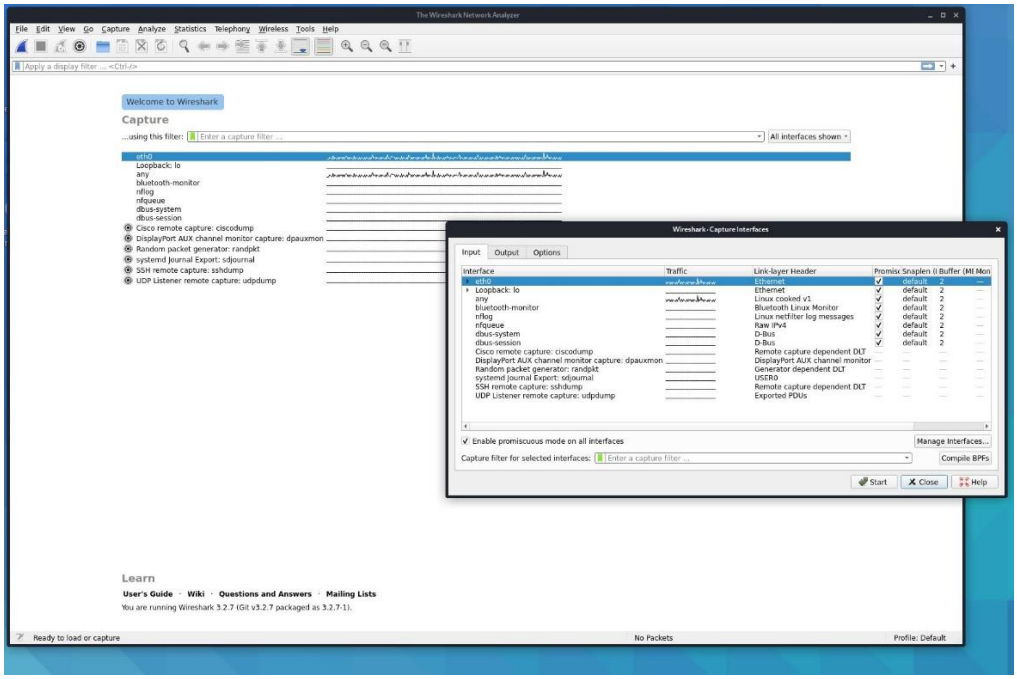
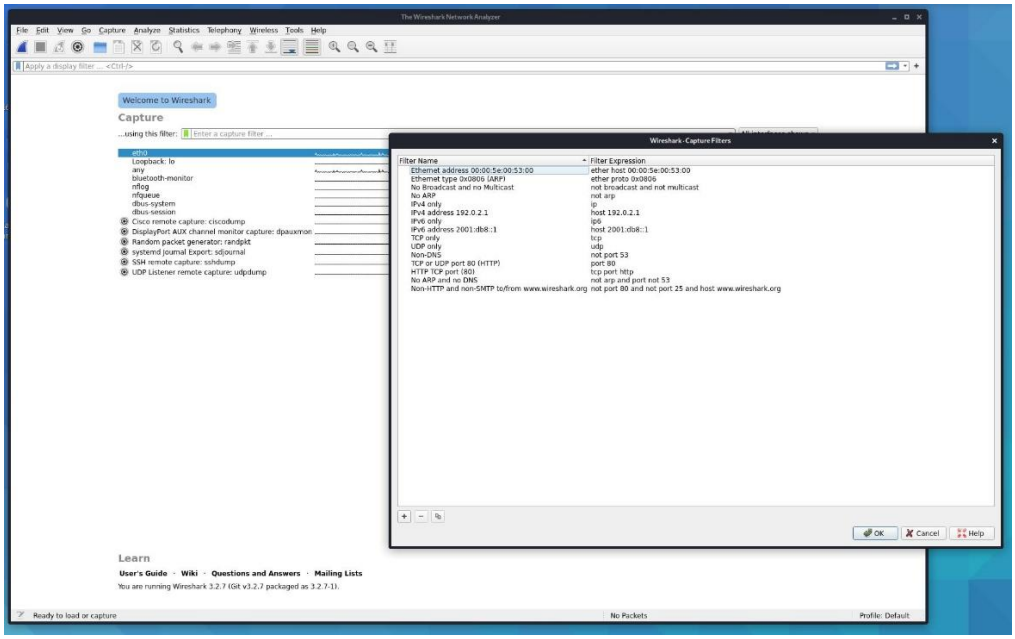
The CEO told me that they wanted to test me to again prove the value of PEN testing to the company. This next test was to demonstrate the aspects of network sniffing. I was again asked to give a bit of a walk through on the process I took.

