Megan Moore
Buff ID: 1014735

Research Report 2 - Wireshark

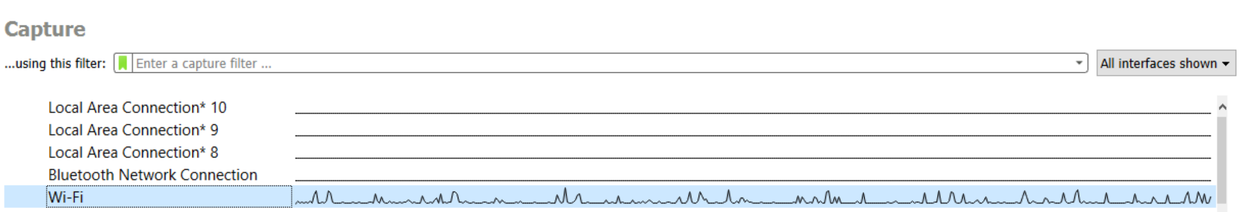
Question 1: what did you do?

For this assignment, we were instructed to download and install a network analysis tool called Wireshark and use it to capture packets in real time. This tool includes a variety of settings and filters which enables one to dig deep into network traffic and inspect individual packets. It does not directly detect cyber-attacks but aids in the identification and investigation of potential attacks. Once a user was skilled, you could use Wireshark to inspect a suspicious program's network traffic, analyze your own network's traffic flow or even troubleshoot network problems.

First, I scanned my home network which is password protected. I was surprised to see our network stretched a good distance past our back fence meaning anyone could park on the public road behind the fence and connect to our network if we left it open. I added network range extenders in our home so we could more seamlessly add wireless cameras but the added range could also be our downfall if we didn't secure it. Second, I then scanned my general neighborhood areas looking to find an unsecured option to connect to and test but amazingly, all of my neighbors intelligently have secured their networks. I know back when I was in college this was not the case as friends would surf for free internet connections all the time. How little we understood what that free YouTube video could have cost us. As I could not find any open networks to capture from, I took a little bit of time and was able to drive to a shopping center area. I found one open network to connect to and use Wireshark to capture packets. From all the packets I captured, home and the open network I found, I was able to inspect and apply filters by type such as "dns" as well as use Wireshark to follow a TCP stream. This allows me to see the entire TCP conversation which occurred between the client and server.

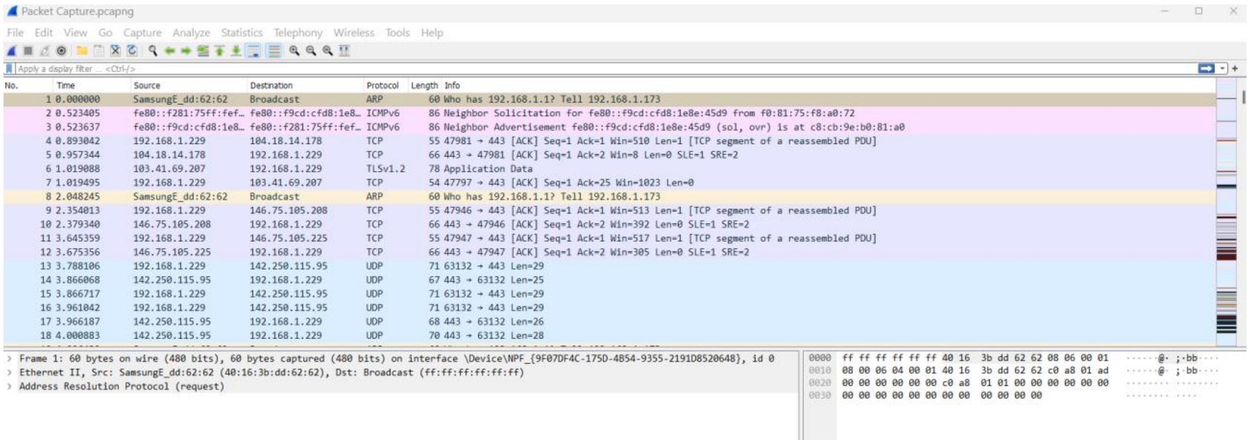
Question 2: what were the results?

As I again was unable to use this tool on my organization's network due to security reasons, I scanned my home network instead. After launching the Wireshark program, I captured a minimum of 5 minutes worth of packets specifically on the WIFI network interface and left the promiscuous mode enabled which allowed me to see all packets on the network aside from just those addressed to my network adapter. By capturing packets, this allows a user to examine the content and behavior of what is occurring on their network.

