# Introduction - Alexis Xavier 33791124

Warning:  This experiment may contain personal details on my own home network

Within this topic, I wanted to conduct my own personal experiments in order to gain a greater understanding of Networking as well as expand on my understanding of cybersecurity overall. This module presented me with an opportunity to investigate further into this. Mainly used two primary tools: nmap and Wireshark.

Nmap - Nmap is commonly known as a network scanner and it is short for network mapper. It is essentially used to discover hosts and services on a computer network through sending packets and analyzing the responses. It is a tool that can also scan IP addresses and ports in a network as well to help detect if certain applications have been installed.

Wireshark - Wireshark is most commonly known to be a network packet analyzer. It shows the user captured packet data, and tries to show as much detail as possible. It helps understand more about what is going on inside a network

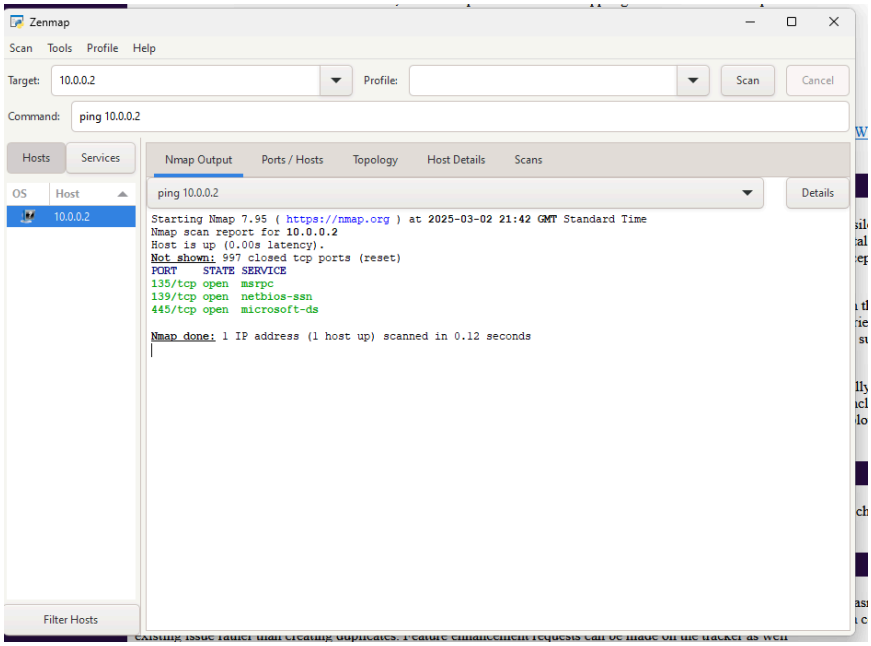
## Summary of Experiment

During this experiment, my goal was to be able to learn more about my own personal home network as well as to help expand my understanding of these tools seeing how they are crucial for later use in the cybersecurity field. They are essential for carrying out penetration testing and therefore I felt that it was important to understand more about.
Essentially, I installed both wireshark and nmap on my own home PC too and analyzed what was going on. I searched up some tutorials on YouTube in order to help get me started in understanding more about the software. At some point, I also wanted to expand this using a more popular network that was not my own, however I felt that it may have been a breach of legislation and therefore I chose not to so the findings of hosts on my networks are limited.

Alongside with this, I also wanted to explore my own router and found out there to be some really interesting functionality in which my router provided.

### Phase 1

My first intention was to be able to successfully install nmap and use its application. I also have nmap installed on Kali Linux however since it was contained within a Virtual Machine, I felt like I wouldnt get the readings that I intended to gain.




When I first opened Nmap, I was presented with an application called ZenMap, at first i was confused, however I later came to know that this is a more visual application with Nmap to help reduce the complexity that a lot of new users may encounter.