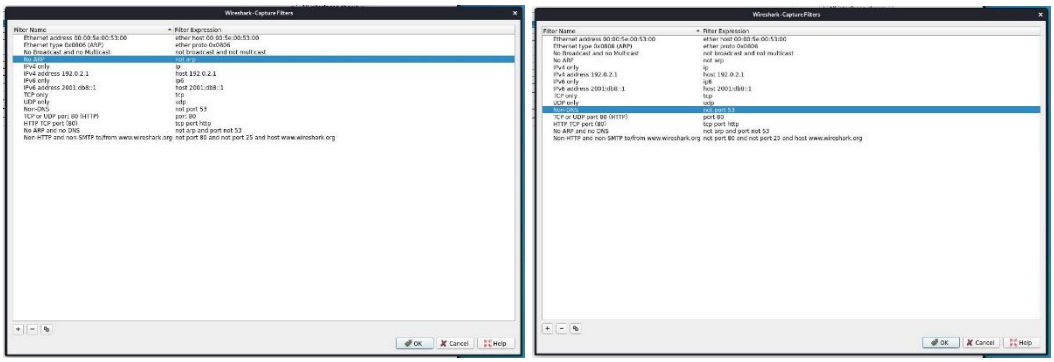
1. After loading into my testing platform, I started up the program Wireshark and navigated to the **Capture** menu and selected the submenu **Interfaces**. This is done to confirm the different interfaces that Wireshark has detected.



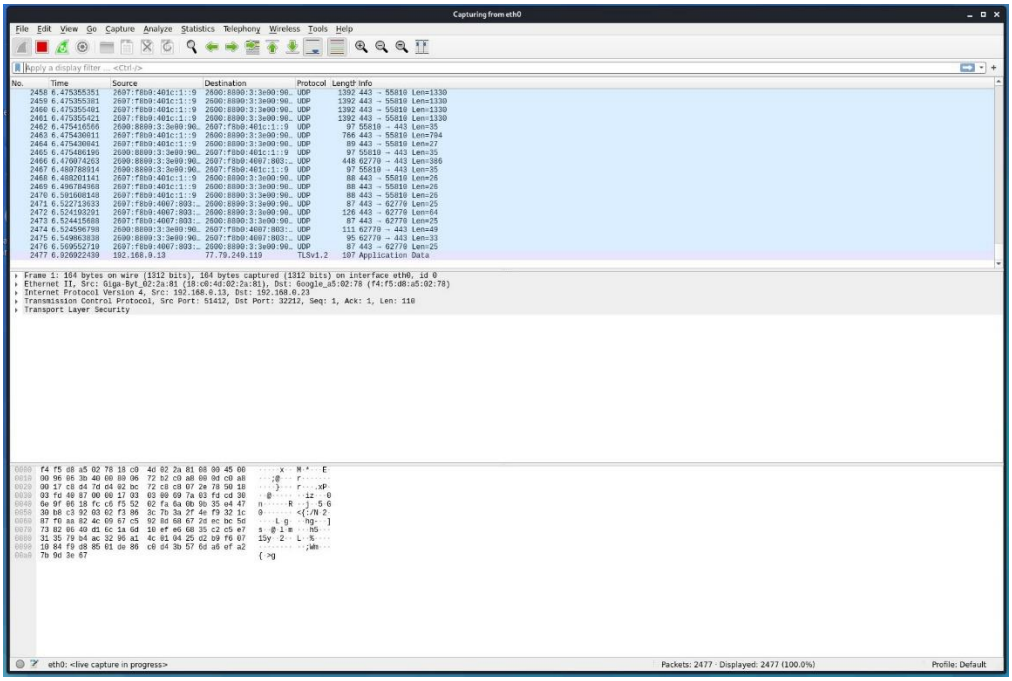
2. As eth0 is our currently selected and active network port, I exit out of the **Interfaces** menu and go back to **Capture** and selected the submenu **Options**.
3. I then click on the *Capture Filter* button and open that dialog box. This lists several different capture filters already configured in Wireshark.