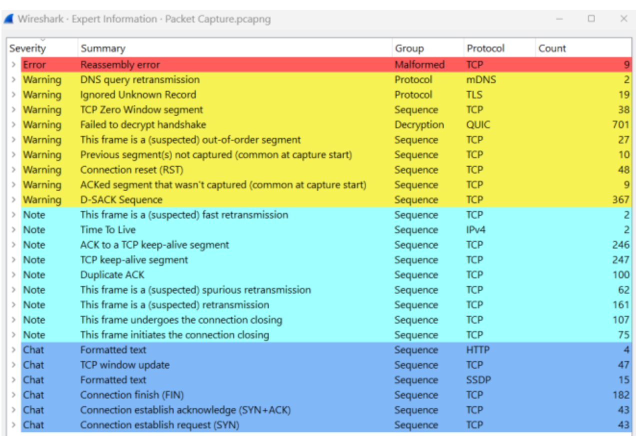


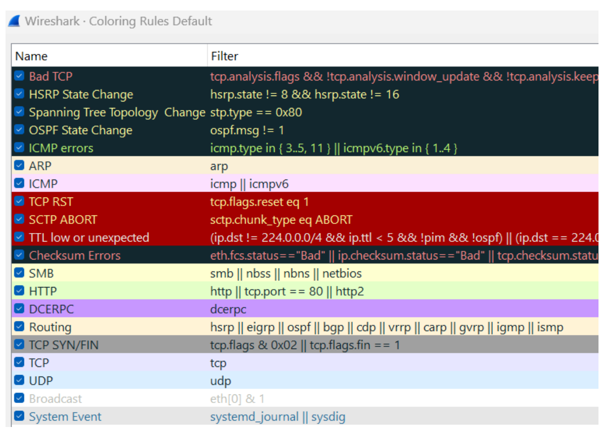
Megan Moore
Buff ID: 1014735

Research Report 2 - Wireshark



After running the scan, I sought to review the Expert Information piece. What's nice about this is it provides a snapshot for all the packets by color. My captures were grouped by severity level seen color coded in each screenshot below. Wireshark color codes and ranks the items. Blue indicates information of usual workflows which were a great deal of my captures. Cyan blue was indicative of when an application returned a simple or common error code. Warnings were highlighted in yellow and represent error codes such as connection issues. Then finally red which can be a more serious error and the grouping I had for those noted malformed packets. The malformed packets indicate there's a bug of sorts and due to this, the packet is aborted. I also included a reference image for the Color Rules from Wireshark. I did not deviate from the default settings.

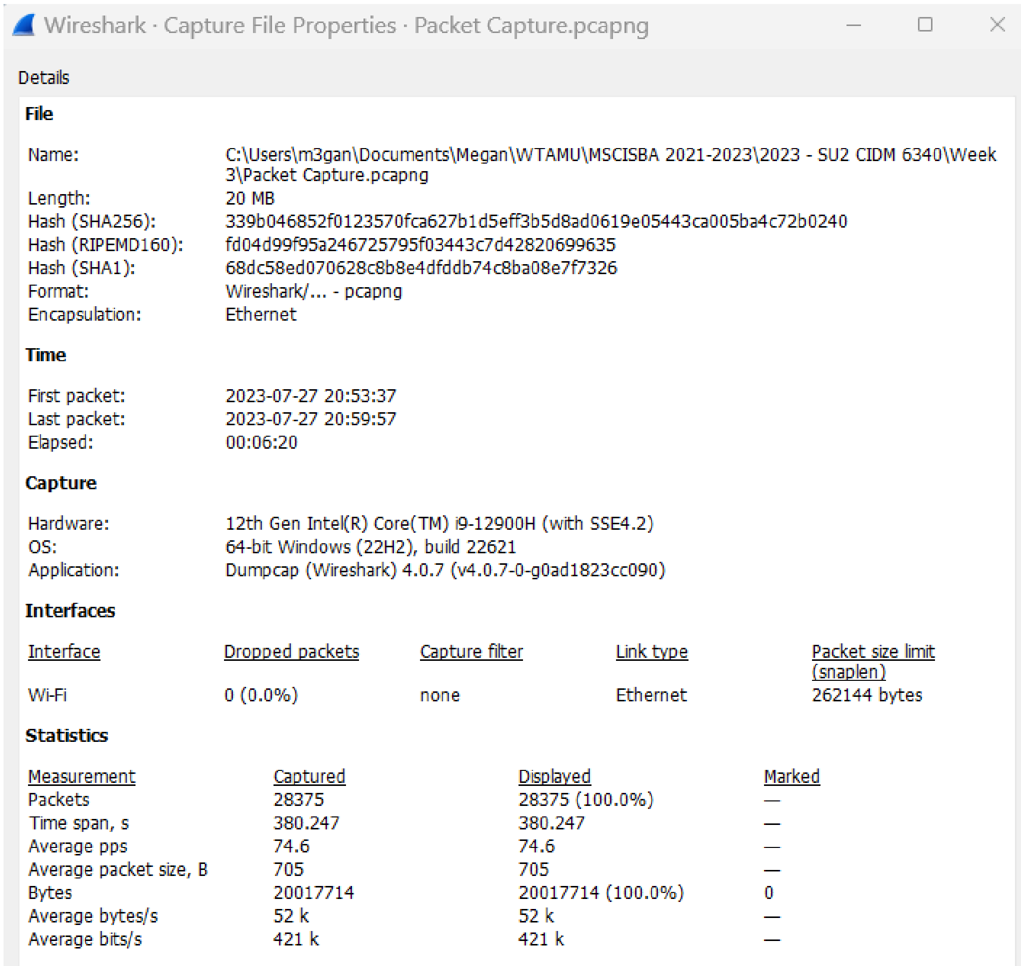




Almost all of my captures were grouped by sequence, which represents when a protocol sequence number is potentially suspicious meaning the sequence was not continuous or a retransmission was detected. I did find it interesting the high severity capture actually occurred on my home network which is secured compared to the open network I briefly captured from. When I reviewed the Capture File Properties, I noticed I had 0 Dropped Packets.