Nmap Output | Ports / Hosts | Topology | Host Details | Scans

nmap -p 1-65535 -T4 -A -v 10.0.0.2

```
Initiating SYN Stealth Scan at 21:52
Scanning 10.0.0.2 [65535 ports]
Discovered open port 135/tcp on 10.0.0.2
Discovered open port 445/tcp on 10.0.0.2
Discovered open port 139/tcp on 10.0.0.2
Discovered open port 9012/tcp on 10.0.0.2
Discovered open port 49665/tcp on 10.0.0.2
Discovered open port 49667/tcp on 10.0.0.2
Discovered open port 61774/tcp on 10.0.0.2
Discovered open port 49671/tcp on 10.0.0.2
Discovered open port 49666/tcp on 10.0.0.2
Discovered open port 49664/tcp on 10.0.0.2
Discovered open port 57621/tcp on 10.0.0.2
Discovered open port 49668/tcp on 10.0.0.2
Discovered open port 7680/tcp on 10.0.0.2
Discovered open port 9013/tcp on 10.0.0.2
Discovered open port 5040/tcp on 10.0.0.2
Completed SYN Stealth Scan at 21:52, 3.03s elapsed (65535 total ports)
Initiating Service scan at 21:52
Scanning 15 services on 10.0.0.2
Service scan Timing: About 46.67% done; ETC: 21:53 (0:00:39 remaining)
Completed Service scan at 21:53, 83.64s elapsed (15 services on 1 host)
Initiating OS detection (try #1) against 10.0.0.2
NSE: Script scanning 10.0.0.2.
Initiating NSE at 21:53
Completed NSE at 21:53, 14.20s elapsed
Initiating NSE at 21:53
Completed NSE at 21:53, 1.16s elapsed
Initiating NSE at 21:53
Completed NSE at 21:53, 0.00s elapsed
Nmap scan report for 10.0.0.2
Host is up (0.00057s latency).
Not shown: 65519 closed tcp ports (reset)
PORT      STATE    SERVICE       VERSION
135/tcp   open     msrpc         Microsoft Windows RPC
137/tcp   filtered netbios-ns
139/tcp   open     netbios-ssn   Microsoft Windows netbios-ssn
445/tcp   open     microsoft-ds?
5040/tcp  open     unknown
7680/tcp  open     pando-pub?
9012/tcp  open     ssl/websocket WebSocket++ 0.8.2
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=Peter Thorson/organizationName=Zaphoyd Studios/stateOrProvinceName=Illinois/countryName=US
| Issuer: commonName=Peter Thorson/organizationName=Zaphoyd Studios/stateOrProvinceName=Illinois/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2011-11-15T21:20:06
```

The command: nmap -p 1-65535 -T4 -A -v 10.0.0.2

This was from the tutorials in which i watched and this command essentially scans all 65,535 TCP ports i.e Full port Scan.
-T4 means that it was level 4 aggressive, and that it would try to be a faster scan, however it also meant that this scan was detectable.
-A meant that it enabled aggressive scan options such as OS detection, version detection, script scanning and traceroute.
- v = Provides a more detailed output in real time.
10.0.0.2 = Target IP address.

**Findings showed that certain ports were open and accessible.**

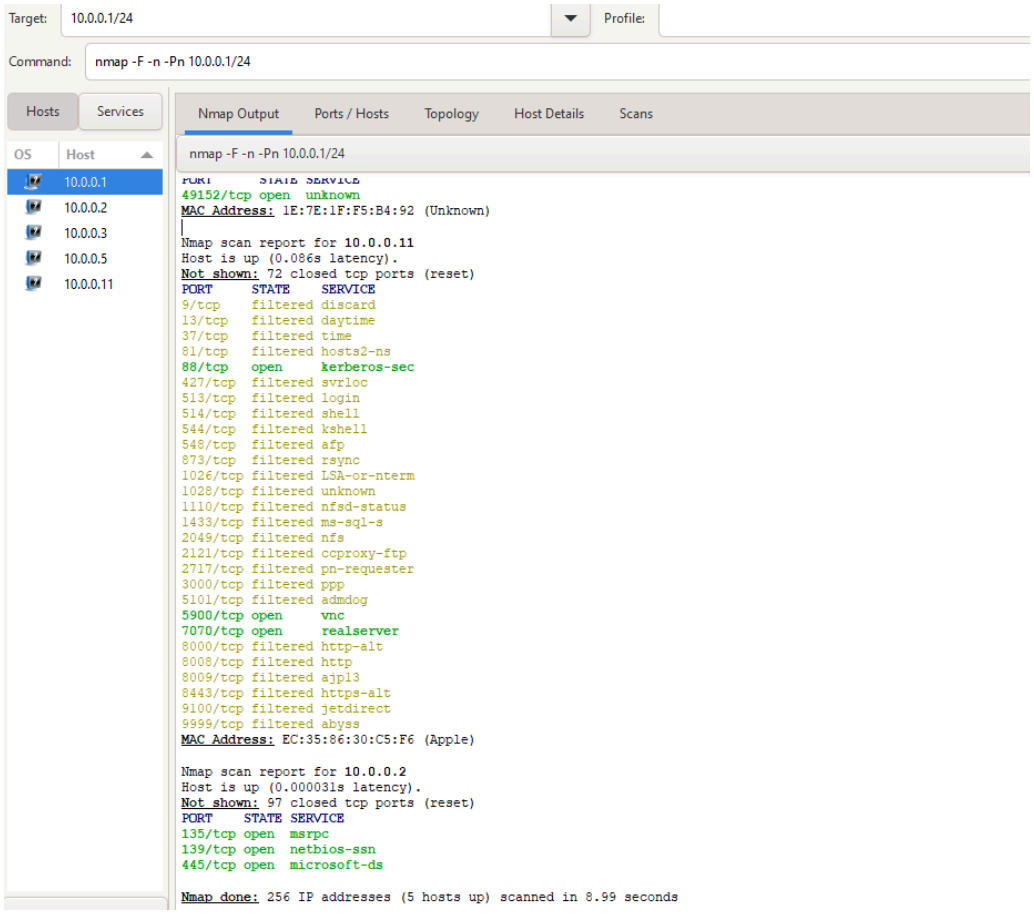
| Port | Service | Description |
|---|---|---|
| 135/tcp | msprc | Microsoft RPC service — typical on Windows machines. |
| 139 | netbios-ssn | Used for Windows file and printer sharing (NetBIOS Session Service). |
| 445 | microsoft-ds | SMB file sharing over TCP. Often used in Windows networks. |
| 9012 | pando-pub? | Custom/ Unknown service |
| 9013 | websocket | Running websocket +++ version 0.8.2. This tells us that 9013 is using SSL/TLS encryption. Self-signed certificate |