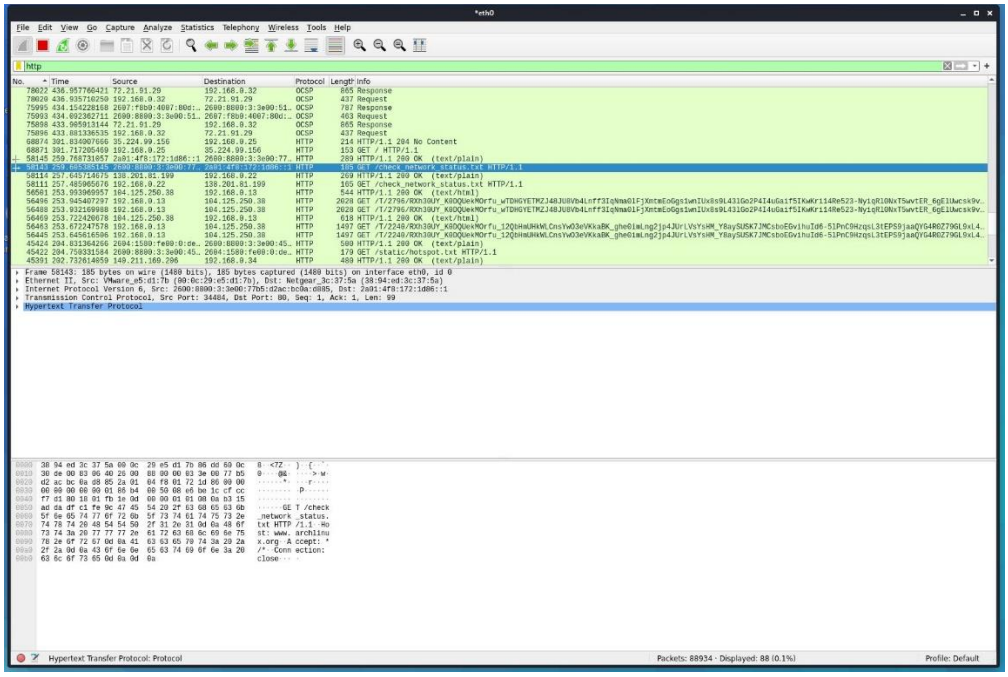
4. Such as the filters for *No ARP* and *Non-DNS*.