Megan Moore
Buff ID: 1014735

Research Report 2 - Wireshark



Wireshark · Capture File Properties · Packet Capture.pcapng

Details

File

Name:	C:\Users\m3gan\Documents\Megan\WTAMU\MSCISBA 2021-2023\2023 - SU2 CIDM 6340\Week 3\Packet Capture.pcapng
Length:	20 MB
Hash (SHA256):	339b046852f0123570fca627b1d5eff3b5d8ad0619e05443ca005ba4c72b0240
Hash (RIPEMD160):	fd04d99f95a246725795f03443c7d42820699635
Hash (SHA1):	68dc58ed070628c8b8e4dfddb74c8ba08e7f7326
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2023-07-27 20:53:37
Last packet:	2023-07-27 20:59:57
Elapsed:	00:06:20

Capture

Hardware:	12th Gen Intel(R) Core(TM) i9-12900H (with SSE4.2)
OS:	64-bit Windows (22H2), build 22621
Application:	Dumpcap (Wireshark) 4.0.7 (v4.0.7-0-g0ad1823cc090)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

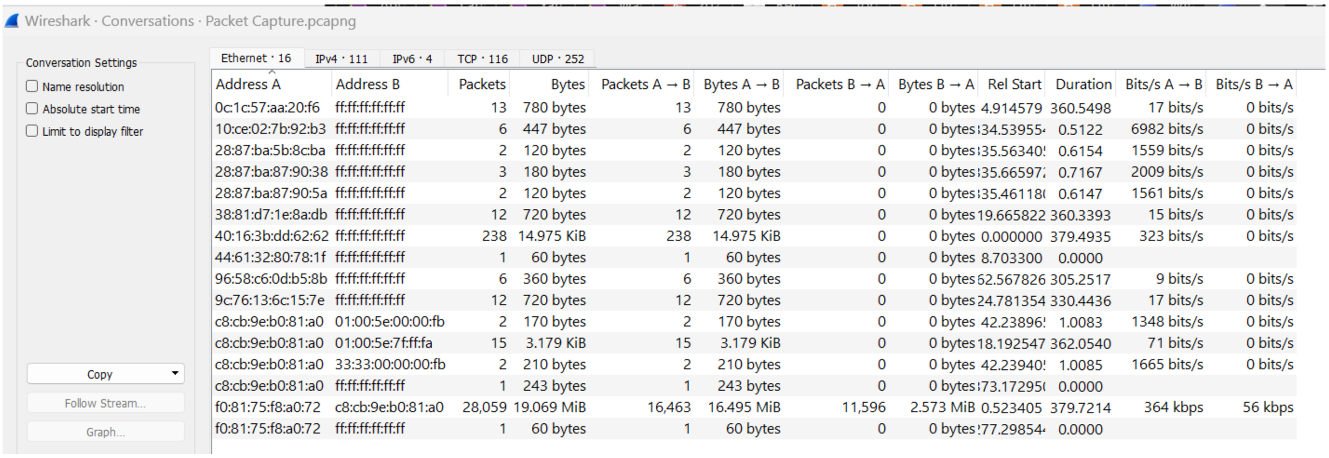
Measurement	Captured	Displayed	Marked
Packets	28375	28375 (100.0%)	—
Time span, s	380.247	380.247	—
Average pps	74.6	74.6	—
Average packet size, B	705	705	—
Bytes	20017714	20017714 (100.0%)	0
Average bytes/s	52 k	52 k	—
Average bits/s	421 k	421 k	—

Digging into the capture I did on my home network using the filtering, I found many of the yellow sequencing items to actually be stemming from our house-wide Ring camera system and what I believe to be is the cameras and Ring Wi-Fi range extenders retransmitting the signal. This brings me a bit of relief and I was somewhat worried initially seeing all the severity issues highlighted from the capture.

In addition to analyzing the scan, I did some Statistic review as well. First, I reviewed the Conversations Statistics for my scan which catalogs traffic between specific Ethernet/IP addresses. Each row in the list shows the statistical values for exactly one conversation.

