Objective--Network Traffic Forensics (Part 1)

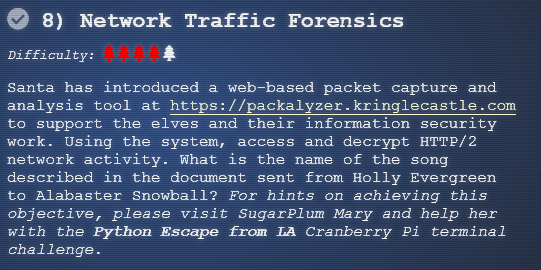
# What you can learn from this

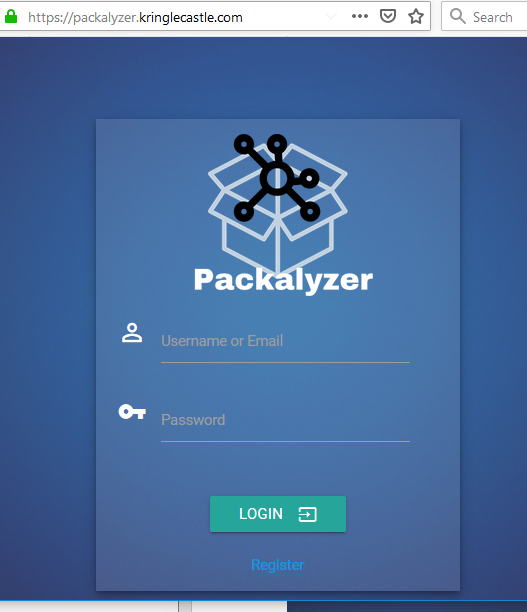
Most of the examples you see in textbooks and Wireshark packet captures are of HTTP 1.1. As you saw in Chris Elgee and Chris Davis’ talk, [HTTP/2: Because 1 Is the Loneliest Number](http://www.youtube.com/watch?v=PC6-mn9g9Cs), most major sites now use HTTP/2 because it is much more efficient.

One reason we don’t see more HTTP/2 is that it is almost always encrypted. If you want to view encrypted web traffic from your own browser for troubleshooting or analysis, Firefox and Chrome both save the pre-master keys that you need to decrypt the traffic. Wireshark can use these keys to display the decrypted traffic to you. This talk, [HTTP/2 Decryption and Analysis in Wireshark](https://www.youtube.com/watch?v=YHOnxlQ6zec), by Chris Davis explains how it works.

It is essential that an IT security professional be able to decrypt HTTP/2. Also, Chris’ talk will let you know what you should be looking for, to solve this objective. Watch the talk now.

# Getting Started

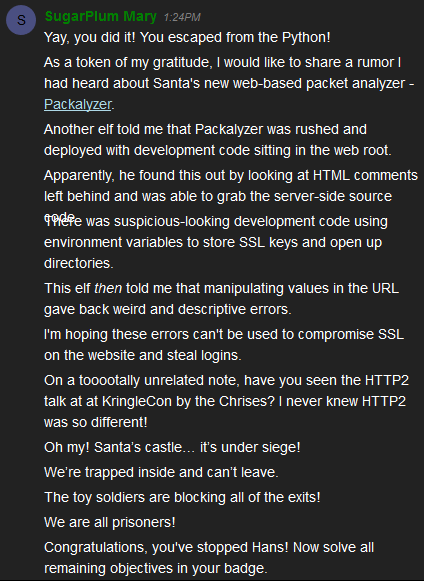
For this challenge, you’ll be working with Santa’s new site, <https://packalyzer.kringlecastle.com/>.   


Create an account for yourself and log in. I’ve heard that the registration page only likes lower case letters.  


Once you can log in, you can take packet captures and download them. If you attended Chris’ talk, you’ll know that you are missing a file, though.

Much of this challenge will involve trying to get the packalyzer site to give you the file you need.

# Hints

The talk in the badge hint is the one we mentioned before, [HTTP/2 Decryption and Analysis in Wireshark](https://www.youtube.com/watch?v=YHOnxlQ6zec), by Chris Davis. Without that you won’t know what to look for. Take careful note of Mary’s comments about comments, environment variables that expose directories, and weird descriptive errors from the URL.  


# Hand In

1. Look for a one-line comment in the HTML index that mentions a file name that might contain source code.
2. Find that file using your browser. There aren’t too many directories you have to look in.