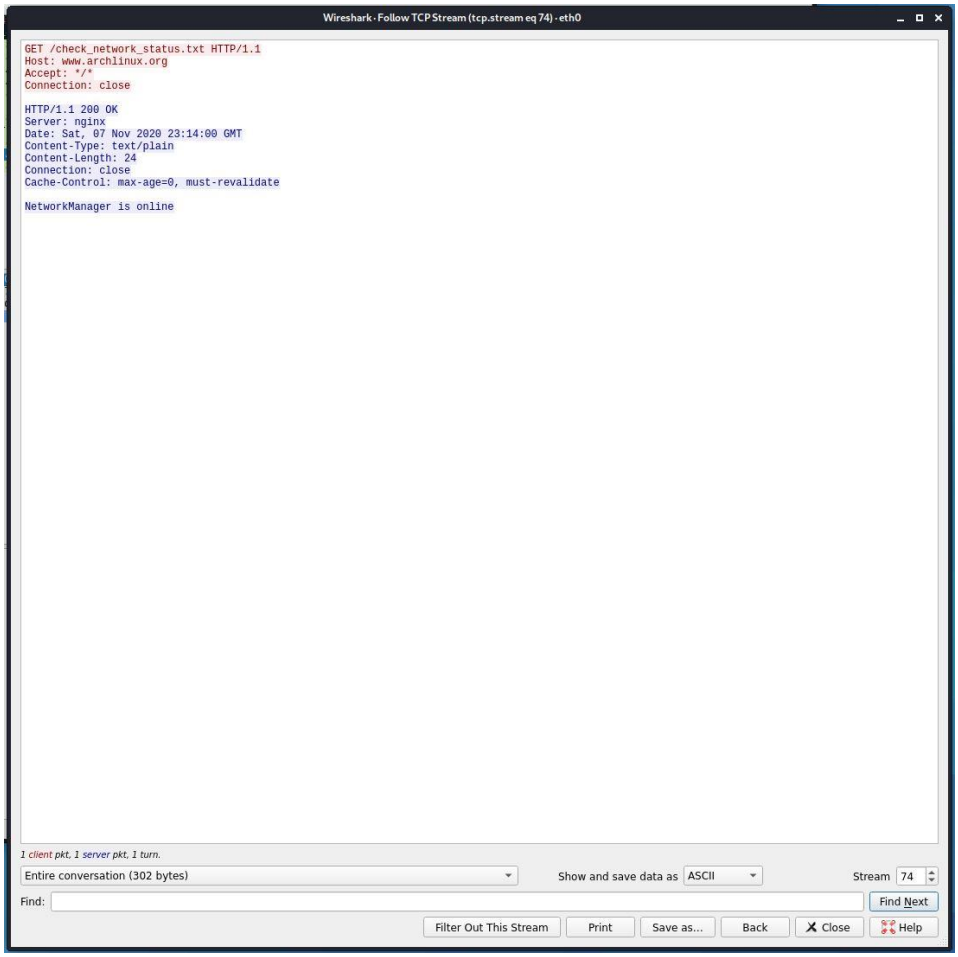
5. I then exit this dialog box and return to the main window. From there I click on the blue finned Start Capture button to begin capturing traffic on this test platform. I visit a few websites to generate traffic.
6. I begin looking for TCP protocol-based packets but I'm initially finding only UDP packets.



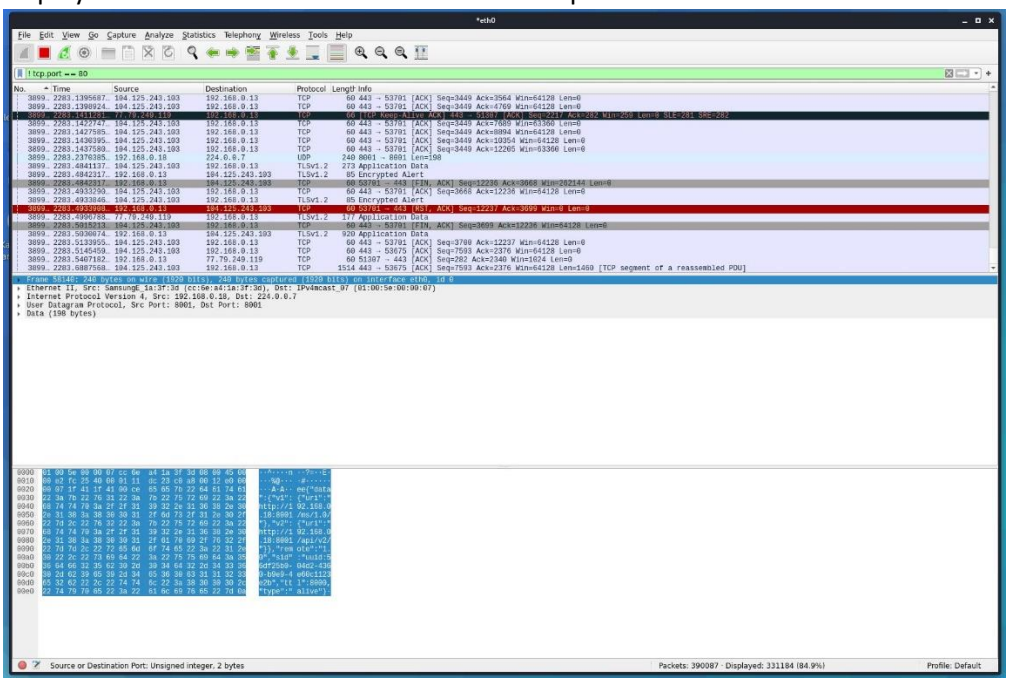
7. I then start looking through the traffic to find where my browser made a request to one of the web sites I visited. There was so much traffic that I had to filter it for HTTP.