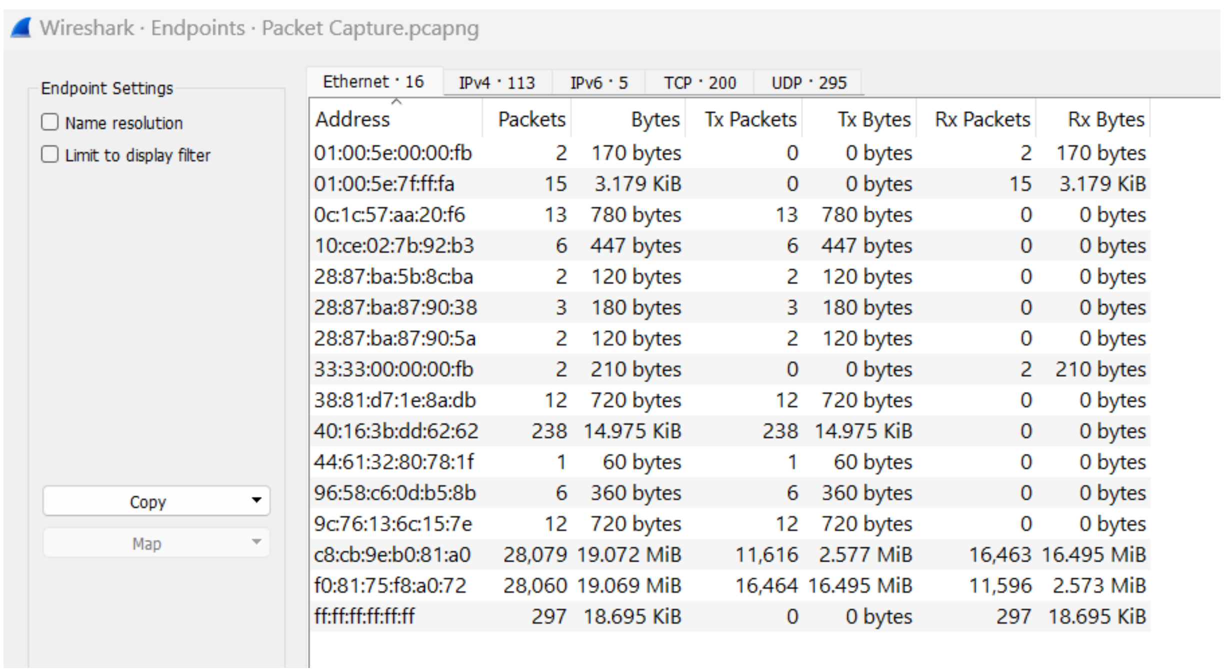
Megan Moore
Buff ID: 1014735

Research Report 2 - Wireshark



Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
0c:1c:57:aa:20:f6	ff:ff:ff:ff:ff:ff	13	780 bytes	13	780 bytes	0	0 bytes	4.914579	360.5498	17 bits/s	0 bits/s
10:ce:02:7b:92:b3	ff:ff:ff:ff:ff:ff	6	447 bytes	6	447 bytes	0	0 bytes	34.53955	0.5122	6982 bits/s	0 bits/s
28:87:ba:5b:8c:ba	ff:ff:ff:ff:ff:ff	2	120 bytes	2	120 bytes	0	0 bytes	35.56340	0.6154	1559 bits/s	0 bits/s
28:87:ba:87:90:38	ff:ff:ff:ff:ff:ff	3	180 bytes	3	180 bytes	0	0 bytes	35.66597	0.7167	2009 bits/s	0 bits/s
28:87:ba:87:90:5a	ff:ff:ff:ff:ff:ff	2	120 bytes	2	120 bytes	0	0 bytes	35.46118	0.6147	1561 bits/s	0 bits/s
38:81:d7:1e:8a:db	ff:ff:ff:ff:ff:ff	12	720 bytes	12	720 bytes	0	0 bytes	19.665822	360.3393	15 bits/s	0 bits/s
40:16:3b:dd:62:62	ff:ff:ff:ff:ff:ff	238	14.975 KiB	238	14.975 KiB	0	0 bytes	0.000000	379.4935	323 bits/s	0 bits/s
44:61:32:80:78:1f	ff:ff:ff:ff:ff:ff	1	60 bytes	1	60 bytes	0	0 bytes	8.703300	0.0000		
96:58:c6:0d:b5:8b	ff:ff:ff:ff:ff:ff	6	360 bytes	6	360 bytes	0	0 bytes	62.567826	305.2517	9 bits/s	0 bits/s
9c:76:13:6c:15:7e	ff:ff:ff:ff:ff:ff	12	720 bytes	12	720 bytes	0	0 bytes	24.781354	330.4436	17 bits/s	0 bits/s
c8:cb:9e:b0:81:a0	01:00:5e:00:00:fb	2	170 bytes	2	170 bytes	0	0 bytes	42.23896	1.0083	1348 bits/s	0 bits/s
c8:cb:9e:b0:81:a0	01:00:5e:7f:ff:fa	15	3.179 KiB	15	3.179 KiB	0	0 bytes	18.192547	362.0540	71 bits/s	0 bits/s
c8:cb:9e:b0:81:a0	33:33:00:00:00:fb	2	210 bytes	2	210 bytes	0	0 bytes	42.23940	1.0085	1665 bits/s	0 bits/s
c8:cb:9e:b0:81:a0	ff:ff:ff:ff:ff:ff	1	243 bytes	1	243 bytes	0	0 bytes	73.17295	0.0000		
f0:81:75:f8:a0:72	c8:cb:9e:b0:81:a0	28,059	19.069 MiB	16,463	16.495 MiB	11,596	2.573 MiB	0.523405	379.7214	364 kbps	56 kbps
f0:81:75:f8:a0:72	ff:ff:ff:ff:ff:ff	1	60 bytes	1	60 bytes	0	0 bytes	77.29854	0.0000		

Second Statistics I reviewed was Endpoints which notes the traffic to and from an Ethernet/IP address. A network endpoint is essentially the endpoint of separate protocol traffic from a specific protocol layer. Broadcast and multicast traffic will be shown separately as additional endpoints. Of course, as these aren't physical endpoints the real traffic will be received by some or all of the listed unicast endpoints.



