William Adams
Lab two: Network Sniffing
IS 3513-002
November 28, 2023
Professor Wooldridge

In lab 4, “Network Sniffing we got to use and experiment with “Wireshark” for the first time. “Wireshark is a network packet analyzer. A network packet analyzer presents captured packet data in as much detail as possible.” (What) I have used Wireshark on and off for the past 2 years, so I was familiar on how to get around the application well. After reading the instructions and the steps on the lab I was confident in my ability to get it done without any major problems. Wireshark is a major application for a cyber security worker in today’s workforce, so I have been trying to gain more knowledge with the application and practicing as much as I can. With this being the last lab in this class, I was very excited to gain some more knowledge, as these labs have helped me develop my thoughts more and just gain more knowledge in general about cyber security applications that can help me in the future.

To start the lab, we were supposed to download and install Wireshark on our device. I’ve had Wireshark downloaded on my device for probably about three years now, so I didn’t have to do the whole download process which was nice. When I did download Wireshark years ago, I didn’t have any problems, it was very simple and an easy installation. With the second step in the lab, we had to put it into capture mode on our wireless connection. This part was easy to follow, I have done packet capturing from time to time, so I had no real trouble with this step. I then accessed a single site for the next step in the lab, I accessed a site on a web browser I was using and then observed what I was looking at after I stopped the capture. When accessing a website using the application Wireshark to capture network traffic, I did see HTTP packets when analyzing the capture that I had conducted when accessing the website. I thought this was pretty interesting because when you do a normal capture of traffic on the network and you don’t access any sites, you don’t get any HTTP packets that come in. I then cleared my capture and then started another capture where I Logged onto a remote service or application via your network ISP or another Internet connection that requires authentication. For this step I had a little trouble, but it was my fault because I wasn’t reading the step correctly. I can’t even say what I was reading it as, but I finally got what I was supposed to do. When logging onto a remote service or application via your network ISP or another Internet connection that requires authentication and capturing the traffic with Wireshark, the captured data will reveal the details of the authentication process. I then had to repeat the steps from step five and six and do them again for 8 other protocols that I had to capture. The first protocol that I had to capture was an ARP protocol. ARP (Address Resolution Protocol) may contribute to local network communication; it plays a limited role in authenticating with remote services. Capturing ARP (Address Resolution Protocol) traffic is generally not a challenging task due to the nature of how ARP operates within a network. ARP is a fundamental protocol used for mapping IP addresses to MAC addresses at the local network level. Unlike some higher-layer protocols that might involve encryption or complex handshakes, ARP operates as a simple broadcast protocol. With the second protocol that I had to capture, I had to capture a TCP protocol from my capturing. When engaging in the logon process to a remote service or application via your network ISP or another Internet connection requiring authentication, TCP serves as the foundational protocol orchestrating the reliable and structured communication between your device and the remote server. The TCP three-way handshake initiates the connection, ensuring a systematic setup that establishes a secure channel for data exchange. There were also a ton of TCP protocols that were captured. The third protocol that I had to capture for this step was UDP (User Datagram Protocol). When logging onto a remote service or application via your network ISP or another Internet connection that requires authentication, the role of UDP (User Datagram Protocol) in the captured traffic is often associated with specific types of applications. Unlike TCP, UDP is a connectionless protocol, which means it doesn't establish a persistent connection before transmitting data. In scenarios requiring authentication, UDP might be involved in certain aspects of the communication. There were tons of UDP packets that were captured during this time. For the next protocol that I had to capture for this step was HTTP (Hypertext Transfer Protocol). HTTP plays a crucial role in facilitating the exchange of information between the client and the remote server. I thought it was really interesting that when you access a site on the internet that there will be HTTP packets that come up. In the case of authentication, the HTTP request may include login credentials, usually in the form of a username and password. For the next protocol that I have captured for this step is HTTPS (TLS). For this specific lab I have used the TLS (Transport Layer Security) protocol. TLS is very crucial for securing the authentication process for a network. The TLS protocols that I had secured was actually pretty interesting, they had YouTube on the source. I did look up something random on YouTube for this specific situation, so I thought it was very cool how it said YouTube on the source. The next protocol that I had captured on this step was FTP (File Transfer Protocol), this was probably the protocol that I had the hardest time to recover and gave me some problems. I honestly wasn't able to recover the protocol, I did watch the video that was provided if we were having trouble, and I couldn't find where to put the connection in loopback mode. I'm not sure if it's because I am on a mac computer and the video that was provided was on windows, I couldn't figure it out sadly. For the next protocol that I needed to capture for this step was ICMP (Internet Control Message Protocol). ICMP packets, while not directly involved in authentication, contribute to the overall network health and efficiency, reflecting the dynamic nature of network communication. I was able to get a lot of packets that came in with this protocol and were named "ICMPv6" so I'm not 100% if that is correct but those were the only ICMP protocols that I had received on my end from my capture. For the last protocol that I had recovered that I needed for this step was DNS (Domain Name System). In the Wireshark capture, the DNS packets reveal the initial steps of the communication, showcasing the translation of human-readable domain names into machine-readable IP addresses. I was able to recover a lot of DNS protocols from my capture, which it was cool to see how my laptop name was on the source. After I had gotten all of the protocols that was required, I restarted my scan and let it run for an hour long and then I stopped to see the resorts. I was at my girlfriend's house while I was doing the scan, it was really fascinating everything that I picked up in the hour that the scan was going on for. I was able to pick up girlfriend's phone source which was really fascinating to see all of that. I was also doing homework while the scan was taking place, it was cool to see how the capture got the websites I was using while the capture was going on. After doing the four-hour capture, it was also really interesting to see how much data and packets were recovered in that time span. During that time, I was just doing homework and watching tv and it is crazy what it can discover when you are capturing data traffic over the network.