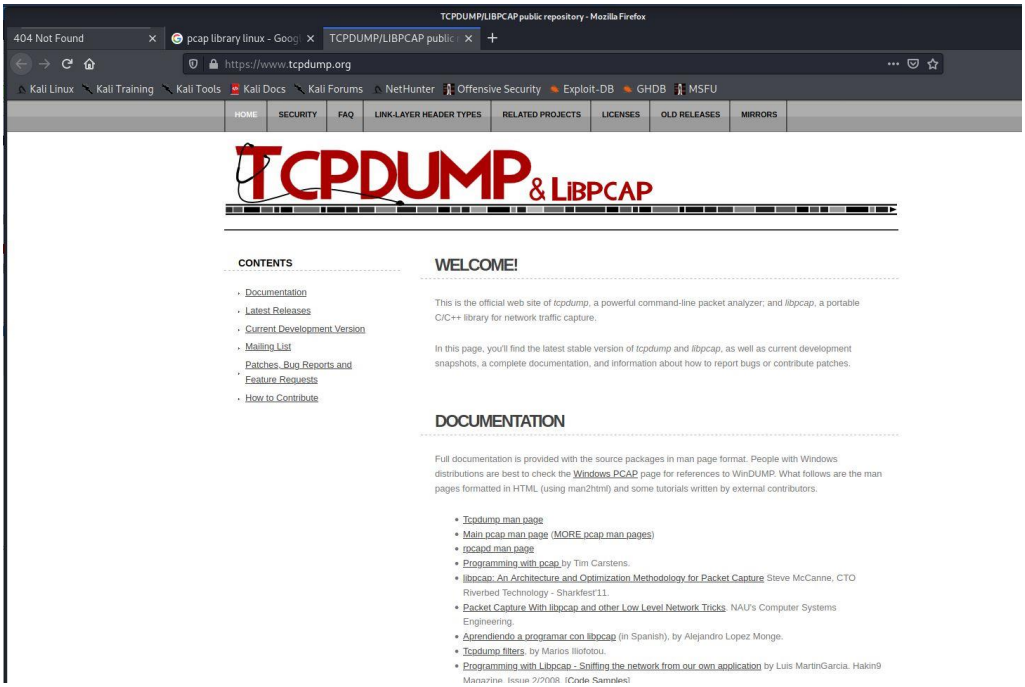
8. Selecting one of the HTTP packets I then go to the **Analyze** menu, the submenu **Follow**, and select the option **TCP Stream**.



9. After closing the *Follow TCP Stream* window, I cleared the *Filter Menu Bar* and used a display filter that would filter out TCP traffic on port 80.