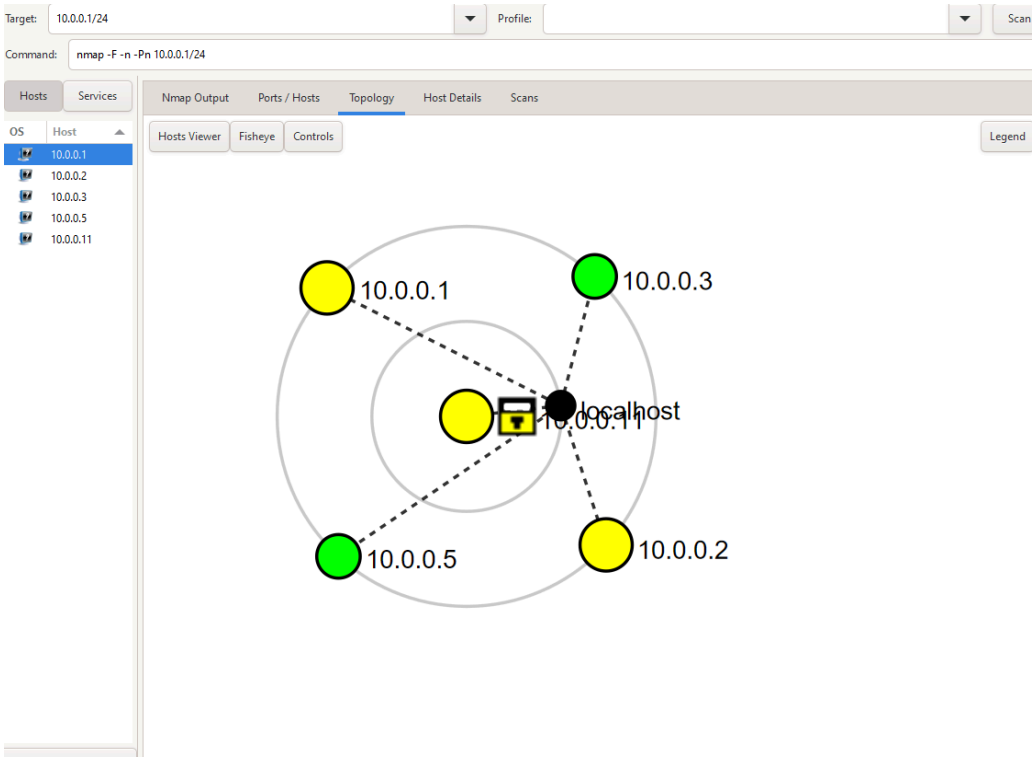
For 137, there is a firewall blocking this port since nmap did not receive a response.

```
 Public Key type: rsa
 Public Key bits: 2048
 Signature Algorithm: sha1WithRSAEncryption
 Not valid before: 2011-11-15T21:20:06
 Not valid after:  2012-11-14T21:20:06
 MD5:   fe49:23f5:2b50:6c58:5d19:0c0d:0176:daa7
_SHA-1: 8c44:2ddf:d594:5b22:204c:17d0:375a:a8ef:6956:ce8d
013/tcp  open     websocket     WebSocket++ 0.8.2
9664/tcp open     msrpc         Microsoft Windows RPC
9665/tcp open     msrpc         Microsoft Windows RPC
9666/tcp open     msrpc         Microsoft Windows RPC
9667/tcp open     msrpc         Microsoft Windows RPC
9668/tcp open     msrpc         Microsoft Windows RPC
9671/tcp open     msrpc         Microsoft Windows RPC
7621/tcp open     unknown
1774/tcp open     unknown
 fingerprint-strings:
   NULL:
_    {"type":"Tier1","version":"1.0"}
 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint at https://nmap.org/cgi-bi
ubmit.cgi?new-service :
F-Port61774-TCP:V=7.95%I=7%D=3/2%Time=67C4D308%P=i686-pc-windows-windows%
Fir(NULL,22,"{\"type\":\"Tier1\",\"version\":\"1\.0\"}\r\n");
evice type: general purpose
unning: Microsoft Windows 10|11
S CPE: cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows_11
S details: Microsoft Windows 10 1607 - 11 23H2
ptime guess: 0.327 days (since Sun Mar  2 14:03:18 2025)
etwork Distance: 0 hops
CP Sequence Prediction: Difficulty=249 (Good luck!)
P ID Sequence Generation: Incremental
ervice Info: OS: Windows; CPE: cpe:/o:microsoft:windows

ost script results:
 smb2-time:
   date: 2025-03-02T21:53:34
_  start_date: N/A
 smb2-security-mode:
   3:1:1:
_    Message signing enabled but not required

SE: Script Post-scanning.
nitiating NSE at 21:53
ompleted NSE at 21:53, 0.00s elapsed
nitiating NSE at 21:53
ompleted NSE at 21:53, 0.00s elapsed
nitiating NSE at 21:53
ompleted NSE at 21:53, 0.00s elapsed
```