HTTP (Hypertext Transfer Protocol) can be a little concerning when you really think about it, especially when transmitting sensitive information. HTTP sends data in plaintext, making it susceptible to interception and unauthorized access. This lack of encryption raises concerns about the confidentiality of personal data, such as login credentials or private messages.

Wireshark has been a very useful and powerful tool for security professionals for many years. Wireshark was founded in 1998 by Gerald Combs and is still one of the most useful tools to date for data capturing over a network. "Wireshark intercepts traffic and converts that binary traffic into human-readable format. This makes it easy to identify what traffic is crossing your network, how much of it, how frequently, how much latency there is between certain hops, and so forth." (Porup 2018) I have only been using Wireshark for a couple years now and I'm no expert or professional, but I could tell how valuable a tool like this can be when finding data traffic. When I had Wireshark capture my network for about an hour, it was insane to me how much data it captured during the time period, so I can only imagine what a security professional can do with the application. One of the most impactful tools that Wireshark offers is that it provides real-time visuality into what is going on with a networks traffic. This will allow professionals to scrutinize packets at a granular level and be able to use this application on the daily. Security professionals use this application to conduct analyses of network traffic patterns. With how many filters that Wireshark offers, it can make it easier to see what you want to view. By investigating and examining packet details, they can identify security threats, vulnerabilities, and unauthorized access to networks. "Given the large volume of traffic that crosses a typical business network, Wireshark's tools to help you filter that traffic are what make it especially useful. Capture filters will collect only the types of traffic you're interested in, and display filters will help you zoom in on the traffic you want to inspect." (Porup 2018) In this lab specifically, the amount of traffic that can come through in such a short amount of time is ridiculous to me. The use of filters with Wireshark can be extremely helpful when it comes to finding specific information. Wireshark can also have great significance in security operations, it can help you find network vulnerabilities and identify them as fast as possible.

Security professions can use Wireshark as a tool to prevent future or detect incidents that are happening. Cyber security analyst can set filters in place for them to focus on specific types of traffic that is coming into the network. This can be malware infections, DDoS attacks, or any unauthorized activity that is happening over the network. "It also gives cybersecurity professionals and cybercrime forensic investigators the ability to trace network connections. Using it, they could access the contents of suspected transactions in order catch criminal and malicious activity." (OT 2021) I thought this quote was very interesting and is crazy how cybercrime investigator teams can use this application to really catch malicious actors. It does make sense on why you could use this application to catch suscepts, using this to look over their network to see if there is any malicious activity that is going on over the network. Wireshark can enhance the support of incident response by providing a detailed record of network during a security breach that may have happened. This is extremely valuable for many reasons, the fact that you can get real time footage of what Is coming in and out of a network is crucial for incident response. The importance of Wireshark in threat intelligence cannot be overstated. By enabling analysts to link existing danger indicators with network activity, the solution improves the organization's capacity to proactively counter new attacks. Security teams can maintain a proactive and knowledgeable security posture by swiftly identifying and responding to patterns linked to hostile actors by utilizing Wireshark's filtering capabilities and integrating threat intelligence feeds. Additionally, Wireshark helps with compliance initiatives by making network activity monitoring and audits easier. Sensitive data must be transmitted securely according to compliance standards, and Wireshark helps businesses to confirm that these rules are being followed. The technology may be used by security experts to examine data flows and make sure that access rules, encryption standards, and other compliance procedures are always followed. “Wireshark can be used to audit network security configurations and policies. By analyzing network traffic, security professionals can identify vulnerabilities, weak spots, and potential security risks.” (Ashwani 2023) I thought this quote was very interesting, security audits are very crucial to any organization due to the fact that it can limit vulnerabilities in the network. Wireshark can unravel patterns that are happening in the network that can be patched by a security audit or there may have been a rarity that has happened on the network that can be looked at from the filters. With how much data and packets are sent and received every single second of every single day, no matter where you are, I can see how this can have a pivotal role in how cyber security professionals do their profession on a day to day basis.

I believe that Wireshark is great for any organization to use but I don’t believe it is the only thing an organization can count on to provide security for there organization. Following security procedures and following security guidelines make a bigger impact on the security of an organization. In other courses that I have taken past and this semester, it has become paramount on how critical security procedures really are in the cyber industry. While Wireshark can provide a great tool that you can use with the use of data capturing over a network, it can only do so much in terms of security. On the other hand, Wireshark is probably the most valuable tool that I have come across while learning about cyber security so I think both can be true about how important Wireshark is.

Appliances

1910	7.870758	Wills-MBP.attlocal…	pki-goog.l.google.…	HTTP	448	GET /gts1c3/MFAwTjBMMEowSDA
1921	7.889867	pki-goog.l.google.…	Wills-MBP.attlocal…	OCSP	798	Response
6095	13.181495	Wills-MBP.attlocal…	pki-goog.l.google.…	HTTP	450	GET /gts1c3/MFAwTjBMMEowSDA
6104	13.199951	pki-goog.l.google.…	Wills-MBP.attlocal…	OCSP	798	Response
6619	15.457630	Wills-MBP.attlocal…	pki-goog.l.google.…	HTTP	452	GET /gts1c3/MFAwTjBMMEowSDA
6677	15.475923	pki-goog.l.google.…	Wills-MBP.attlocal…	OCSP	799	Response
7338	18.018577	Wills-MBP.attlocal…	pki-goog.l.google.…	HTTP	448	GET /gts1c3/MFAwTjBMMEowSDA
7341	18.033813	pki-goog.l.google.…	Wills-MBP.attlocal…	OCSP	798	Response
9093	23.271086	Wills-MBP.attlocal…	pki-goog.l.google.…	HTTP	448	GET /gts1c3/MFAwTjBMMEowSDA
9097	23.286358	pki-goog.l.google.…	Wills-MBP.attlocal…	OCSP	799	Response
9823	24.595271	rr3.sn-q4fl6n6d.go…	Wills-MBP.attlocal…	HTTP	352	HTTP/1.1 204 No Content
9826	24.623313	Wills-MBP.attlocal…	rr3.sn-q4fl6n6d.go…	HTTP	110	Continuation
9829	24.640816	rr3.sn-q4fl6n6d.go…	Wills-MBP.attlocal…	HTTP	352	HTTP/1.1 204 No Content
9832	24.641257	Wills-MBP.attlocal…	rr3.sn-q4fl6n6d.go…	HTTP	110	Continuation

Figure 1 my first http captures when I first entered a site.

