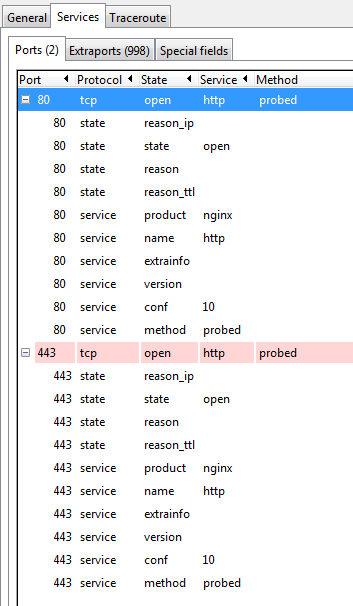
These are the results of the port scan (I believe):



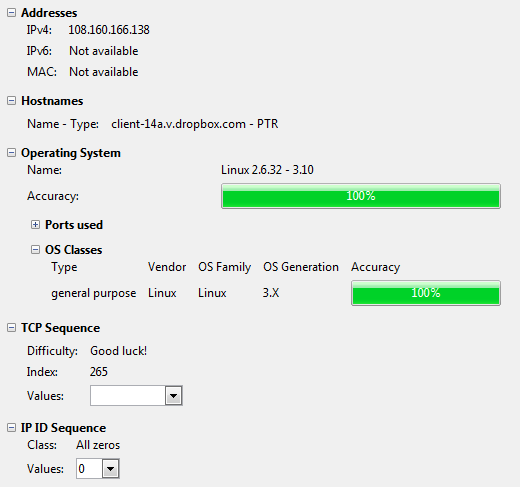
But, I do believe that the above result is not thorough enough, so this is how it performed it after looking at 1000 ports:

C:\__Configs\CONFIGS -- SnagIt Screenshots\IMAGE -- snagit #0003 (4-18-2015 @ 3.18.00 AM) on GIGABYTE.png

The below is a more thorough analysis of the ports:

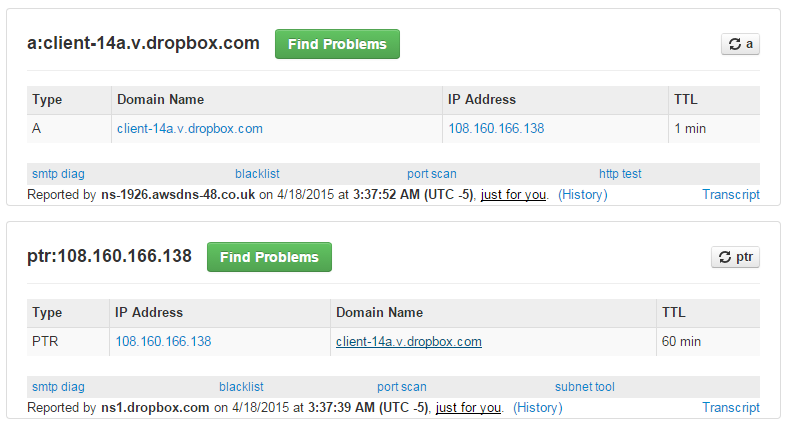


[4.1c] To provide the nmap TCP/IP fingerprints and IP ID sequence, I am not sure that I understand what is being requested here. If the below is not what is wanted, then I do not know enough to understand what you want.

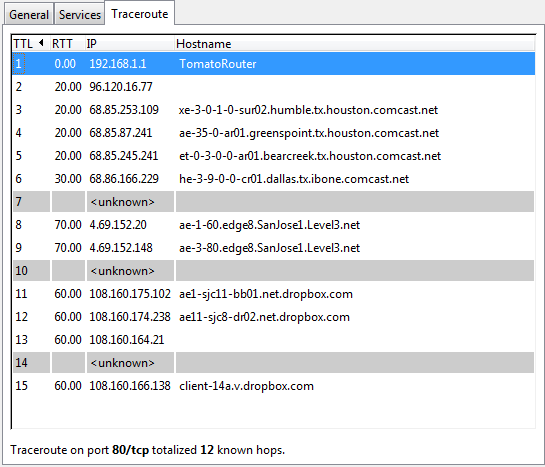


But, I do not understand what I am expecting to see at the bottom of this screenshot. I see TCP Sequence and IP-ID Sequence but there is no values that the combo boxes will take.