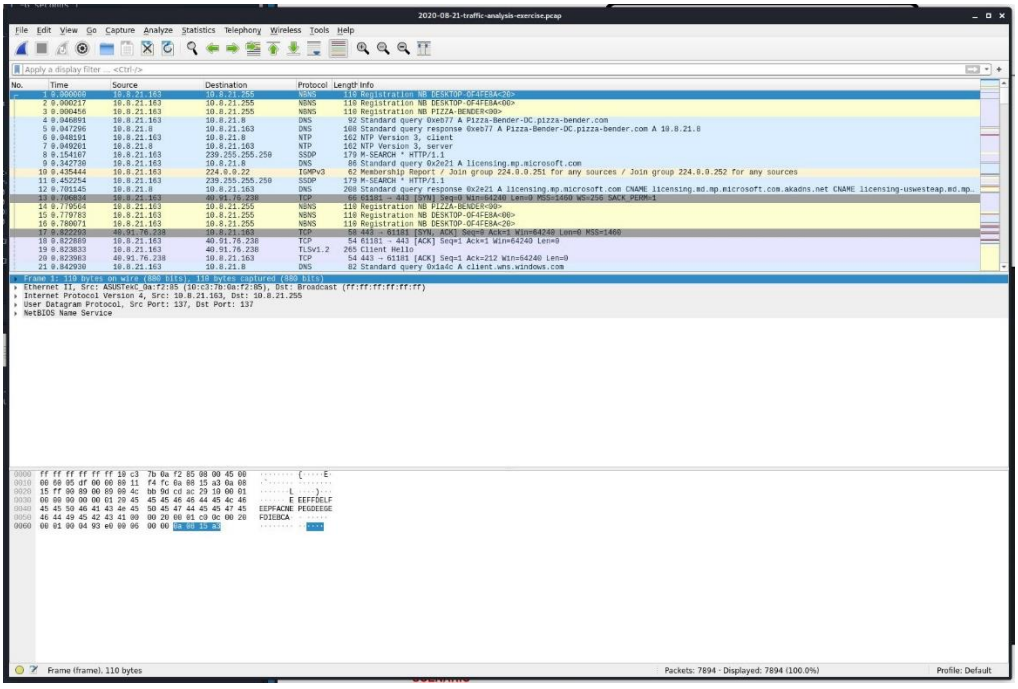


I was then tasked by the CEO to go to www.pcapr.net and demonstrate PCAPs. Unfortunately, the website in question no longer seems to be up, would get a 404 error when trying to go to that site. I did do a Google search of pcap library linux and found the site www.tcpdump.org.

