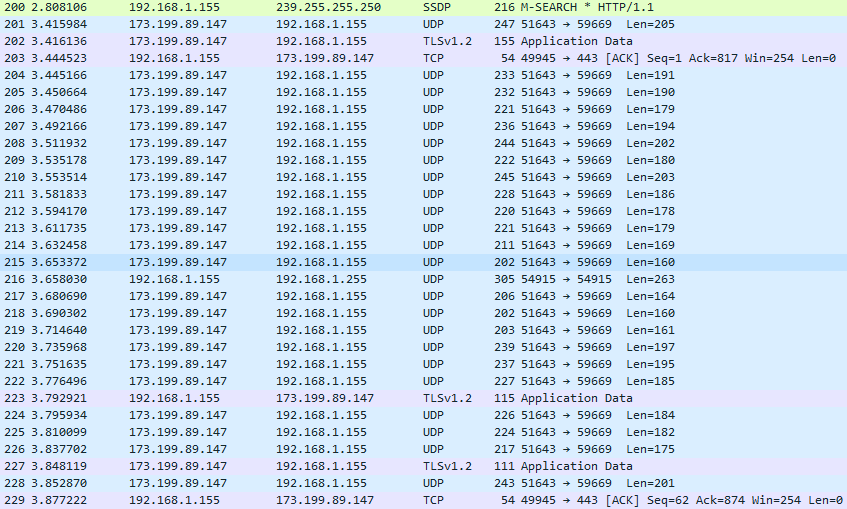
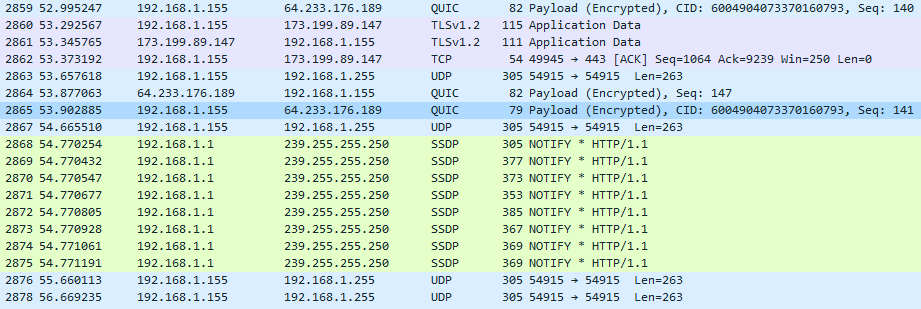
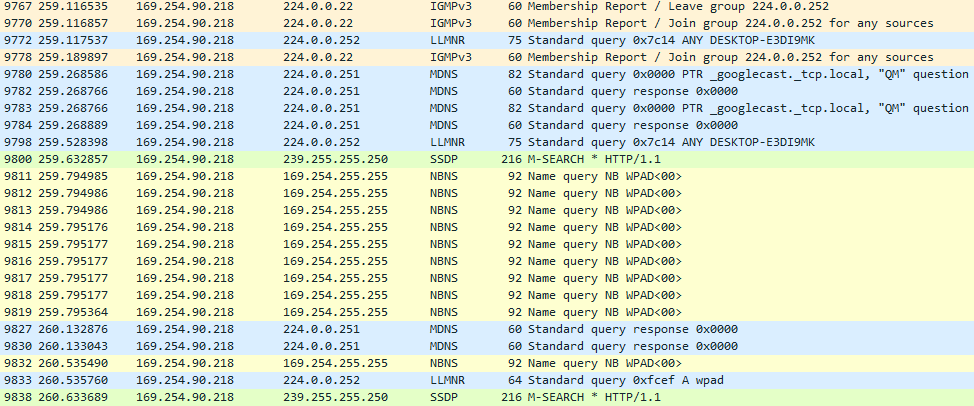
Network Sniffing

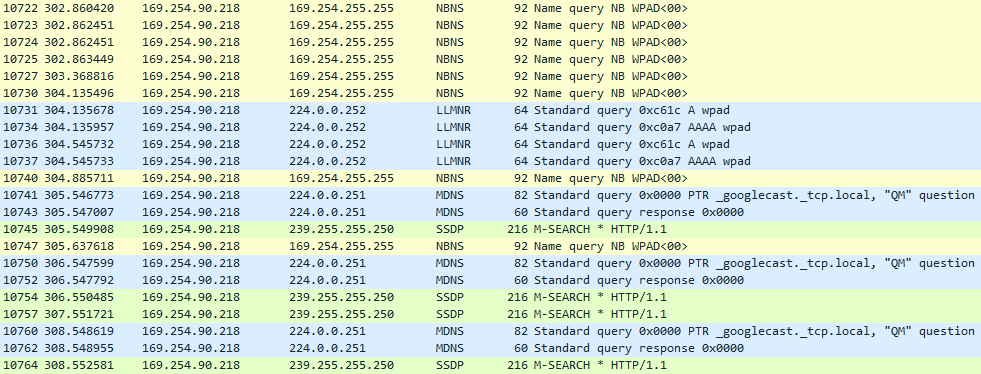
After downloading and installing Wireshark, I captured and analyzed a few of my desktop computer’s packets. I was able to see the different types of protocols used with my traffic (DNS, MDNS, NBNS, SSDP, UDP, etc.). I thought it was interesting that I was getting UDP calls on a very frequent basis, until 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](https://discordapp.com/developers/docs/topics/voice-connections)). 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:





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 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.





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.