This provides a tail end of the Nmap scan with some deeper system level information.

I then conducted another scan later on within the day, as it was only me connected to the router at the time. This time the command line was: nmap -F -n -Pn 10.0.0.1/24

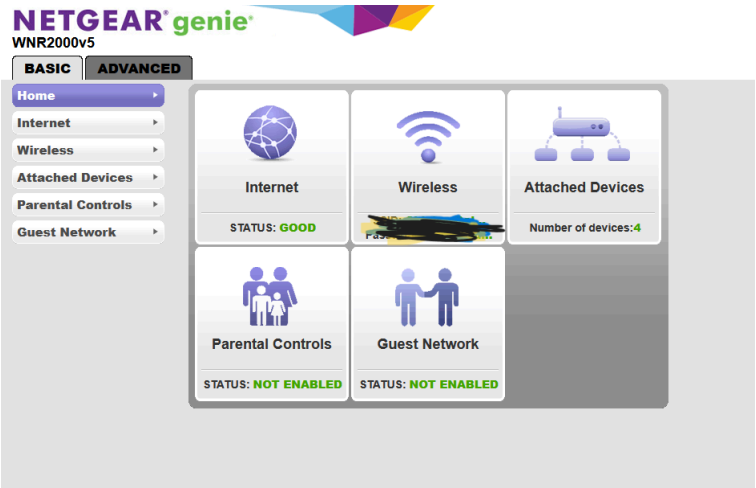

Target: 10.0.0.1/24
Profile:
Command: nmap -F -n -Pn 10.0.0.1/24
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
nmap -F -n -Pn 10.0.0.1/24
OS
Host
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.5
10.0.0.11
49152/tcp open unknown
MAC Address: 1E:7E:1F:F5:B4:92 (Unknown)
Nmap scan report for 10.0.0.11
Host is up (0.086s latency).
Not shown: 72 closed tcp ports (reset)
PORT STATE SERVICE
9/tcp filtered discard
13/tcp filtered daytime
37/tcp filtered time
81/tcp filtered hosts2-ns
88/tcp open kerberos-sec
427/tcp filtered svrloc
513/tcp filtered login
514/tcp filtered shell
544/tcp filtered kshell
548/tcp filtered afp
873/tcp filtered rsync
1026/tcp filtered LSA-or-nterm
1028/tcp filtered unknown
1110/tcp filtered nfsd-status
1433/tcp filtered ms-sql-s
2049/tcp filtered nfs
2121/tcp filtered ccproxy-ftp
2717/tcp filtered pn-requester
3000/tcp filtered ppp
5101/tcp filtered admdog
5900/tcp open vnc
7070/tcp open realserver
8000/tcp filtered http-alt
8008/tcp filtered http
8009/tcp filtered ajp13
8443/tcp filtered https-alt
9100/tcp filtered jetdirect
9999/tcp filtered abyss
MAC Address: EC:35:86:30:C5:F6 (Apple)
Nmap scan report for 10.0.0.2
Host is up (0.000031s latency).
Not shown: 97 closed tcp ports (reset)
PORT STATE SERVICE
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
Nmap done: 256 IP addresses (5 hosts up) scanned in 8.99 seconds


Target: 10.0.0.1/24
Profile:
Command: nmap -F -n -Pn 10.0.0.1/24
Hosts
Services
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
Hosts Viewer
Fisheye
Controls
Legend
OS
Host
10.0.0.1
10.0.0.2
10.0.0.3
10.0.0.5
10.0.0.11
10.0.0.1
10.0.0.3
localhost
10.0.0.5
10.0.0.2

Findings:

These findings showed me that there were around 5 live hosts on the subnet. The green means that there is good visibility however the yellow meant that they were scanned with limited information or mostly filtered ports.