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
01:00:5e:00:00:fb	2	170 bytes	0	0 bytes	2	170 bytes
01:00:5e:7f:ff:fa	15	3.179 KiB	0	0 bytes	15	3.179 KiB
0c:1c:57:aa:20:f6	13	780 bytes	13	780 bytes	0	0 bytes
10:ce:02:7b:92:b3	6	447 bytes	6	447 bytes	0	0 bytes
28:87:ba:5b:8c:ba	2	120 bytes	2	120 bytes	0	0 bytes
28:87:ba:87:90:38	3	180 bytes	3	180 bytes	0	0 bytes
28:87:ba:87:90:5a	2	120 bytes	2	120 bytes	0	0 bytes
33:33:00:00:00:fb	2	210 bytes	0	0 bytes	2	210 bytes
38:81:d7:1e:8a:db	12	720 bytes	12	720 bytes	0	0 bytes
40:16:3b:dd:62:62	238	14.975 KiB	238	14.975 KiB	0	0 bytes
44:61:32:80:78:1f	1	60 bytes	1	60 bytes	0	0 bytes
96:58:c6:0d:b5:8b	6	360 bytes	6	360 bytes	0	0 bytes
9c:76:13:6c:15:7e	12	720 bytes	12	720 bytes	0	0 bytes
c8:cb:9e:b0:81:a0	28,079	19.072 MiB	11,616	2.577 MiB	16,463	16.495 MiB
f0:81:75:f8:a0:72	28,060	19.069 MiB	16,464	16.495 MiB	11,596	2.573 MiB
ff:ff:ff:ff:ff:ff	297	18.695 KiB	0	0 bytes	297	18.695 KiB

Third I reviewed the Protocol Hierarchy of the captured packets from my home network. This displays a tree of all the protocols from the capture. Each row contains the values of one protocol. I thought a neat feature was that two of the columns (Percent Packets and Percent Bytes) also serve as bar graphs. I found it interesting to learn that packets usually contain multiple protocols. As a result, more than one protocol will be counted for each packet. In my screenshot 100% of packets are IP and 84.1% are TCP. I found this interesting as together that is much more than 100%. To explain this, I researched more on how Wireshark calculates this and discovered that protocol layers can consist of packets that won't contain any higher layer protocol, so the sum of all higher layer packets may not sum to the protocol's packet count. This can be caused by segments and fragments reassembled in other frames and potentially other undissected data. I also noticed entry in the PDUs column could be greater than that of Packets. For my scan, there are many more TLS PDUs than there are packets.

Wireshark · Protocol Hierarchy Statistics · Packet Capture.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	28375	100.0	20017714	421 k	0	0	0	28375
Ethernet	100.0	28375	2.1	411335	8654	0	0	0	28375
Internet Protocol Version 6	0.8	216	0.0	8640	181	0	0	0	216
User Datagram Protocol	0.6	158	0.0	1264	26	0	0	0	158
Multicast Domain Name System	0.0	2	0.0	86	1	2	86	1	2
Domain Name System	0.5	156	0.1	12084	254	156	12084	254	156
Internet Control Message Protocol v6	0.2	58	0.0	1752	36	58	1752	36	58
Internet Protocol Version 4	98.4	27913	2.8	558260	11 k	0	0	0	27913
User Datagram Protocol	14.3	4050	0.2	32400	681	0	0	0	4050
TP-Link Smart Home Protocol	0.0	3	0.0	87	1	3	87	1	3
Simple Service Discovery Protocol	0.1	15	0.0	2625	55	15	2625	55	15
QUIC IETF	12.0	3391	5.6	1130455	23 k	3391	948978	19 k	3639
NetBIOS Datagram Service	0.0	1	0.0	201	4	0	0	0	1
SMB (Server Message Block Protocol)	0.0	1	0.0	119	2	0	0	0	1
SMB MailSlot Protocol	0.0	1	0.0	25	0	0	0	0	1
Microsoft Windows Browser Protocol	0.0	1	0.0	33	0	1	33	0	1
Multicast Domain Name System	0.0	2	0.0	86	1	2	86	1	2
Domain Name System	0.2	52	0.0	4111	86	52	4111	86	52
Data	2.1	586	0.1	20611	433	586	20611	433	586
Transmission Control Protocol	84.1	23863	89.3	17872036	376 k	19665	15401708	324 k	23863
Transport Layer Security	14.4	4083	86.9	17389434	365 k	4083	15161984	318 k	4250
Malformed Packet	0.0	9	0.0	0	0	9	0	0	9
Hypertext Transfer Protocol	0.0	4	0.0	536	11	4	536	11	4
Data	0.4	102	0.0	1377	28	102	1377	28	102
Address Resolution Protocol	0.9	246	0.1	11154	234	246	11154	234	246