2369	7.189911	e.52ec1212bf18.spa…	Wills-MBP.attlocal…	TLSv1…	987	Application Data, Application Data
2370	7.190060	Wills-MBP.attlocal…	e.52ec1212bf18.spa…	TCP	86	62975 → https(443) [ACK] Seq=1132 Ack=1138876 Win=2046528 Len=0 TSval=2679333140 TSecr=1430597330
2371	7.495015	66:d2:48:30:4a:33	Spanning-tree-(for…	STP	52	Conf. Root = 0/0/60:d2:48:30:4a:32 Cost = 0 Port = 0x8003
2372	7.803212	Samsung.attlocal.n…	st-routers.mcast.n…	UDP	242	vcom-tunnel(8001) → vcom-tunnel(8001) Len=200
2373	7.803213	66:d2:48:30:4a:33	Broadcast	0x7373	121	Ethernet II
2374	7.803213	Samsung.attlocal.n…	192.168.1.255	UDP	77	40524 → 15600 Len=35
2375	8.417869	66:d2:48:30:4a:33	Spanning-tree-(for…	STP	52	Conf. Root = 0/0/60:d2:48:30:4a:32 Cost = 0 Port = 0x8003
2376	8.724333	66:d2:48:30:4a:33	Broadcast	0x7373	121	Ethernet II
2377	9.030908	SamsungElect_10:f6…	Broadcast	ARP	60	Who has 192.168.1.254? Tell 192.168.1.212
2378	9.135259	fe80::62d2:48ff:fe…	Wills-MBP.attlocal…	ICMPv6	86	Neighbor Solicitation for fe80::1c10:3dd9:59e0:9310 from 60:d2:48:30:4a:30
2379	9.135342	Wills-MBP.attlocal…	fe80::62d2:48ff:fe…	ICMPv6	78	Neighbor Advertisement fe80::1c10:3dd9:59e0:9310 (sol)
2380	9.141548	218.253.158.34.bc.…	Wills-MBP.attlocal…	TCP	77	http(80) → 62294 [PSH, ACK] Seq=1 Ack=1 Win=16 Len=11 TSval=193544038 TSecr=857858900
2381	9.141626	Wills-MBP.attlocal…	218.253.158.34.bc.…	TCP	66	62294 → http(80) [ACK] Seq=1 Ack=12 Win=2047 Len=0 TSval=857918800 TSecr=193544038
2382	9.438514	Wills-MBP.attlocal…	2606:4700:3035::68…	TCP	74	62971 → https(443) [ACK] Seq=1 Ack=1 Win=65535 Len=0
2383	9.454862	2606:4700:3035::68…	Wills-MBP.attlocal…	TCP	86	[TCP ACKed unseen segment] https(443) → 62971 [ACK] Seq=1 Ack=2 Win=8 Len=0 TSval=264727982 TSecr=
2384	9.496007	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	107	Standard query 0xe4ac PTR 218.253.158.34.in-addr.arpa
2385	9.520765	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	160	Standard query response 0xe4ac PTR 218.253.158.34.in-addr.arpa PTR 218.253.158.34.bc.googleusercont
2386	9.644827	66:d2:48:30:4a:33	Spanning-tree-(for…	STP	52	Conf. Root = 0/0/60:d2:48:30:4a:32 Cost = 0 Port = 0x8003
2387	9.645226	Samsung.attlocal.n…	st-routers.mcast.n…	UDP	242	vcom-tunnel(8001) → vcom-tunnel(8001) Len=200
2388	9.645308	66:d2:48:30:4a:33	Broadcast	0x7373	121	Ethernet II
2389	9.939668	Wills-MBP.attlocal…	ord31s22-in-x0e.1e…	TCP	74	62964 → https(443) [ACK] Seq=1 Ack=1 Win=2048 Len=0
2390	9.977033	ord31s22-in-x0e.1e…	Wills-MBP.attlocal…	TCP	86	[TCP ACKed unseen segment] https(443) → 62964 [ACK] Seq=1 Ack=2 Win=274 Len=0 TSval=1365875747 TSe
2391	9.999589	Wills-MBP.attlocal…	ec2-35-80-210-97.u…	TLSv1…	105	Application Data
2392	10.060512	ec2-35-80-210-97.u…	Wills-MBP.attlocal…	TCP	66	https(443) → 62604 [ACK] Seq=1 Ack=40 Win=116 Len=0 TSval=2054801040 TSecr=3069768039
2393	10.060513	ec2-35-80-210-97.u…	Wills-MBP.attlocal…	TLSv1…	118	Application Data
2394	10.060610	Wills-MBP.attlocal…	ec2-35-80-210-97.u…	TCP	66	62604 → https(443) [ACK] Seq=40 Ack=53 Win=2047 Len=0 TSval=3069768100 TSecr=2054801041
2395	10.495784	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x6d88 PTR 3.0.d.0.5.1.8.6.0.0.0.0.0.0.0.0.0.0.0.0.5.3.0.3.0.0.7.4.6.0.6.2.ip6.arpa
2396	10.495896	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0xa96e PTR e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.9.0.8.0.9.0.0.4.0.b.8.f.7.0.6.2.ip6.arpa
2397	10.496045	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	105	Standard query 0xf289 PTR 97.210.80.35.in-addr.arpa
2398	10.507005	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	191	Standard query response 0xa96e PTR e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.9.0.8.0.9.0.0.4.0.b.8.f.7.0.6.2.
2399	10.526445	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	214	Standard query response 0x6d88 No such name PTR 3.0.d.0.5.1.8.6.0.0.0.0.0.0.0.0.0.0.0.0.5.3.0.3.0.
2400	10.559273	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	167	Standard query response 0xf289 PTR 97.210.80.35.in-addr.arpa PTR ec2-35-80-210-97.us-west-2.comput
2401	10.566379	66:d2:48:30:4a:33	Spanning-tree-(for…	STP	52	Conf. Root = 0/0/60:d2:48:30:4a:32 Cost = 0 Port = 0x8003
2402	10.576219	Wills-MBP.attlocal…	2606:4700:3032::ac…	TCP	74	62589 → https(443) [ACK] Seq=1 Ack=1 Win=2048 Len=0
2403	10.589382	2606:4700:3032::ac…	Wills-MBP.attlocal…	TCP	86	[TCP ACKed unseen segment] https(443) → 62589 [ACK] Seq=1 Ack=2 Win=8 Len=0 TSval=2902711814 TSecr
2404	10.618622	Samsung.attlocal.n…	239.255.255.250	UDP	77	45440 → 15600 Len=35
2405	10.874552	66:d2:48:30:4a:33	Broadcast	0x7373	121	Ethernet II
2406	11.147834	Wills-MBP.attlocal…	ip-2a03-5640-f502-…	TLSv1…	397	Application Data
2407	11.147905	Wills-MBP.attlocal…	ip-2a03-5640-f502-…	TCP	1474	62925 → https(443) [ACK] Seq=2306 Ack=33541 Win=2590 Len=1388 TSval=4074565076 TSecr=2423699504 [T
2408	11.147957	Wills-MBP.attlocal…	ip-2a03-5640-f502-…	TLSv1…	1398	Application Data
2409	11.181126	SamsungElect_10:f6…	Broadcast	ARP	60	Who has 192.168.1.254? Tell 192.168.1.212
2410	11.194385	Wills-MBP.attlocal…	ip-2a03-5640-f502-…	TLSv1…	167	Application Data
2411	11.194477	Wills-MBP.attlocal…	ip-2a03-5640-f502-…	TCP	1474	62925 → https(443) [ACK] Seq=5087 Ack=33541 Win=2590 Len=1388 TSval=4074565122 TSecr=2423699504 [T

