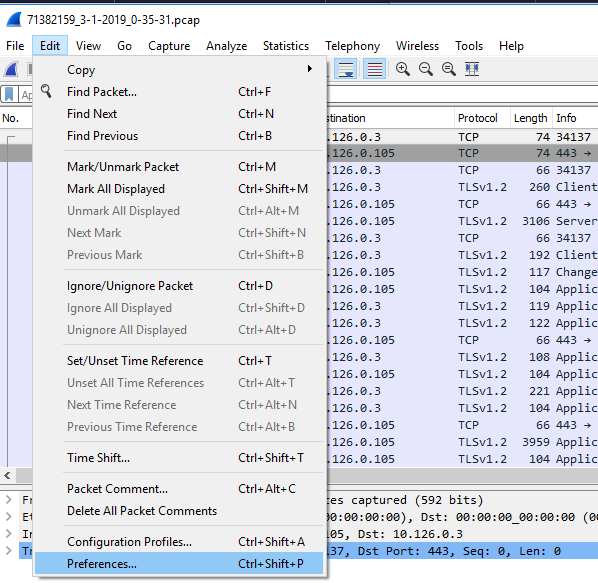
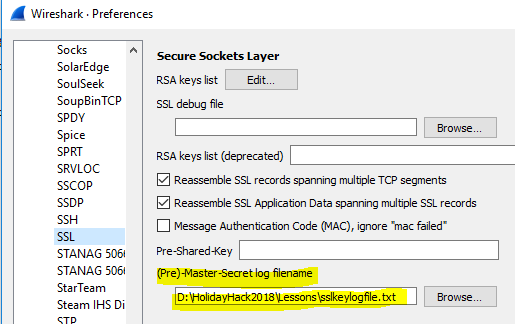
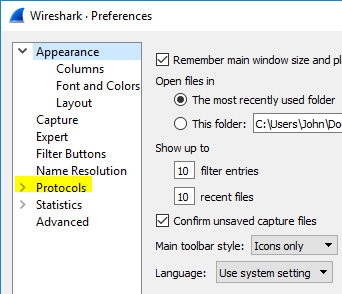
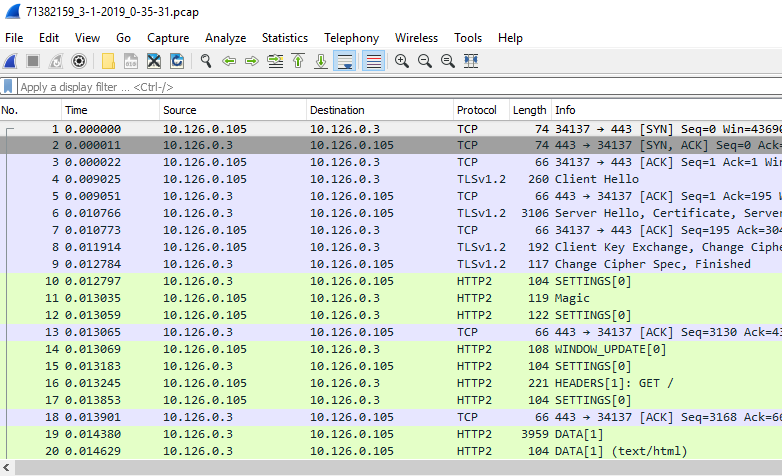
Objective--Network Traffic Forensics (Part 4)

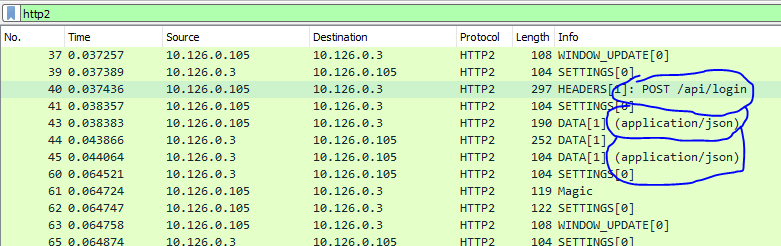
# Solution (or part of it)

Now that we have sslkeylogfile.txt, or whatever you named it, we can decrypt the packet capture that the Packalyzer server gives us. Be sure to tell Packalyzer to sniff, then download the packet, and then use <https://packalyzer.kringlecastle.com/dev/packalyzer_clientrandom_ssl.log/> to download the key file. Don’t wait too long between taking the packet capture and downloading the key file; otherwise they won’t match, and the traffic will not be decrypted.