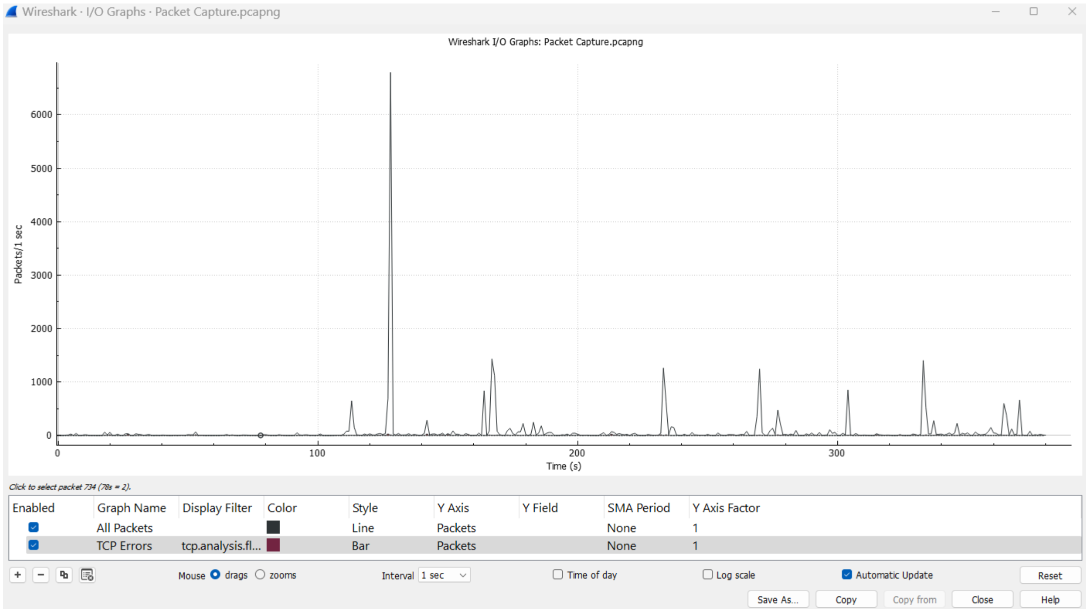
No display filter.

Close Copy

Last, I revied the IO Graph which visualizes the number of packets in time. The chart drawing area is shown along with a customizable list of graphs. These graphs are divided into time intervals. Wireshark's I/O Graph window doesn't distinguish between missing and zero values. For scatter plots it is assumed that zero values indicate missing data, and those values are omitted. Zero values are shown in line graphs, and bar charts. The bar graph displays steep points over time where high amounts of packets were captured. I felt this probably very realistic and network communication comes and goes depending on what is being used at that moment.

Megan Moore
Buff ID: 1014735

Research Report 2 - Wireshark



For the open network capture, I received a high number of DNS query retransmission items. I looked into this and it simply means the DNS query has no corresponding response received by the host and so it sends the query again. Often retransmission like this occurs due to the Windows client timing out too quickly so the server response and the client's retransmission basically pass each other in the network.

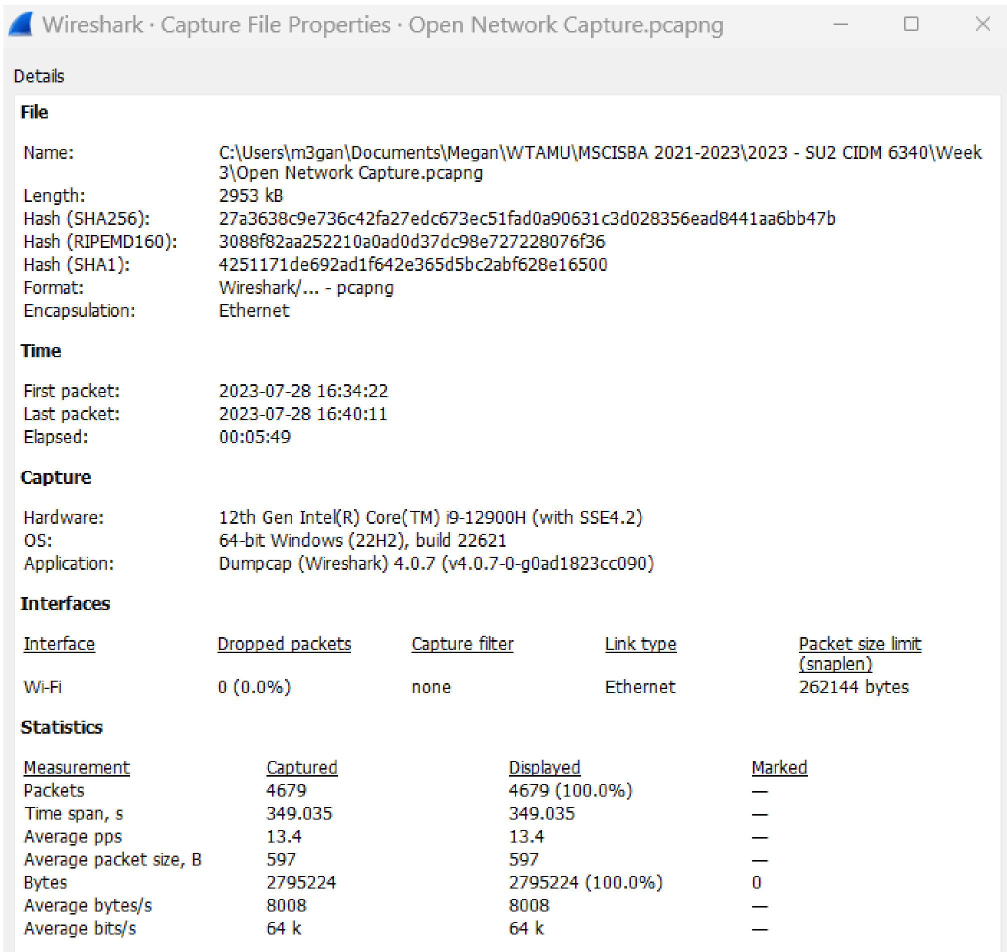
Wireshark · Expert Information · Open Network Capture.pcapng