Frame 2379: 78 bytes on wire (624 bits), 78 bytes captured (624 bits) on interface en0, id 0

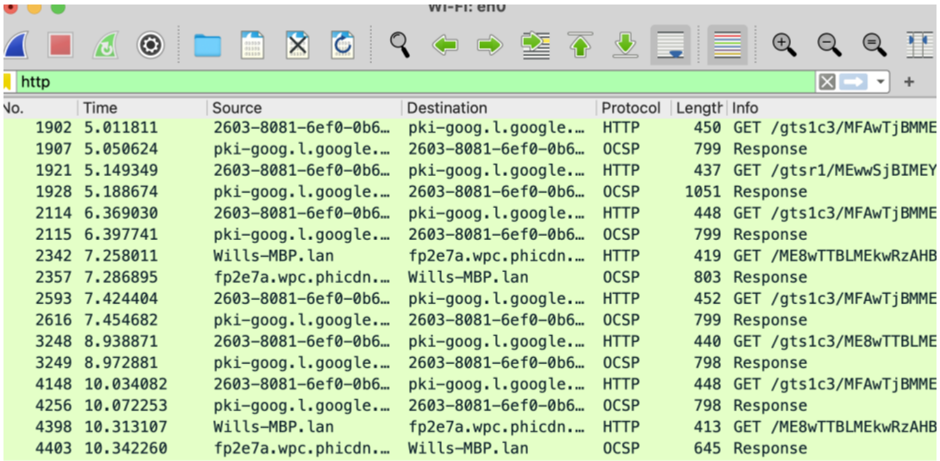
Figure 2 I logged in to my UTSA ASAP account and this is what I saw.

No.	Time	Source	Destination	Protocol	Length	Info
14	1.229426	Samsung.attlocal.n…	Wills-MBP.attlocal…	ARP	42	Who has 192.168.1.248?
15	1.229488	Wills-MBP.attlocal…	Samsung.attlocal.n…	ARP	42	192.168.1.248 is at a4
26	1.843140	Samsung.attlocal.n…	Broadcast	ARP	60	Who has 192.168.1.254?
1203	3.684808	Samsung.attlocal.n…	Broadcast	ARP	60	Who has 192.168.1.254?
1346	5.530808	SAB-WS01.attlocal.…	Broadcast	ARP	42	Who has 192.168.1.51?
1376	5.835732	Samsung.attlocal.n…	Broadcast	ARP	60	Who has 192.168.1.254?
1381	6.142661	SAB-WS01.attlocal.…	Broadcast	ARP	42	Who has 192.168.1.51?
1390	7.065662	SAB-WS01.attlocal.…	Broadcast	ARP	42	Who has 192.168.1.51?
1483	7.678785	Samsung.attlocal.n…	Broadcast	ARP	60	Who has 192.168.1.254?
2341	9.214025	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.253?
2342	9.214026	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.150?
2343	9.214090	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.85?
2344	9.214404	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.204?
2345	9.214735	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.64?
2346	9.214735	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.210?
2347	9.215131	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.249?
2348	9.215132	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.239?
2349	9.215132	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.213?
2350	9.215496	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.248?
2351	9.215501	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.251?
2352	9.215537	Wills-MBP.attlocal…	dsldevice.attlocal…	ARP	42	192.168.1.248 is at a4
2515	9.828779	Samsung.attlocal.n…	Broadcast	ARP	60	Who has 192.168.1.254?
4617	11.671673	Samsung.attlocal.n…	Broadcast	ARP	60	Who has 192.168.1.254?
5344	13.822921	Samsung.attlocal.n…	Broadcast	ARP	60	Who has 192.168.1.254?