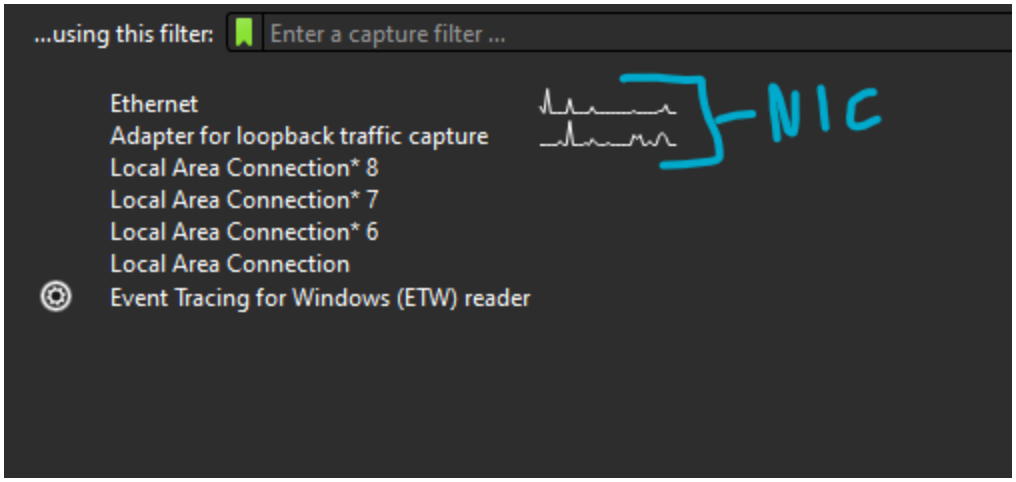
After this experiment, I went and checked on my own home router too see if the reading from nmap was accurate:



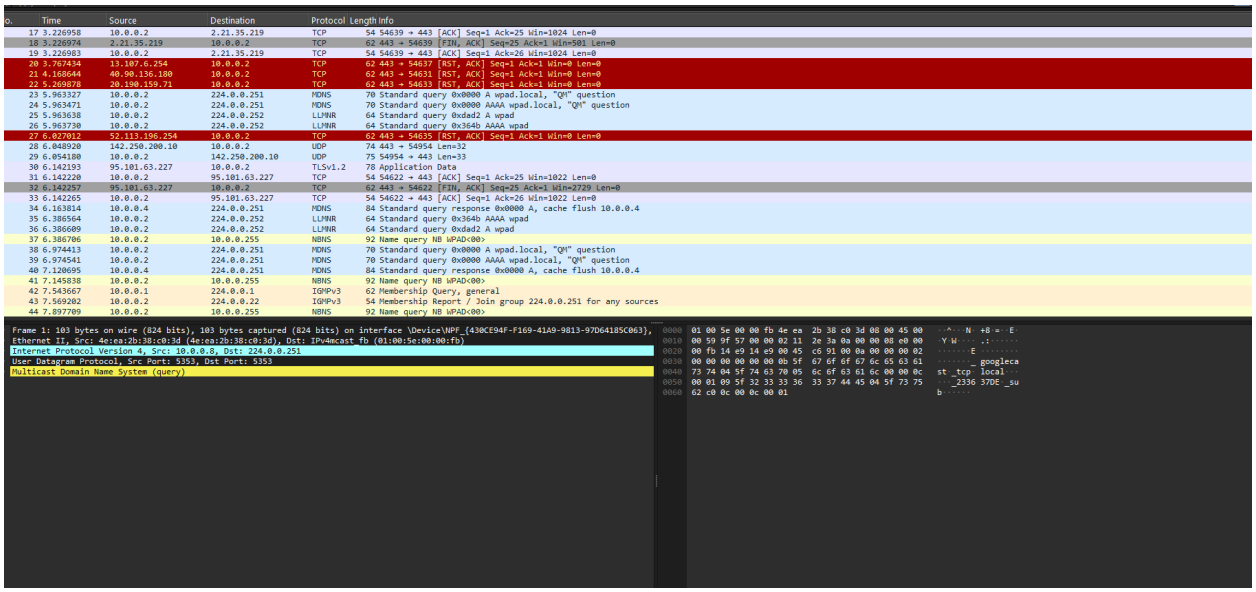

As a result of me wanting to dive deeper, I found a function of my router that would essentially route packets towards my own PC, this meant that when I would run wireshark, I wouldn't be getting only the PC packets, but also the packets for the other hosts. This was known as port forwarding. Although I wanted to test it out, I felt like if I start tinkering my family wouldn't be happy with the wifi being reset if something went wrong.