Severity	Summary	Group	Protocol	Count
Error	Reassembly error	Malformed	TCP	6
Warning	This frame is a (suspected) out-of-order segment	Sequence	TCP	6
Warning	TCP Zero Window segment	Sequence	TCP	2
Warning	ACKed segment that wasn't captured (common at capture start)	Sequence	TCP	2
Warning	Previous segment(s) not captured (common at capture start)	Sequence	TCP	5
Warning	Failed to decrypt handshake	Decryption	QUIC	59
Warning	Connection reset (RST)	Sequence	TCP	20
Warning	D-SACK Sequence	Sequence	TCP	73
Note	This session reuses previously negotiated keys (Session resumption)	Sequence	TLS	2
Note	ACK to a TCP keep-alive segment	Sequence	TCP	60
Note	TCP keep-alive segment	Sequence	TCP	60
Note	This frame undergoes the connection closing	Sequence	TCP	58
Note	This frame initiates the connection closing	Sequence	TCP	57
Note	Duplicate ACK	Sequence	TCP	35
Note	This frame is a (suspected) spurious retransmission	Sequence	TCP	17
Note	This frame is a (suspected) retransmission	Sequence	TCP	42
Chat	Formatted text	Sequence	SSDP	12
Chat	Connection finish (FIN)	Sequence	TCP	115
Chat	TCP window update	Sequence	TCP	1
Chat	Connection establish acknowledge (SYN+ACK)	Sequence	TCP	52
Chat	Connection establish request (SYN)	Sequence	TCP	52
Chat	Formatted text	Sequence	HTTP	38

Megan Moore
Buff ID: 1014735

Research Report 2 - Wireshark

In the Capture File properties, I again had 0 dropped packets. But what I noticed is a sharp decrease in the number of packets scanned. While my home network turned up with over 28,000 the open network scan only counted 4679.


Wireshark · Capture File Properties · Open Network Capture.pcapng

Details

File

Name:	C:\Users\m3gan\Documents\Megan\WTAMU\MSCISBA 2021-2023\2023 - SU2 CIDM 6340\Week 3\Open Network Capture.pcapng
Length:	2953 kB
Hash (SHA256):	27a3638c9e736c42fa27edc673ec51fad0a90631c3d028356ead8441aa6bb47b
Hash (RIPEMD160):	3088f82aa252210a0ad0d37dc98e727228076f36
Hash (SHA1):	4251171de692ad1f642e365d5bc2abf628e16500
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2023-07-28 16:34:22
Last packet:	2023-07-28 16:40:11
Elapsed:	00:05:49

Capture

Hardware:	12th Gen Intel(R) Core(TM) i9-12900H (with SSE4.2)
OS:	64-bit Windows (22H2), build 22621
Application:	Dumpcap (Wireshark) 4.0.7 (v4.0.7-0-g0ad1823cc090)

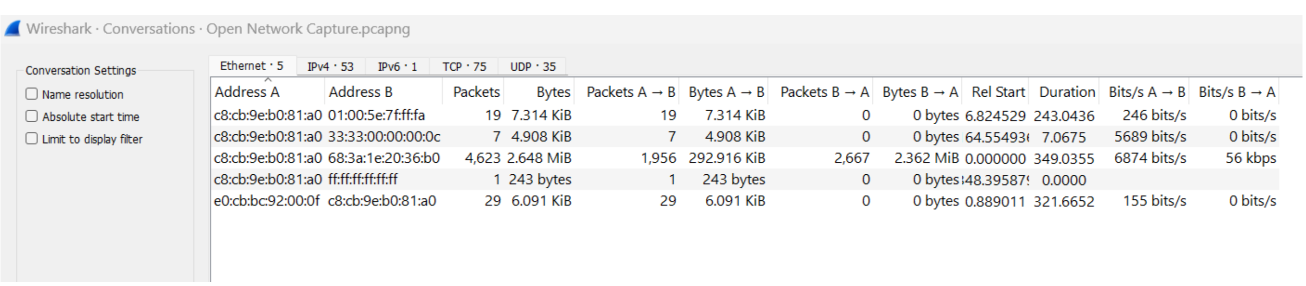
Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Wi-Fi	0 (0.0%)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	4679	4679 (100.0%)	—
Time span, s	349.035	349.035	—
Average pps	13.4	13.4	—
Average packet size, B	597	597	—
Bytes	2795224	2795224 (100.0%)	0
Average bytes/s	8008	8008	—
Average bits/s	64 k	64 k	—

I also reviewed all of the same Statistics for the open network capture. Most notable differences were a higher percentage of TCP protocol captures and the bytes of the packets were much larger.


Wireshark · Conversations · Open Network Capture.pcapng

Conversation Settings
- Name resolution
- Absolute start time
- Limit to display filter

Ethernet · 5 | IPv4 · 53 | IPv6 · 1 | TCP · 75 | UDP · 35

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
c8:cb:9e:b0:81:a0	01:00:5e:7f:ff:fa	19	7.314 KiB	19	7.314 KiB	0	0 bytes	6.824529	243.0436	246 bits/s	0 bits/s
c8:cb:9e:b0:81:a0	33:33:00:00:00:0c	7	4.908 KiB	7	4.908 KiB	0	0 bytes	64.55493(	7.0675	5689 bits/s	0 bits/s
c8:cb:9e:b0:81:a0	68:3a:1e:20:36:b0	4,623	2.648 MiB	1,956	292.916 KiB	2,667	2.362 MiB	0.000000	349.0355	6874 bits/s	56 kbps
c8:cb:9e:b0:81:a0	ff:ff:ff:ff:ff:ff	1	243 bytes	1	243 bytes	0	0 bytes	148.39587!	0.0000		
e0:cb:bc:92:00:0f	c8:cb:9e:b0:81:a0	29	6.091 KiB	29	6.091 KiB	0	0 bytes	0.889011	321.6652	155 bits/s	0 bits/s