Figure 3 ARP protocols captured.

16	1.351263	Wills-MBP.attlocal…	2606:4700:3037::68…	TCP	74	62977 → https(44
17	1.365055	2606:4700:3037::68…	Wills-MBP.attlocal…	TCP	86	[TCP ACKed unsee
32	2.750355	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TCP	78	57414 → https(44
34	2.765448	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	74	https(443) → 574
35	2.765907	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TCP	66	57414 → https(44
36	2.765909	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TLSv1…	583	Client Hello (SN
40	2.782207	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	66	https(443) → 574
49	2.793103	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TLSv1…	1466	Server Hello, Ch
50	2.793104	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	714	https(443) → 574
51	2.793105	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	1466	https(443) → 574
52	2.793105	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	714	https(443) → 574
53	2.793105	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	1466	https(443) → 574
54	2.793106	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	1466	https(443) → 574
55	2.793107	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TLSv1…	411	Application Data
56	2.794107	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TCP	66	57414 → https(44
57	2.799456	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TLSv1…	130	Change Cipher Sp
58	2.800875	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TLSv1…	448	Application Data
59	2.801159	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TLSv1…	1407	Application Data
61	2.814899	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TLSv1…	620	Application Data
62	2.814899	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TLSv1…	97	Application Data
63	2.815263	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TCP	66	57414 → https(44
64	2.815387	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TLSv1…	97	Application Data
65	2.820657	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	66	https(443) → 574
66	2.829690	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	66	https(443) → 574
75	2.857169	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TLSv1…	319	Application Data
76	2.857170	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TLSv1…	1097	Application Data
77	2.857170	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TLSv1…	105	Application Data
78	2.861440	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TCP	66	57414 → https(44
79	2.861707	Wills-MBP.attlocal…	96.10.190.35.bc.go…	TLSv1…	105	Application Data
80	2.880600	96.10.190.35.bc.go…	Wills-MBP.attlocal…	TCP	66	https(443) → 574
1344	5.266226	Wills-MBP.attlocal…	2606:4700:3033::68…	TCP	74	62968 → https(44
1345	5.281301	2606:4700:3033::68…	Wills-MBP.attlocal…	TCP	86	[TCP ACKed unsee
1408	7.505932	Wills-MBP.attlocal…	ec2-52-88-253-199.…	TCP	1514	62982 → https(44
1409	7.505934	Wills-MBP.attlocal…	ec2-52-88-253-199.…	TLSv1…	193	Application Data
1410	7.505966	Wills-MBP.attlocal…	ec2-52-88-253-199.…	TLSv1…	112	Application Data
1414	7.516856	Wills-MBP.attlocal…	www.googleapis.com	TCP	98	62985 → https(44
1416	7.530493	www.googleapis.com	Wills-MBP.attlocal…	TCP	94	https(443) → 629
1417	7.530575	Wills-MBP.attlocal…	www.googleapis.com	TCP	86	62985 → https(44
1418	7.530904	Wills-MBP.attlocal…	www.googleapis.com	TLSv1…	859	Client Hello (SN

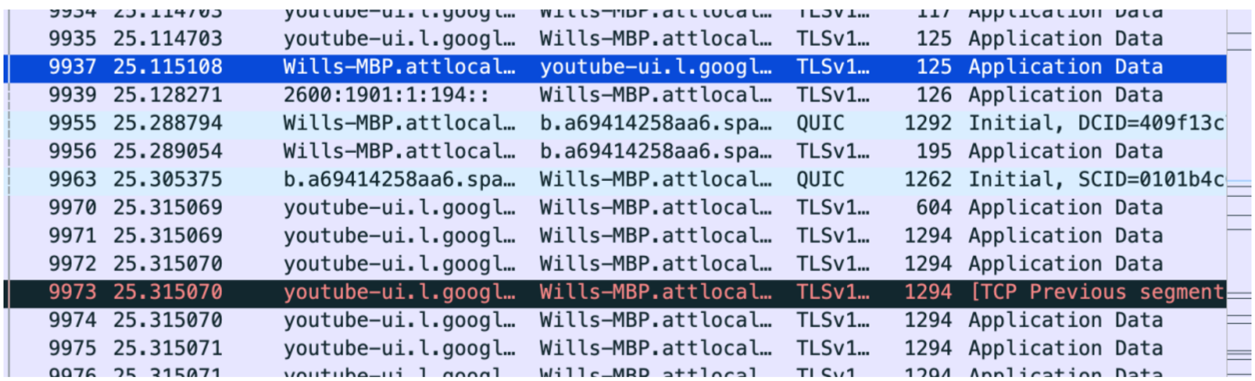
Figure 4 TCP protocols captured.



No.	Time	Source	Destination	Protocol	Length	Info
1902	5.011811	2603-8081-6ef0-0b6…	pki-goog.l.google.…	HTTP	450	GET /gts1c3/MFAwTjBMME
1907	5.050624	pki-goog.l.google.…	2603-8081-6ef0-0b6…	OCSP	799	Response
1921	5.149349	2603-8081-6ef0-0b6…	pki-goog.l.google.…	HTTP	437	GET /gtsr1/MEwwSjBIMEY
1928	5.188674	pki-goog.l.google.…	2603-8081-6ef0-0b6…	OCSP	1051	Response
2114	6.369030	2603-8081-6ef0-0b6…	pki-goog.l.google.…	HTTP	448	GET /gts1c3/MFAwTjBMME
2115	6.397741	pki-goog.l.google.…	2603-8081-6ef0-0b6…	OCSP	799	Response
2342	7.258011	Wills-MBP.lan	fp2e7a.wpc.phicdn.…	HTTP	419	GET /ME8wTTBLMEkwRzAHB
2357	7.286895	fp2e7a.wpc.phicdn.…	Wills-MBP.lan	OCSP	803	Response
2593	7.424404	2603-8081-6ef0-0b6…	pki-goog.l.google.…	HTTP	452	GET /gts1c3/MFAwTjBMME
2616	7.454682	pki-goog.l.google.…	2603-8081-6ef0-0b6…	OCSP	799	Response
3248	8.938871	2603-8081-6ef0-0b6…	pki-goog.l.google.…	HTTP	440	GET /gts1c3/ME8wTTBLME
3249	8.972881	pki-goog.l.google.…	2603-8081-6ef0-0b6…	OCSP	798	Response
4148	10.034082	2603-8081-6ef0-0b6…	pki-goog.l.google.…	HTTP	448	GET /gts1c3/MFAwTjBMME
4256	10.072253	pki-goog.l.google.…	2603-8081-6ef0-0b6…	OCSP	798	Response
4398	10.313107	Wills-MBP.lan	fp2e7a.wpc.phicdn.…	HTTP	413	GET /ME8wTTBLMEkwRzAHB
4403	10.342260	fp2e7a.wpc.phicdn.…	Wills-MBP.lan	OCSP	645	Response