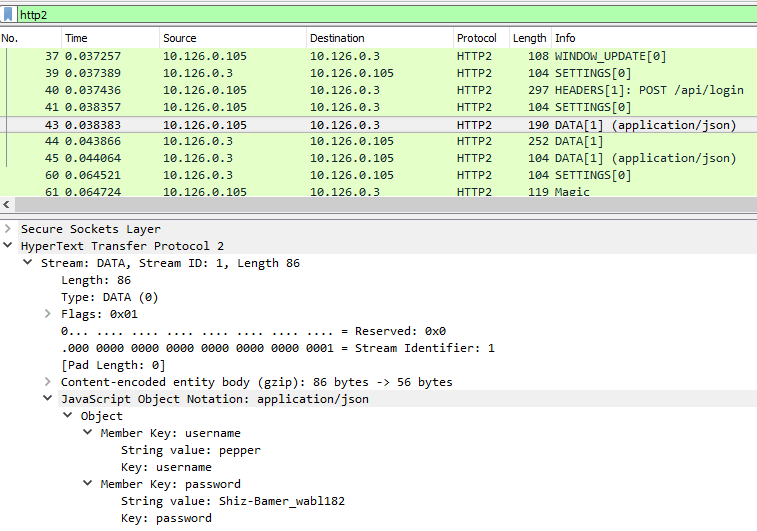
When you open the pcap file from Packalyzer, you will see that the traffic is all encrypted by TLS 1.2. As in Chris’ video, we need to edit Wireshark’s preferences to include the key file. Select Edit > Preferences…  


Then select protocols and scroll down to SSL.  


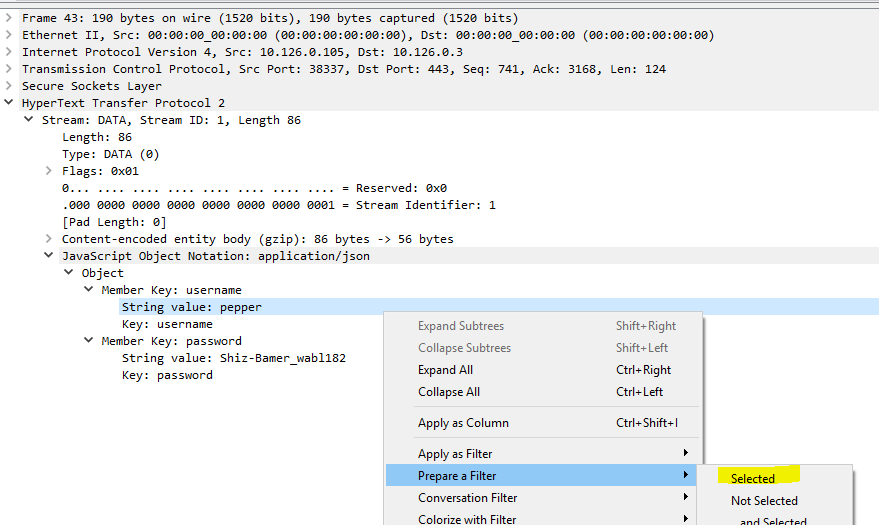
With SSL selected, insert the path to the SSLKEYLOGFILE we found. The traffic will magically be decrypted. Remember, this only works because the browser (or other application) was recording the keys it used. We could not have decrypted the traffic just by intercepting it.  