Megan Moore
Buff ID: 1014735

Research Report 2 - Wireshark

Wireshark · Endpoints · Open Network Capture.pcapng

Endpoint Settings
- [ ] Name resolution
- [ ] Limit to display filter

Ethernet · 6 | IPv4 · 54 | IPv6 · 2 | TCP · 121 | UDP · 44

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
01:00:5e:7f:ff:fa	19	7.314 KiB	0	0 bytes	19	7.314 KiB
33:33:00:00:00:0c	7	4.908 KiB	0	0 bytes	7	4.908 KiB
68:3a:1e:20:36:b0	4,623	2.648 MiB	2,667	2.362 MiB	1,956	292.916 KiB
c8:cb:9e:b0:81:a0	4,679	2.666 MiB	1,983	305.376 KiB	2,696	2.368 MiB
e0:cb:bc:92:00:0f	29	6.091 KiB	29	6.091 KiB	0	0 bytes
ff:ff:ff:ff:ff:ff	1	243 bytes	0	0 bytes	1	243 bytes

Wireshark · Protocol Hierarchy Statistics · Open Network Capture.pcapng

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDUs
Frame	100.0	4679	100.0	2795224	64 k	0	0	0	4679
Ethernet	100.0	4679	2.4	68374	1567	0	0	0	4679
Internet Protocol Version 6	0.1	7	0.0	280	6	0	0	0	7
User Datagram Protocol	0.1	7	0.0	56	1	0	0	0	7
Data	0.1	7	0.2	4592	105	7	4592	105	7
Internet Protocol Version 4	98.9	4626	3.3	92520	2120	0	0	0	4626
User Datagram Protocol	6.8	318	0.1	2544	58	0	0	0	318
Simple Service Discovery Protocol	0.3	12	0.1	2100	48	12	2100	48	12
QUIC IETF	5.1	240	2.9	82390	1888	240	66642	1527	261
NetBIOS Datagram Service	0.0	1	0.0	201	4	0	0	0	1
SMB (Server Message Block Protocol)	0.0	1	0.0	119	2	0	0	0	1
SMB MailSlot Protocol	0.0	1	0.0	25	0	0	0	0	1
Microsoft Windows Browser Protocol	0.0	1	0.0	33	0	1	33	0	1
Domain Name System	1.2	58	0.2	6190	141	58	6190	141	58
Data	0.1	7	0.2	4592	105	7	4592	105	7
Transmission Control Protocol	92.1	4308	90.6	2533820	58 k	2786	1442891	33 k	4308
Transport Layer Security	30.7	1438	47.8	1334822	30 k	1438	1293697	29 k	1461
Malformed Packet	0.1	6	0.0	0	0	6	0	0	6
Hypertext Transfer Protocol	0.8	38	40.2	1123932	25 k	24	5009	114	38
Online Certificate Status Protocol	0.0	2	0.0	942	21	2	942	21	2
Line-based text data	0.2	11	0.0	242	5	11	242	5	11
Data	0.0	1	39.9	1114800	25 k	1	1114800	25 k	1
Data	0.9	40	0.0	40	0	40	40	0	40
Address Resolution Protocol	1.0	46	0.1	1702	39	46	1702	39	46

No display filter.

Close | Cop

While I never saw any suspicious attempts to access my own laptop, perhaps if I left a scan run much longer there is a possibility I would. Many individuals do not have firewalls or encryption on their devices nor does the open network have attack prevention built in. In this case, a cyber-attack using an open network is highly likely. In order to safely utilize an open network, a user needs to be familiar with their attack surface and what doesn't look normal when reviewing a scan.

Megan Moore
Buff ID: 1014735

Research Report 2 - Wireshark

Question 3: what did you learn?

This assignment opened my eyes to the truth behind what I have always heard of "Never connect to an open Wi-Fi network". I learned my entire connection is essentially broadcasted, and when on an open network it's unencrypted, so all that someone needs to watch what I do is a wireless interface capture tool in monitoring mode. While the initial capture looks like garbled amounts of characters and information, using a tool to filter it organizes it very easily. Without any encryption, it would be comparable to me writing my passwords down on the back of a postcard and sending that through the public mail system. Anyone could see it plain as day.

Also, I had no idea how simple it is for black hat hackers to intentionally broadcast their own signal pretending to be a McDonalds or Starbucks open network and then capture all of the traffic automatically. This is how important information like passwords, banking logins or work security features can be stolen. And that individual doesn't have to sort out your information that moment. They can save the capture and dig through it at a later time. In short, I learned if I need to use a connection and there are only public open options available, I should never trust it and use VPN. A VPN tunnel adds a layer of safety. The open network also has many suspicious communications occurring over the network.

While reviewing the results, I noticed Wireshark does not explicitly display the user attack surface. What it does do however, is it provides the necessary tools and information to help a user understand their network vulnerabilities and potential attack vectors. By a user analyzing network traffic, they are able to gain insight into the devices, services and protocols present on the network in use. In this way, Wireshark is a very valuable tool for detecting a cyber-attack through analyzing suspicious activity and network communication.