Jay Offerdahl
EECS 565
Mini-Project 3

Network Sniffing

After downloading and installing Wireshark, I captured and analyzed a few of my desktop computer's packets. I was able to see the different types of protocols used with my traffic (DNS, MDNS, NBNS, SSDP, UDP, etc.). I thought it was interesting that I was getting UDP calls on a very frequent basis, until I realized that I had a voice chat client open called DiscordApp. This voice chat client, after doing some research, uses UDP for both receiving and transmitting of voice (source). During the same time, I was listening to music using Google Play uses both TCP and UDP, so it was comforting to see both of these protocols being used here. I've included a few screenshots of the traffic that I sniffed from my desktop below:

200	2.808106	192.168.1.155	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
201	3.415984	173.199.89.147	192.168.1.155	UDP	247 51643 → 59669 Len=205
202	3.416136	173.199.89.147	192.168.1.155	TLSv1.2	155 Application Data
203	3.444523	192.168.1.155	173.199.89.147	TCP	54 49945 → 443 [ACK] Seq=1 Ack=817 Win=254 Len=0
204	3.445166	173.199.89.147	192.168.1.155	UDP	233 51643 → 59669 Len=191
205	3.450664	173.199.89.147	192.168.1.155	UDP	232 51643 → 59669 Len=190
206	3.470486	173.199.89.147	192.168.1.155	UDP	221 51643 → 59669 Len=179
207	3.492166	173.199.89.147	192.168.1.155	UDP	236 51643 → 59669 Len=194
208	3.511932	173.199.89.147	192.168.1.155	UDP	244 51643 → 59669 Len=202
209	3.535178	173.199.89.147	192.168.1.155	UDP	222 51643 → 59669 Len=180
210	3.553514	173.199.89.147	192.168.1.155	UDP	245 51643 → 59669 Len=203
211	3.581833	173.199.89.147	192.168.1.155	UDP	228 51643 → 59669 Len=186
212	3.594170	173.199.89.147	192.168.1.155	UDP	220 51643 → 59669 Len=178
213	3.611735	173.199.89.147	192.168.1.155	UDP	221 51643 → 59669 Len=179
214	3.632458	173.199.89.147	192.168.1.155	UDP	211 51643 → 59669 Len=169
215	3.653372	173.199.89.147	192.168.1.155	UDP	202 51643 → 59669 Len=160
216	3.658030	192.168.1.155	192.168.1.255	UDP	305 54915 → 54915 Len=263
217	3.680690	173.199.89.147	192.168.1.155	UDP	206 51643 → 59669 Len=164
218	3.690302	173.199.89.147	192.168.1.155	UDP	202 51643 → 59669 Len=160
219	3.714640	173.199.89.147	192.168.1.155	UDP	203 51643 → 59669 Len=161
220	3.735968	173.199.89.147	192.168.1.155	UDP	239 51643 → 59669 Len=197
221	3.751635	173.199.89.147	192.168.1.155	UDP	237 51643 → 59669 Len=195
222	3.776496	173.199.89.147	192.168.1.155	UDP	227 51643 → 59669 Len=185
223	3.792921	192.168.1.155	173.199.89.147	TLSv1.2	115 Application Data
224	3.795934	173.199.89.147	192.168.1.155	UDP	226 51643 → 59669 Len=184
225	3.810099	173.199.89.147	192.168.1.155	UDP	224 51643 → 59669 Len=182
226	3.837702	173.199.89.147	192.168.1.155	UDP	217 51643 → 59669 Len=175
227	3.848119	173.199.89.147	192.168.1.155	TLSv1.2	111 Application Data
228	3.852870	173.199.89.147	192.168.1.155	UDP	243 51643 → 59669 Len=201
229	3.877222	192.168.1.155	173.199.89.147	TCP	54 49945 → 443 [ACK] Seq=62 Ack=874 Win=254 Len=0

2859	52.995247	192.168.1.155	64.233.176.189	QUIC	82 Payload (Encrypted), CID: 6004904073370160793, Seq: 140
2860	53.292567	192.168.1.155	173.199.89.147	TLSv1.2	115 Application Data
2861	53.345765	173.199.89.147	192.168.1.155	TLSv1.2	111 Application Data
2862	53.373192	192.168.1.155	173.199.89.147	TCP	54 49945 → 443 [ACK] Seq=1064 Ack=9239 Win=250 Len=0
2863	53.657618	192.168.1.155	192.168.1.255	UDP	305 54915 → 54915 Len=263
2864	53.877063	64.233.176.189	192.168.1.155	QUIC	82 Payload (Encrypted), Seq: 147
2865	53.902885	192.168.1.155	64.233.176.189	QUIC	79 Payload (Encrypted), CID: 6004904073370160793, Seq: 141
2867	54.665510	192.168.1.155	192.168.1.255	UDP	305 54915 → 54915 Len=263
2868	54.770254	192.168.1.1	239.255.255.250	SSDP	305 NOTIFY * HTTP/1.1
2869	54.770432	192.168.1.1	239.255.255.250	SSDP	377 NOTIFY * HTTP/1.1
2870	54.770547	192.168.1.1	239.255.255.250	SSDP	373 NOTIFY * HTTP/1.1
2871	54.770677	192.168.1.1	239.255.255.250	SSDP	353 NOTIFY * HTTP/1.1
2872	54.770805	192.168.1.1	239.255.255.250	SSDP	385 NOTIFY * HTTP/1.1
2873	54.770928	192.168.1.1	239.255.255.250	SSDP	367 NOTIFY * HTTP/1.1
2874	54.771061	192.168.1.1	239.255.255.250	SSDP	369 NOTIFY * HTTP/1.1
2875	54.771191	192.168.1.1	239.255.255.250	SSDP	369 NOTIFY * HTTP/1.1
2876	55.660113	192.168.1.155	192.168.1.255	UDP	305 54915 → 54915 Len=263
2878	56.669235	192.168.1.155	192.168.1.255	UDP	305 54915 → 54915 Len=263

