Forensics Report

Lakshit Verma

August 15, 2024

1 LIT CTF — Kuwahara

Given is a bunch of files, an encoded image and a noisy image.

The file `encoder.py` isn't really important, it just contains a bunch of helper functions to simplify the algorithm.

`main.py` has the code we're interested in. It describes how the Kuwahara algorithm is supposed to work and gives us a template to implement its decryption.

Things I learned in this:

1. The working of the Kuwahara denoising algorithm.
2. Reversing custom image steganography.

Kuwahara Algorithm

This splits an image into several square windows and performs a simple algorithm on them.

1. Split the window into 4 quadrants.
2. Calculate the arithmetic mean $\bar{x}$ and std deviation σ.
3. Set the center pixel in the window to the $\bar{x}$ of the quadrant with the smallest variance.

In the files, we're given code and asked to implement the Kuwahara Algorithm. We're also given an encryption function using our Kuwahara implementation, and asked to make the corresponding decryption function.
The encoding process is as follows:

1. Get the $\bar{x}$ and σ for all windows in the image, with a window size of 5x5.
2. Find the quadrant with the smallest variance.
3. Now from $(x, y) = (window\_size, window\_size)$ for each character m in the encoding message.
   (a) Replace the pixel at $(x, y$ with the $(m-1)th$ smallest variant quadrant's mean.
   (b) Increment y.
   (c) Increment x and set $y = window\_size$ only if $y = image\_width - window\_size$.

To decode the message, we simply reverse this algorithm. And once we encounter a pixel the same as one processed with normal Kuwahara, we can be sure the message has ended and exit. The end message is first encoded into ASCII and base-3 and we need to decode this too.
Completing all these steps, we get our flag.

2 CTFZone — Losing Information

Looked at this challenge after the CTF ended. The hints around finding deleted info led me to the volatility MFT parser plugin. Using it, however I wasn't able to find any relevant info.
I also extracted a `.pcapng` file from the `filescan` dump, but it contained nothing of note, just encrypted TLS traffic.