## Phase 2

Within this phase, I officially installed wireshark and had a look around the packets in which were being sent and received. When using wireshark, similarly to nmap, I watched a youtube tutorial in order to help get me started.



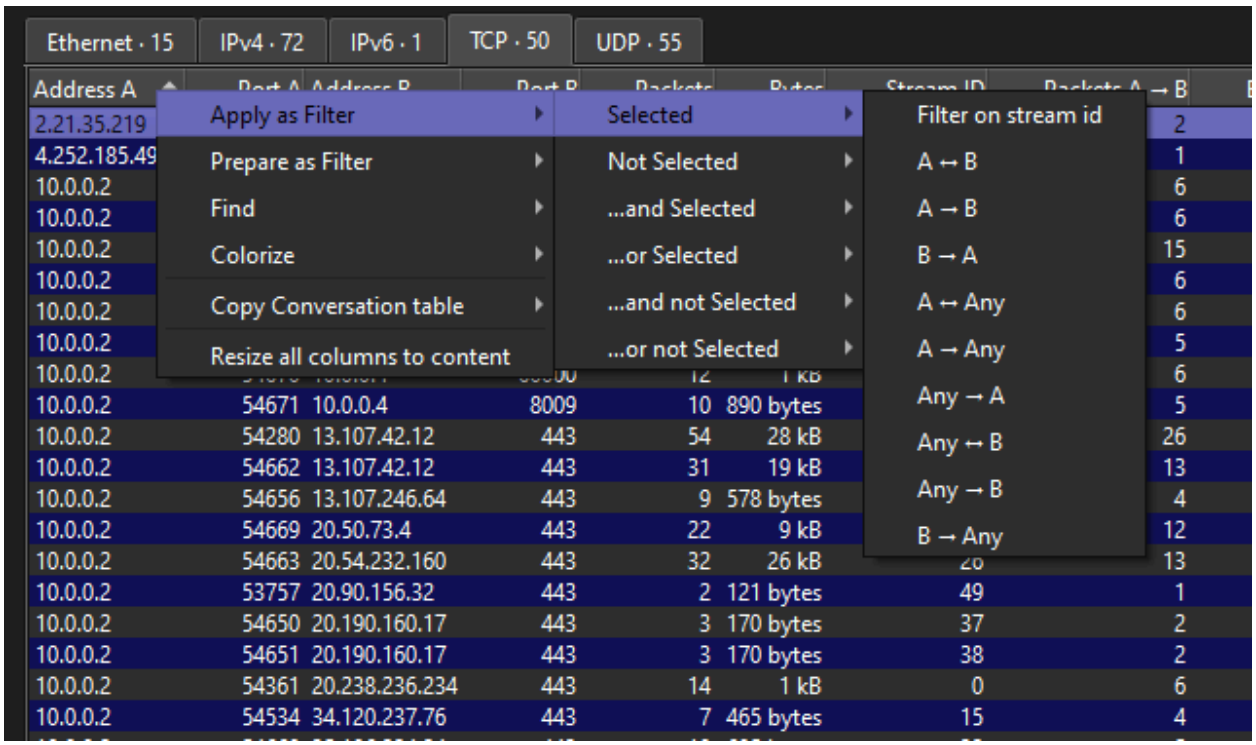

First I opened up wireshark and it showed me what I was connected too, from there I decided to have a look at my ethernet.



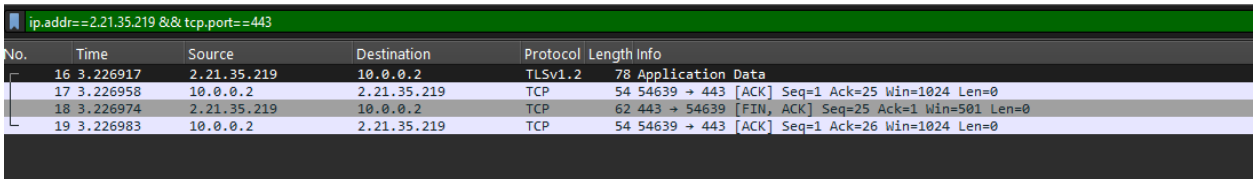

When I pressed ethernet, my computer began picking up a variety of packets. Once I felt like I had enough data, I then pressed the stop button. Each row that is currently being displayed on the screen is shown as a single packet.

So first I clicked on statistics and then clicked on conversations, this brought me to a page that shows me all of the PCAP that I have just captured.