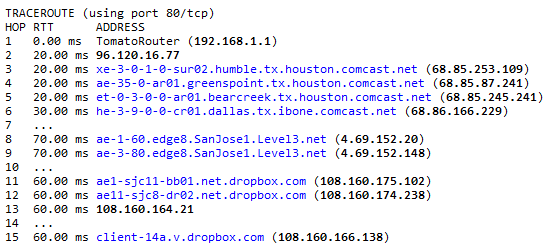
[4.1d] The reverse DNS lookup is given below:



[4.1e] The traceroute was given by Zenmap so it is provided below:



But, the text printout is given below which is not as organized but I have seen before.

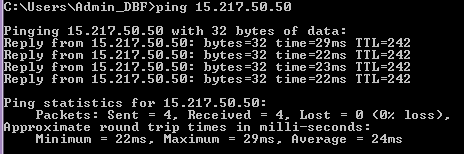


**4.2. Analysis of 15.217.50.50 which should be Hewlett Packard’s company.**

In this section, I am performing the following analyzes on the given IP address and am providing the following information on each:

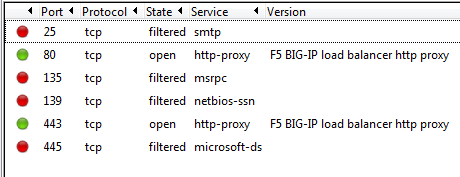
1. ICMP ping reachability and latency,
2. nmap port scan (use your own computer or your friends computer as target IP) ,
3. nmap TCP/IP fingerprints and IP ID sequence,
4. reverse DNS, and
5. traceroute.

[4.2a] The ping command has the following screenshot:



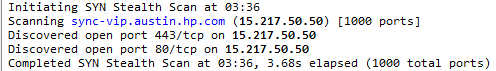
This screenshot is communicating that it is latency between my computer and this IP address is between 22ms and 29ms (or 0.029s latency), whereas the reachability is possible and is not restricted.

[4.2b] Now, using Zenmap, I scanned the ports to get the below ports that are open or used:

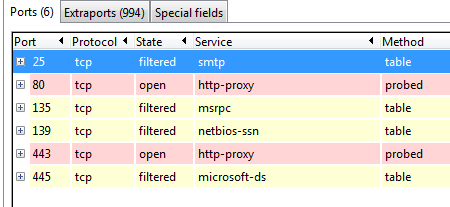


The above looks different than the prior analysis, so I can see that there are open ports and filtered ports.

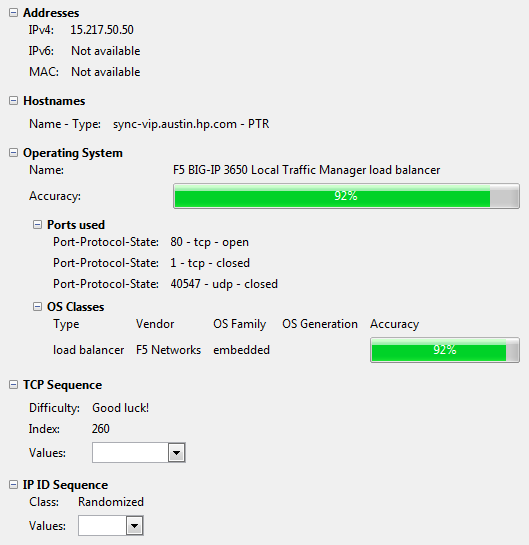
But, I do believe that the above result is not thorough enough, so this is how it performed it after looking at 1000 ports:



The below is a more thorough analysis of the ports:

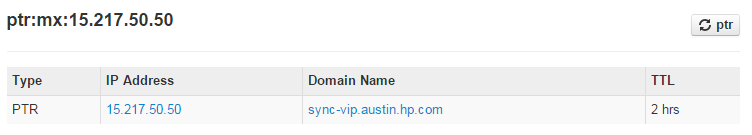


[4.2c] To provide the nmap TCP/IP fingerprints and IP ID sequence, I am not sure that I understand what is being requested here. If the below is not what is wanted, then I do not know enough to understand what you want.