.

Figure 5 HTTP protocols captured.



9935	25.114703	youtube-ui.l.googl…	Wills-MBP.attlocal…	TLSv1…	125	Application Data
9937	25.115108	Wills-MBP.attlocal…	youtube-ui.l.googl…	TLSv1…	125	Application Data
9939	25.128271	2600:1901:1:194::	Wills-MBP.attlocal…	TLSv1…	126	Application Data
9955	25.288794	Wills-MBP.attlocal…	b.a69414258aa6.spa…	QUIC	1292	Initial, DCID=409f13c
9956	25.289054	Wills-MBP.attlocal…	b.a69414258aa6.spa…	TLSv1…	195	Application Data
9963	25.305375	b.a69414258aa6.spa…	Wills-MBP.attlocal…	QUIC	1262	Initial, SCID=0101b4c
9970	25.315069	youtube-ui.l.googl…	Wills-MBP.attlocal…	TLSv1…	604	Application Data
9971	25.315069	youtube-ui.l.googl…	Wills-MBP.attlocal…	TLSv1…	1294	Application Data
9972	25.315070	youtube-ui.l.googl…	Wills-MBP.attlocal…	TLSv1…	1294	Application Data
9973	25.315070	youtube-ui.l.googl…	Wills-MBP.attlocal…	TLSv1…	1294	[TCP Previous segment
9974	25.315070	youtube-ui.l.googl…	Wills-MBP.attlocal…	TLSv1…	1294	Application Data
9975	25.315071	youtube-ui.l.googl…	Wills-MBP.attlocal…	TLSv1…	1294	Application Data

Figure 6 TLS protocols captured.

101	3.580526	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
102	3.580527	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
103	3.580527	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
104	3.580528	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
105	3.580529	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
106	3.580529	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
107	3.580530	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
108	3.580530	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
109	3.580531	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
110	3.580531	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
111	3.580877	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44
112	3.580895	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44
113	3.580973	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44
114	3.580992	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44
115	3.581062	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44
116	3.595021	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
117	3.595023	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
118	3.595024	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
119	3.595025	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
120	3.595025	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
121	3.595025	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
122	3.595026	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
123	3.595027	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
124	3.595027	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
125	3.595028	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
126	3.595029	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
127	3.595029	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
128	3.595030	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
129	3.595030	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
130	3.595031	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
131	3.595031	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
132	3.595032	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
133	3.595032	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
134	3.595033	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
135	3.595033	i.66a21586b342.spa…	Wills-MBP.attlocal…	UDP	1262	https(443) → 605
136	3.595318	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44
137	3.595337	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44
138	3.595379	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44
139	3.595450	Wills-MBP.attlocal…	i.66a21586b342.spa…	UDP	105	60596 → https(44

Figure 7 UDP protocols captured.