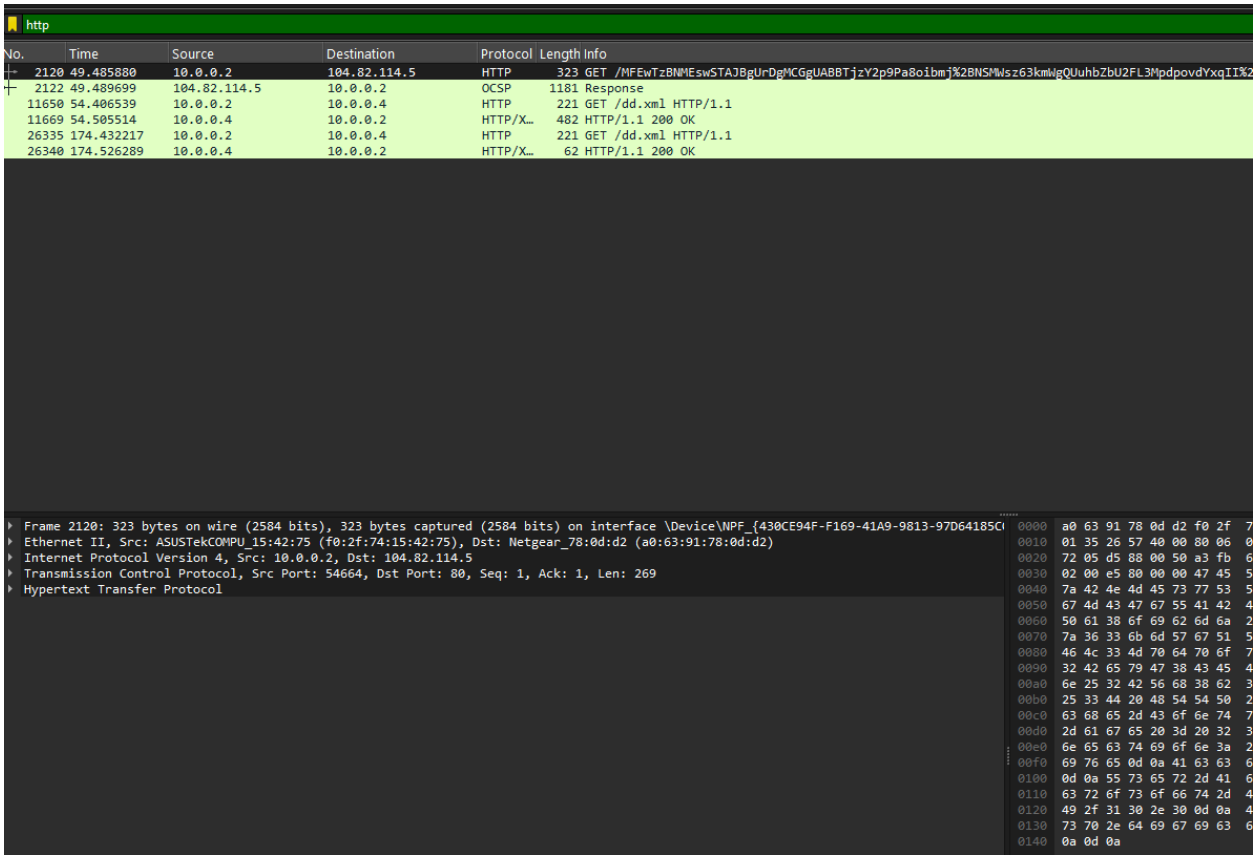

This allows me to see conversations, in particular, I look at TCP conversations and see any trends which may occur between my computer and another destination.



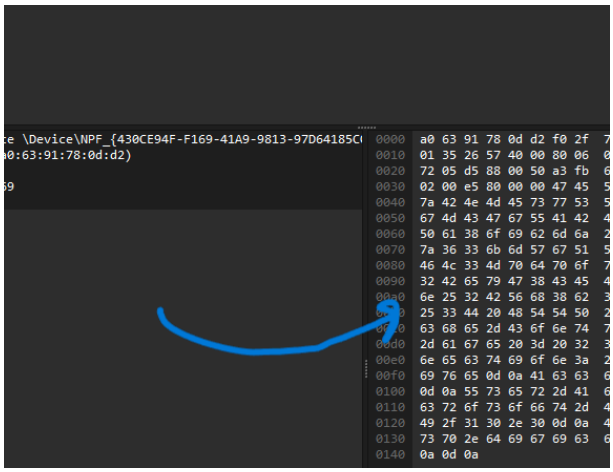

I then proceeded to filter out these packets through taking from A ←→ Any.