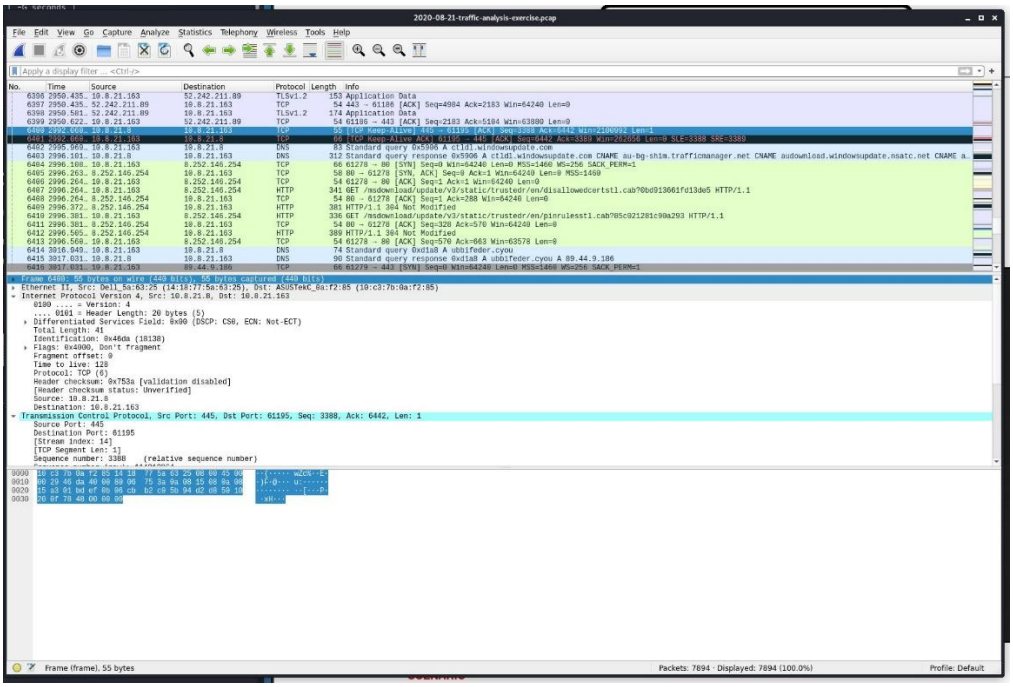


After reading through the site and checking my Kali VM, I discovered that Kali Linux is already loaded with the current versions of both Tcpdump and Libpcap.

I then looked up a basic trojan malware packet capture examples. I downloaded one and opened it in Wireshark and began to analyze it.



I found that the TCP and HTTP packets that indicated the site location from where I believe the trojan was picked up, in this example.



Logs

11/06/20 15:32 - Powered on Kali VM & logged in

11/06/20 15:36 - Powered on Manjaro VM & logged in

11/06/20 15:39 - Powered on Ubuntu VM & logged in

11/06/20 15:43 - Powered on Fedora VM & logged in

11/06/20 15:46 - Powered on Windows 10 VM & logged in

11/06/20 15:50 - Start up Wireshark in Kali VM

11/06/20 15:55 - In Wireshark, navigated to **Capture** menu, submenu **Interfaces**

11/06/20 15:58 - In Wireshark, exited **Interfaces** submenu and navigated to **Options** submenu, still under **Capture** menu

11/06/20 16:01 - In Wireshark, from **Options**, selected *Capture Filter*

11/06/20 16:05 - In Wireshark, checked the capture filters preconfigured on Wireshark

11/06/20 16:06 - In Wireshark, exited *Capture Filter*

11/06/20 16:07 - In Wireshark, began capturing traffic on eth0 of Kali VM

11/06/20 16:10 - In web browser, visited random sites and did a Google search to generate traffic

11/06/20 16:15 - In Wireshark, checked traffic for TCP protocol packets

11/06/20 16:19 - In Wireshark, filtered traffic for HTTP

11/06/20 16:22 - In Wireshark, selected an HTTP packet, went to **Analyze** menu, submenu **Follow**, selected option **TCP Stream**

11/06/20 16:25 - In Wireshark, exited *Follow TCP Stream* window

11/06/20 16:26 - In Wireshark, cleared *Filter Menu Bar*

11/06/20 16:30 - In Wireshark, used display filter ***! tcp.port == 80*** to filter out TCP traffic on port 80

11/06/20 16:33 - In web browser, navigated to www.pcapr.net, got 404 error

11/06/20 16:37 - In web browser, ran Google search for pcap library linux

11/06/20 16:39 - In web browser, navigated to www.tcpdump.org

11/06/20 16:42 - In web browser, read FAQ on www.tcpdump.org

11/06/20 16:57 - Opened terminal

11/06/20 16:58 - In terminal, ran ***sudo tcpdump -h*** command

11/06/20 17:00 - In web browser, ran Google search for trojan malware packet capture examples

11/06/20 17:07 - In web browser, navigated to www.malware-traffic-analysis.net

11/06/20 17:16 - In web browser, downloaded malware packet capture example from 08/21/2020

11/06/20 17:18 - In Wireshark, opened malware packet capture example

11/06/20 17:19 - Began analysis of downloaded malware packet capture example

11/06/20 17:31 - Found, what I believed to be site location info from where trojan may have been acquired

11/06/20 17:33 - Closed web browser

11/06/20 17:34 - Closed terminal

11/06/20 17:35 - Powered down Kali VM

11/06/20 17:37 - Powered down Manjaro VM

11/06/20 17:39 - Powered down Ubuntu VM

11/06/20 17:41 - Powered down Fedora VM

11/06/20 17:43 - Powered down Windows 10