6	0.774191	fe80::62d2:48ff:fe…	ff02::1:ff5c:6318	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:c89c:a03f:4e5c:6318 from 60:d2:48:30:4a:30
7	0.774192	fe80::62d2:48ff:fe…	ff02::1:ff00:2a8c	ICMPv6	86	Neighbor Solicitation for fe80::18d5:a0c7:c700:2a8c from 60:d2:48:30:4a:30
8	0.774926	fe80::62d2:48ff:fe…	ff02::1:ff6e:25ab	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:ddf3:fb05:a76e:25ab from 60:d2:48:30:4a:30
9	0.774927	fe80::62d2:48ff:fe…	ff02::1:ff3d:540c	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:cfa:950:b53d:540c from 60:d2:48:30:4a:30
10	0.775010	Wills-MBP.attlocal…	fe80::62d2:48ff:fe…	ICMPv6	86	Neighbor Advertisement 2600:1700:1100:a150:ddf3:fb05:a76e:25ab (sol, ovr) is at a4:83:e7:53:
11	0.775053	Wills-MBP.attlocal…	fe80::62d2:48ff:fe…	ICMPv6	86	Neighbor Advertisement 2600:1700:1100:a150:cfa:950:b53d:540c (sol, ovr) is at a4:83:e7:53:be
15	0.794647	fe80::62d2:48ff:fe…	ff02::1:ff00:3b	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150::3b from 60:d2:48:30:4a:30
16	0.794670	fe80::62d2:48ff:fe…	ff02::1:ffe0:9310	ICMPv6	86	Neighbor Solicitation for fe80::1c10:3dd9:59e0:9310 from 60:d2:48:30:4a:30
17	0.794734	Wills-MBP.attlocal…	fe80::62d2:48ff:fe…	ICMPv6	86	Neighbor Advertisement 2600:1700:1100:a150::3b (sol, ovr) is at a4:83:e7:53:be:bf
18	0.794785	Wills-MBP.attlocal…	fe80::62d2:48ff:fe…	ICMPv6	86	Neighbor Advertisement fe80::1c10:3dd9:59e0:9310 (sol, ovr) is at a4:83:e7:53:be:bf
19	0.794916	fe80::62d2:48ff:fe…	ff02::1:ff00:22	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150::22 from 60:d2:48:30:4a:30
20	0.795451	fe80::62d2:48ff:fe…	ff02::1:ff10:f633	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:9e8c:6eff:fe10:f633 from 60:d2:48:30:4a:30
24	0.815282	fe80::62d2:48ff:fe…	ff02::1:ff10:f633	ICMPv6	86	Neighbor Solicitation for fe80::9e8c:6eff:fe10:f633 from 60:d2:48:30:4a:30
25	0.815284	fe80::62d2:48ff:fe…	ff02::1:ffbb:1a67	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:6010:8237:b0bb:1a67 from 60:d2:48:30:4a:30
26	0.815645	fe80::62d2:48ff:fe…	ff02::1:ff22:5867	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:31a3:e376:9522:5867 from 60:d2:48:30:4a:30
28	0.816270	fe80::62d2:48ff:fe…	ff02::1:ff59:297a	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:387a:8e59:59:297a from 60:d2:48:30:4a:30
31	0.835708	fe80::62d2:48ff:fe…	ff02::1:ff00:13	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150::13 from 60:d2:48:30:4a:30
32	0.836475	fe80::62d2:48ff:fe…	ff02::1:ff83:ab3e	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:6d5b:f208:2683:ab3e from 60:d2:48:30:4a:30
33	0.836475	fe80::62d2:48ff:fe…	ff02::1:ff22:5867	ICMPv6	86	Neighbor Solicitation for fe80::31a3:e376:9522:5867 from 60:d2:48:30:4a:30
35	0.836550	fe80::62d2:48ff:fe…	ff02::1:ffab:60a0	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:d81:a4b6:44ab:60a0 from 60:d2:48:30:4a:30
40	0.856206	fe80::62d2:48ff:fe…	ff02::1:ffe8:6b76	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:4b3:d569:ede8:6b76 from 60:d2:48:30:4a:30
42	0.856989	fe80::62d2:48ff:fe…	ff02::1:ff42:71bb	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:a0e0:945f:8d42:71bb from 60:d2:48:30:4a:30
43	0.856990	fe80::62d2:48ff:fe…	ff02::1:ff8a:3c2b	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:2db7:165d:468a:3c2b from 60:d2:48:30:4a:30
45	0.857854	fe80::62d2:48ff:fe…	ff02::1:ffbf:a832	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:d2c:b0a:73bf:a832 from 60:d2:48:30:4a:30
51	0.876884	fe80::62d2:48ff:fe…	ff02::1:ff28:14a7	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:65e3:f736:6628:14a7 from 60:d2:48:30:4a:30
53	0.876901	fe80::62d2:48ff:fe…	ff02::1:ff1e:ddec	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:24c1:9092:ce1e:ddec from 60:d2:48:30:4a:30
55	0.877611	fe80::62d2:48ff:fe…	ff02::1:ffb9:1d7	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:c858:5339:5db9:1d7 from 60:d2:48:30:4a:30
56	0.878408	fe80::62d2:48ff:fe…	ff02::1:ff26:6243	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:40c4:168c:8026:6243 from 60:d2:48:30:4a:30
64	0.897182	fe80::62d2:48ff:fe…	ff02::1:ffe2:2d4b	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:d99b:e858:e2e2:2d4b from 60:d2:48:30:4a:30
65	0.897905	fe80::62d2:48ff:fe…	ff02::1:ff23:c047	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:557e:6bfa:a323:c047 from 60:d2:48:30:4a:30
67	0.898215	fe80::62d2:48ff:fe…	ff02::1:ffc8:e581	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:b8d2:fdc0:a0c8:e581 from 60:d2:48:30:4a:30
69	0.898941	fe80::62d2:48ff:fe…	ff02::1:ff45:9ff2	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:dceb:890d:8245:9ff2 from 60:d2:48:30:4a:30
78	0.917797	fe80::62d2:48ff:fe…	ff02::1:ff15:11be	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:86:8261:7815:11be from 60:d2:48:30:4a:30
80	0.918458	fe80::62d2:48ff:fe…	ff02::1:ff28:4384	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:4937:edc0:8528:4384 from 60:d2:48:30:4a:30
82	0.919087	fe80::62d2:48ff:fe…	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150::1 from 60:d2:48:30:4a:30
83	0.919110	fe80::62d2:48ff:fe…	ff02::1:ffbf:3ec9	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:6500:c4bc:f4bf:3ec9 from 60:d2:48:30:4a:30
86	0.920728	fe80::62d2:48ff:fe…	ff02::1:ff3c:d659	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:92f8:2eff:fe3c:d659 from 60:d2:48:30:4a:30
87	0.920920	fe80::62d2:48ff:fe…	ff02::1:ff3c:d659	ICMPv6	86	Neighbor Solicitation for fe80::92f8:2eff:fe3c:d659 from 60:d2:48:30:4a:30
88	0.921114	fe80::62d2:48ff:fe…	ff02::1:ffc0:b185	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:cbe:523b:46c0:b185 from 60:d2:48:30:4a:30
89	0.921251	fe80::62d2:48ff:fe…	ff02::1:ff00:2a	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150::2a from 60:d2:48:30:4a:30
93	0.921993	fe80::62d2:48ff:fe…	ff02::1:ff31:c420	ICMPv6	86	Neighbor Solicitation for fe80::4b2:939:6d31:c420 from 60:d2:48:30:4a:30
94	0.922211	fe80::62d2:48ff:fe…	ff02::1:ff93:b47d	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:54d9:5031:ee93:b47d from 60:d2:48:30:4a:30
95	0.922614	fe80::62d2:48ff:fe…	ff02::1:ff8c:35aa	ICMPv6	86	Neighbor Solicitation for 2600:1700:1100:a150:39c0:97f5:cd8c:35aa from 60:d2:48:30:4a:30

Figure 8 ICMP protocols captured.

