Very very very Hidden

Description:

Finding a flag may take many steps, but if you look diligently it won't be long until you find the light at the end of the tunnel. Just remember, sometimes you find the hidden treasure, but sometimes you find only a hidden map to the treasure. try_me.pcap.

- 1. Open the file with a pcap analyzer like Wireshark.
- When I checked the HTTP traffic, I found two requests for PNG files: duck.png and evil_duck.png. I downloaded these files by following these steps: File --> Export Objects --> HTTP --> Save All. I tried using "zsteg" and other steganography tools, but I couldn't reveal anything.
- 3. I decided to look at the other HTTP requests and the sites that were visited. I searched for this in Wireshark using the filter: (http.request or ssl.handshake.type==1). I found that many sites were visited, including GitHub, Amazon, Microsoft, and finally, PowerShell.org. To be honest, I have doubts about the last one, "PowerShell.org."

Look at this:

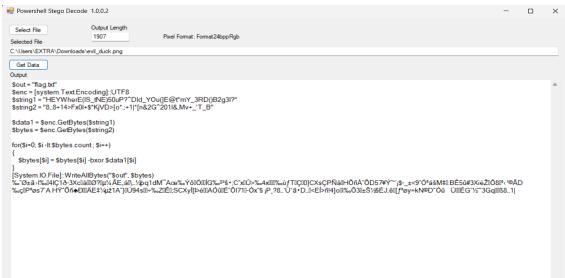
```
444/ 106.3/2312 192.168.1.189 52.114.135.165 1L5V1.2 5/1 Client Helio (SNI=presence.teams.microsoft.com)
4584 110.441210 192.168.1.189 54.147.39.126 HTTP 546 6ET /NothingSus/evil_duck.png HTTP/1.1
4695 110.459289 192.168.1.189 65.55.44.19 TL5V1.2 571 Client Helio (SNI=presence.teams.microsoft.com)
6481 124.387379 192.168.1.189 40.126.7.100 TL5V1.2 74 Client Helio (SNI=login.microsoftonline.com)
6540 124.899317 192.168.1.189 52.113.194.132 TL5V1.2 630 Client Helio (SNI=login.microsoftonline.com)
6572 125.199643 192.168.1.189 144.259.77.206 QUIC 192 Initial, DCIDmea3366912836653, PKNI: , CRYPTO, PADDING
6740 128.043344 192.168.1.189 104.212.2183 TL5V1.3 571 (Zint Helio (SNI=login.microsoft.com)
6749 128.043344 192.168.1.189 104.212.2183 TL5V1.3 571 (Zint Helio (SNI=login.microsoft.com)
```

After downloading "evil_duck.png," there is a handshake with the PowerShell.org site.

Hmm, this made me think about the PowerShell script on GitHub called Invoke-PSImage. This script hides data within an image's pixels (more information on GitHub).

So, I searched for a tool that could extract data hidden by Invoke-PSImage, and I found this: https://github.com/PCsXcetra/Decode_PS_Stego. It's a Windows tool that can help us extract the hidden data.

After running the tool and opening the PNG file with it, we found the following script:



Just xor string1 and string2 and you get the flag : $picoCTF\{n1c3_job_f1nd1ng_th3_s3cr3t_in_the_im@g3\}\;.$