It than presented me with this page which filtered out the packets. After this, i decided to look at HTTP protocol packets in particular

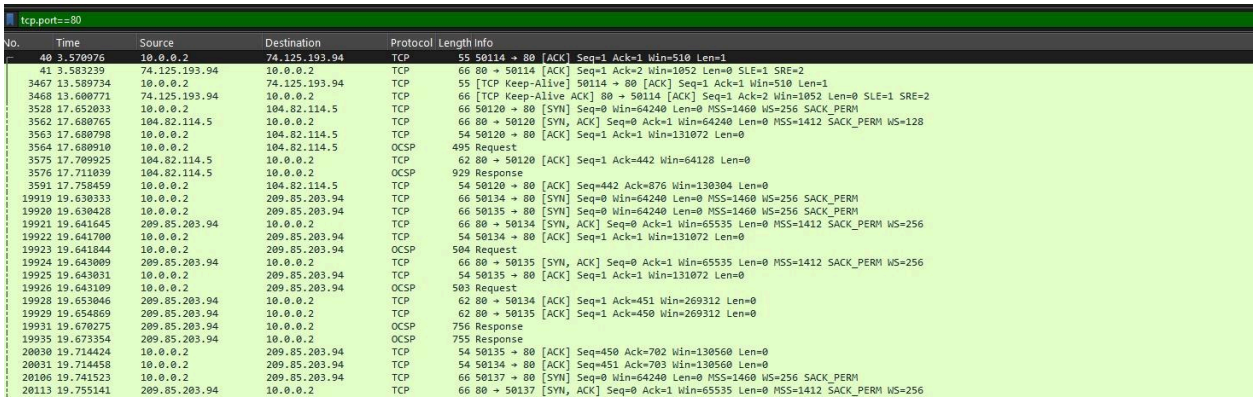



This allows me to be able to see HTTPs and if I input this information into a compiler, I would then be able to view the website itself. Therefore if you open up a phishing email or send it to a dodgy website, through the use of wireshark, I am able to see the information of those packets and where they are being sent.

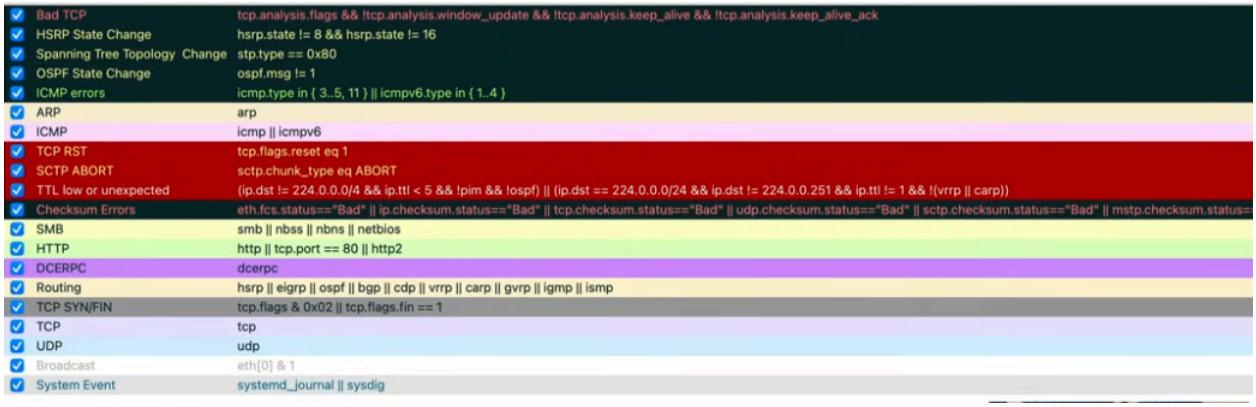



We can also see here from the packets that the data is encrypted. A feature in which wireshark offers is that for hardcore security, I am able to upload a decryption key and directly see the data that is being sent in those packets.


From this picture, you can see ARP working in action. This means that my netgear router is trying to locate a device on my local network with the ip address 10.0.0.4, this is broadcasted on all of the devices on the subnet that receives this request. The router is trying to find the MAC address of 10.0.0.4 in order to deliver packets to it.




I inputted a command that looked at port 80 using ‘’Tcp.port == 80 command’’, which scans all the packets from this port and presents them to me. Through this I was able to look at data that was being received and sent from this port.




In wireshark, certain packets are coloured in order for the user to understand more about the packets being sent as well as being able to easily identify any insecure packets that I may want to later investigate into.

## Conclusion

In conclusion to my personal experiment, I tried to analyze my own home personal network using wireshark and nmap. When using these tools, I did not have a specific agenda or goal in mind, I was really only exploring around my network trying to learn more about the software application. Seeing how my pathway in computer science is cybersecurity, I strongly felt that networks is one of my most important modules and it presented to me an opportunity to be able to use these tools in real practice and present them. I also felt that nmap and wireshark would help provide me with a step-up in understanding more about networking and cybersecurity later down the line.

References:

https://nmap.org/
https://www.wireshark.org/
https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs
https://www.freecodecamp.org/news/what-is-nmap-and-how-to-use-it-a-tutorial-for-the-greatest-scanning-tool-of-all-time/
https://www.youtube.com/watch?v=wt_xMols4Ww&t=431s
https://www.youtube.com/watch?v=qTaOZrDnMzQ
https://www.malware-traffic-analysis.net/