Eliana Weinstein

An Encrypted Kidnapping- Implementation and How to Solve

Wireshark (Network)

- Player needs to look through the wire shark capture and find the image titled “thailand.jpg”. They should find it by using key words from the story.
- 

- This image hints that they will need to use DNS enumeration.

Integration with Operating System Features

- Player needs to analyze the image, hidden in the image will be a domain and a bunch of subdomains.
- I added a hidden comment within the photo using terminal on my mac. I used the command - sudo exiftool -Comment="Domain: google.com. Secret Subdomains: amp, api, web, download, mail, ns3, support" ~/Downloads/thailand.jpg
- **Domain: google.com**
    - **Subdomains:**
        - **amp, api, web, download, mail, ns3, support**
- The player can find this comment on terminal by using the command exiftool -Comment ~/Downloads/thailand.jpg

DNS enumeration and Python

- Player now needs to write up a DNS enumeration python code so they can find the ip addresses of the subdomains listed in the image.
- They should find these results:
    - **amp - 142.250.75.78**
    - **api - 142.250.75.68**
    - **web - 142.250.75.78**
    - **download - 142.250.75.68**
    - **mail - 142.251.37.69**
    - **ns3 - 216.239.36.10**
    - **support - - 142.251.37.78**

- They should use the hint from the photo - “The union of the last 4 digits of each IP address might “**shift**” your perspective of this message”, and realize that they need to combine the last 4 digits of each ip address into one large number **75787568757875683769361037778.**
- From here they should determine that they need to **shift** each letter of the text on the image by the amount on that number with a hash algorithm
- (I will attach code i used to hash message, and code user needs to write to dehash it)

Hashing

- Player should write hash algorithm (I attached), and if done correctly, “oy{xz?57prn}y3iwp6g8P~nN: =” will become “https://imgur.com/a/MxmN7z6”
- The link takes you to this image- the map!
-

- The player now has a substitution map
- **substitution_map = {**
- **'P':'z', 'h': 'X', 't': 'L', 'p': 'R', 's': 'M', ':': 'Y', '/': 'Q', 'w': 'K', '.': ',',**
- **'y': 'F', 'o': 'N', 'u': 'B', 'b': 'C', 'e': 'W', 'r': 'A', 'l': 'D', 'n': 'I',**
- **'T': 'V', 'k': 'O', 'B': 'i', 'Q': 'S', 'c': 'G', 'x': 'U', '-': 'E', 'H': 'H', '8': '7'**
- **}**
- They should now use the hint from the story that the map will help decrypt the message on the strange device. They now know they need to write a decryption algorithm to decrypt the encrypted message.
- The message:

- 

- XLLRMYQQKKK,FNBLBCW,GNmQKaLGX?v=VOizSGFEHU7
- (I will attach code i used to encrypt message, and code user needs to write to decrypt it)

Encryption

- User will now Substitution Encryption to decrypt the message
- **Substitution encryption** is a type of encryption where each character in the plaintext is replaced by another character according to a predefined **substitution map**. The substitution map defines how each character is substituted for another character in the alphabet or character set.
- **Substitution Map**: A **substitution map** is a one-to-one mapping between the original characters (plaintext) and the substituted characters (ciphertext)
- **“XLLRMYQQKKK,FNBLBCW,GNmQKaLGX?v=VOizSGFEHU7**” will become **https://www.youtube.com/watch?v=TkBPQcy-Hx8**
- They will receive a link to a youtube video (private, can only access with the link)
- 

- The video is a creepy AI generated guy in a mask speaking in a distorted voice saying the coordinates of where the friend is located.

The End!