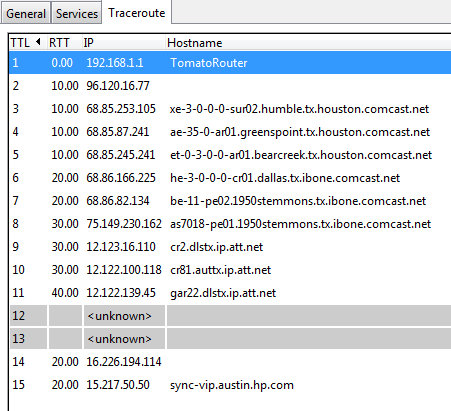


But, I do not understand what I am expecting to see at the bottom of this screenshot. I see TCP Sequence and IP-ID Sequence but there is no values that the combo boxes will take.

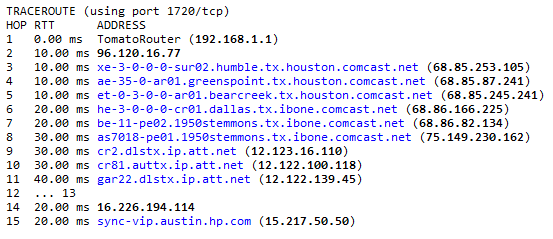
[4.2d] The reverse DNS lookup is given below:



[4.2e] The traceroute was given by Zenmap so it is provided below:



But, the text printout is given below which is not as organized but I have seen before.



**Summary:**

This assignment really caused me to enter deeper into a domain that I don’t really have an extreme amount of the experience. So, it took me much longer than it should have to perform some of the tasks, but upon repeating the tasks, I was much faster and understood what to do. Throughout this assignment, I had to install many applications, configure their settings, learn how to work the interfaces, and research on the web to understand what I am supposed to do. Therefore, I hope it is understood that I worked hard and dedicated myself to this task.

Truly, I wish that I would have understood more of this years ago, because it has always been intimidating to me and not something that I would readily accept new tasks in this domain without extensive training. However, I feel more comfortable with the above tasks. I do not yet understand what all is meant by the above, but I know how to use some of the tools, know some of the buzzwords, and can do basic analysis when given problems from the domain.

1. On the website, Downloads.CNet.com, is where I looked for the most popular downloads for network monitoring and packet tracing analysis tools. [↑](#footnote-ref-1)
2. The VBA code used is Analyze\_Sheet\_1. [↑](#footnote-ref-2)
3. The VBA code used is Analyze\_Sheet\_2. [↑](#footnote-ref-3)
4. This is compiled from the material at this location:

   <http://en.wikipedia.org/wiki/Simple_Service_Discovery_Protocol> [↑](#footnote-ref-4)
5. The VBA code used is Analyze\_Sheet\_3. [↑](#footnote-ref-5)
6. I utilized information from this web source:

   <https://www.chromium.org/quic> [↑](#footnote-ref-6)
7. The information is used from this website: <http://www.ehow.com/list_7655707_nbns-protocols.html> [↑](#footnote-ref-7)
8. This Kaspersky.com information provided me insight into this:

   <http://forum.kaspersky.com/lofiversion/index.php/t305010.html> [↑](#footnote-ref-8)
9. The VBA code used is Analyze\_Sheet\_6a and Analyze\_Sheet\_6b. [↑](#footnote-ref-9)
10. Windows 7 does not have a native “nmap” command. So, I had to research this, and I found this website to download a version to use: <http://nmap.org/book/inst-windows.html>. However, that application gave too much trouble and was for an earlier edition of Windows, so I had to find another application which is listed as: http://filehippo.com/download\_nmap/ [↑](#footnote-ref-10)
11. Windows does not have a native “traceroute” command. So, I will do the following: FDJKFJSDJFKDSJLKFDSLJK [↑](#footnote-ref-11)