Next, I told Wireshark to sniff my laptop's packets, which I had previously connected via Ethernet to my desktop computer. Once Wireshark was running, I opened up the internet and searched

Jay Offerdahl
EECS 565
Mini-Project 3

for a few pages, which produced a massive amount of LLMNR, NBNS, and SSDP protocol uses, among others. These protocols were used for standard queries, name queries, and searches over http. In other cases, there were Standard query responses, which came after the standard queries, which makes sense. I've included a few screen shots below of the output I was receiving from Wireshark.

9767	259.116535	169.254.90.218	224.0.0.22	IGMPv3	60 Membership Report / Leave group 224.0.0.252
9770	259.116857	169.254.90.218	224.0.0.22	IGMPv3	60 Membership Report / Join group 224.0.0.252 for any sources
9772	259.117537	169.254.90.218	224.0.0.252	LLMNR	75 Standard query 0x7c14 ANY DESKTOP-E3DI9MK
9778	259.189897	169.254.90.218	224.0.0.22	IGMPv3	60 Membership Report / Join group 224.0.0.252 for any sources
9780	259.268586	169.254.90.218	224.0.0.251	MDNS	82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
9782	259.268766	169.254.90.218	224.0.0.251	MDNS	60 Standard query response 0x0000
9783	259.268766	169.254.90.218	224.0.0.251	MDNS	82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
9784	259.268889	169.254.90.218	224.0.0.251	MDNS	60 Standard query response 0x0000
9798	259.528398	169.254.90.218	224.0.0.252	LLMNR	75 Standard query 0x7c14 ANY DESKTOP-E3DI9MK
9800	259.632857	169.254.90.218	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
9811	259.794985	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9812	259.794986	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9813	259.794986	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9814	259.795176	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9815	259.795177	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9816	259.795177	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9817	259.795177	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9818	259.795177	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9819	259.795364	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9827	260.132876	169.254.90.218	224.0.0.251	MDNS	60 Standard query response 0x0000
9830	260.133043	169.254.90.218	224.0.0.251	MDNS	60 Standard query response 0x0000
9832	260.535490	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
9833	260.535760	169.254.90.218	224.0.0.252	LLMNR	64 Standard query 0xfcef A wpad
9838	260.633689	169.254.90.218	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1

10722	302.860420	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
10723	302.862451	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
10724	302.862451	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
10725	302.863449	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
10727	303.368816	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
10730	304.135496	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
10731	304.135678	169.254.90.218	224.0.0.252	LLMNR	64 Standard query 0xc61c A wpad
10734	304.135957	169.254.90.218	224.0.0.252	LLMNR	64 Standard query 0xc0a7 AAAA wpad
10736	304.545732	169.254.90.218	224.0.0.252	LLMNR	64 Standard query 0xc61c A wpad
10737	304.545733	169.254.90.218	224.0.0.252	LLMNR	64 Standard query 0xc0a7 AAAA wpad
10740	304.885711	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
10741	305.546773	169.254.90.218	224.0.0.251	MDNS	82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
10743	305.547007	169.254.90.218	224.0.0.251	MDNS	60 Standard query response 0x0000
10745	305.549908	169.254.90.218	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
10747	305.637618	169.254.90.218	169.254.255.255	NBNS	92 Name query NB WPAD<00>
10750	306.547599	169.254.90.218	224.0.0.251	MDNS	82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
10752	306.547792	169.254.90.218	224.0.0.251	MDNS	60 Standard query response 0x0000
10754	306.550485	169.254.90.218	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
10757	307.551721	169.254.90.218	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1
10760	308.548619	169.254.90.218	224.0.0.251	MDNS	82 Standard query 0x0000 PTR _googlecast._tcp.local, "QM" question
10762	308.548955	169.254.90.218	224.0.0.251	MDNS	60 Standard query response 0x0000
10764	308.552581	169.254.90.218	239.255.255.250	SSDP	216 M-SEARCH * HTTP/1.1

I found it interesting that all of my UDP and TCP packets were encrypted, but this makes sense since they carry sensitive data, even if it's just music or my friend's voices. When looking at SSDP protocol for either HTTP requests or responses, I was able to see some information stored in the packet. For example, on an HTTP ok response, I found that the internet gateway device's uuid was 60049043-8df9-4073-9a3e-72723d61a8d3, as well as other information.

However, like I said earlier, most of the application data that I encountered was encrypted as to protect the data from people like me who might be sniffing a network. This is comforting, but at the same time alarming. I do not like the fact that I was able to see these packets so effortlessly, and I hope I can someday learn enough to be able to defend against malicious attackers with this type of access. In a final discussion, I was definitely astounded to see just how many different protocols were being used, and how quickly they were firing off in this project.