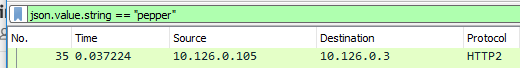
As we examine the traffic using the http2 Display Filter that Chris showed us, we do see something of interest. There is a Header with POST /api/login. Remember that HTTP/2 puts the headers and data into different frames.  


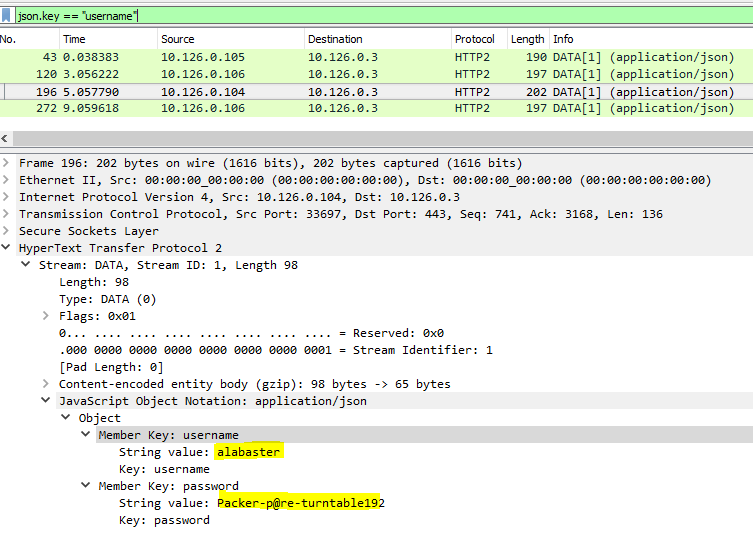
Remember that Follow > TCP Stream will just show us the encrypted traffic. Follow > SSL Stream is better in that shows the decrypted traffic, but it is still in gzip, so we cannot read it. The HTTP/2 section of the data pane is much better.

When we look at the DATA frames with application/json data, some of them bring joy to an attacker’s heart.  


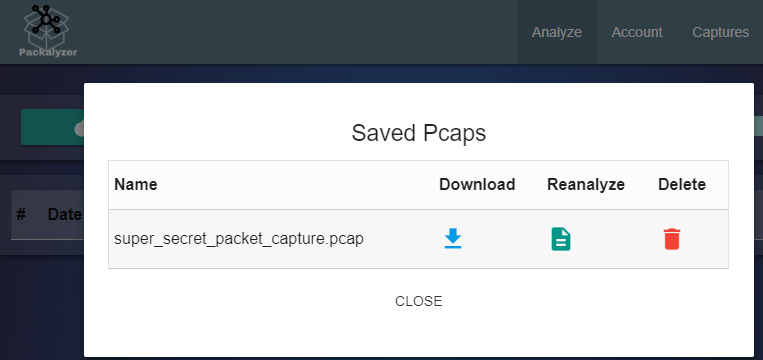
So, Pepper’s credentials are pepper and Shiz-Bamer\_wabl182. Cool.

The display filter Chris gave us, “http2.data.data && http2 contains username” does not work here. I’m not sure why, but perhaps we can make our own filter. The Wireshark feature that creates display filters when you right-click > Prepare a Filter > Selected is very powerful. Some of the elements in the JSON data don’t allow it, but the String value for pepper allows it.  


That gives us a display filter. 

That helps, but we want something that shows use all the packets that have a username (or password would do.) After some fiddling, I arrived at the display filter json.key==username.  


There are four packets that contain credentials, and one of them has Alabaster’s.  
Perhaps Alabaster’s credentials will get us into Packalyzer.

Look at the pcap Alabaster has stored! We are getting close to the end.  


# Hand In

Download the super\_secret\_packet\_capture.pcap file and discover its secrets. You will have to extract a file from an SMTP attachment. Once you do, you can answer the question: What is the song that Alabaster and Holly are discussing? Thankfully, the packet capture is plain text SMTP.

1. How is the attached file encoded?
2. How did you extract the file from the SMTP stream?
3. What is the name of the song?