7	1.094713	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	102	Standard query 0x3096
8	1.094879	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	108	Standard query 0x8606
10	1.105498	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	140	Standard query respon
11	1.105499	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	136	Standard query respon
12	1.105499	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	108	Standard query respon
22	2.094411	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x9d27
23	2.094510	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x1875
24	2.094661	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	106	Standard query 0xfa7e
25	2.094888	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	104	Standard query 0x276b
26	2.094963	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	106	Standard query 0xb9b0
27	2.095045	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x04f2
29	2.101207	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	188	Standard query respon
30	2.101207	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	189	Standard query respon
31	2.101208	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	142	Standard query respon
32	2.101208	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	106	Standard query respon
49	2.136796	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	154	Standard query respon
61	2.249018	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	212	Standard query respon
70	3.095421	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x376a
71	3.095458	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x0172
72	3.095504	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	106	Standard query 0xa8a8
73	3.095594	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	105	Standard query 0x3cd2
74	3.095629	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	107	Standard query 0xbfd8
75	3.106252	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	152	Standard query respon
76	3.106253	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	163	Standard query respon
77	3.112890	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	161	Standard query respon
78	3.199172	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	167	Standard query respon
79	3.201224	dsldevice6.attloca…	Wills-MBP.attlocal…	DNS	212	Standard query respon
116	4.095901	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x19de
117	4.096007	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x0ed1
118	4.096313	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x101c
119	4.096462	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x7575
120	4.096594	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	107	Standard query 0x875f
121	4.096674	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	105	Standard query 0x2026
122	4.096784	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x8390
123	4.096911	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0x42e6
124	4.097000	Wills-MBP.attlocal…	dsldevice6.attloca…	DNS	152	Standard query 0xe85d

Figure 9 DNS protocols captured.

Hash (SHA256):	bf9d2aeed6aaab56ac54ed2834df1fab86f70f8d61c1ae75472d16b00
Hash (SHA1):	7dd5c9ed63c83baa8f4262501071d468476057e1
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2023-12-05 15:51:34
Last packet:	2023-12-05 16:57:09
Elapsed:	01:05:34

Capture

Hardware:	Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz (with SSE4.2)
OS:	macOS 13.3.1, build 22E261 (Darwin 22.4.0)
Application:	Dumpcap (Wireshark) 4.2.0 (v4.2.0-0-g54eedfc63953)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet si (snaplen)
Wi-Fi	0 (0.0%)	none	Ethernet	524288 l

Statistics

Measurement	Captured	Displayed	Marked
Packets	196409	196409 (100.0%)	—
Time span, s	3934.966	3934.966	—
Average pps	49.9	49.9	—
Average packet size, B	680	680	—
Bytes	133501458	133501458 (100.0%)	0
Average bytes/s	33 k	33 k	—
Average bits/s	271 k	271 k	—

Figure 10 1 hour report of network scan.

185…	220.875381	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.248? Tell 192.168.1.254
185…	220.875382	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.249? Tell 192.168.1.254
185…	220.875481	Wills-MBP.attlocal…	dsldevice.attlocal…	ARP	42	192.168.1.248 is at a4:83:e7:53:be:bf
185…	220.875775	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.250? Tell 192.168.1.254
187…	221.796764	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.251? Tell 192.168.1.254
187…	221.796765	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.252? Tell 192.168.1.254
187…	221.796791	dsldevice.attlocal…	Broadcast	ARP	60	Who has 192.168.1.253? Tell 192.168.1.254
187…	227.699619	dsldevice.attlocal…	Wills-MBP.attlocal…	ARP	60	192.168.1.254 is at 60:d2:48:30:4a:30

Figure 11 ARP protocols with MAC and IP addresses.

Home Network Diagram

Figure 12 ISP Modem

Figure 13 Router

Figure 14 Laptop

Name:	/var/folders/03/8bvv7cgj361fgmfqpvqp82_w0000gn/T/wireshark_Wi-FiH07ZE2.pcapng
Length:	251 MB
Hash (SHA256):	7eaa6419cad844c0b91b4fa9c49f4b57644f0e00120d63c86e729d34078c7d03
Hash (SHA1):	a440715e0dc8865c9af9854bd5f368d273f8eced
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2023-12-05 17:56:09
Last packet:	2023-12-05 21:56:59
Elapsed:	04:00:49

Capture

Hardware:	Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz (with SSE4.2)
OS:	macOS 13.3.1, build 22E261 (Darwin 22.4.0)
Application:	Dumpcap (Wireshark) 4.2.0 (v4.2.0-0-g54eedfc63953)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit (snaplen)
Wi-Fi	0 (0.0%)	none	Ethernet	524288 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	393135	393135 (100.0%)	—
Time span, s	14449.975	14449.975	—
Average pps	27.2	27.2	—
Average packet size, B	607	607	—
Bytes	238559308	238559308 (100.0%)	0
Average bytes/s	16 k	16 k	—
Average bits/s	132 k	132 k	—

Figure 15 4-hour network scan.

Bibliography

Ashwani, K. (2023, September 22). *What is wireshark and use cases of Wireshark?*. DevOpsSchool.com. https://www.devopsschool.com/blog/what-is-wireshark-and-use-cases-of-wireshark/

Ot, A. (2021, October 18). *What is wireshark and how can you use it to secure your network?*. MUO. https://www.makeuseof.com/what-is-wireshark/

Porup, J. (2018, September 17). *What is wireshark? what this essential troubleshooting tool does and how to use it*. CSO Online. https://www.csoonline.com/article/566309/what-is-wireshark-what-this-essential-troubleshooting-tool-does-and-how-to-use-it.html

What is Wireshark. Chapter 1. introduction. (n.d.). https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html