**Digital Forensic Analysis**

**Title –** Capture the Flag / Hidden Message Extraction from Image Files

**Author –** Miguel Sanchez

**Summary of analysis**

A digital forensic investigation is required to review 18 digital photographs that may potentially have encrypted messages encoded within them. While the images appear to not be tampered with in any way, we will be leveraging the least significant bit method to confirm if there are embedded messages within each photo. The least significant bit method is one of the most popular ways to implant data within an image with very little to no impact to the quality of the image. While reviewing the large population of photographs that needed to be analyzed, the digital forensic team decided to use python scripts along with Steganography module within python to automate the review of each photo and confirm if there are hidden messages within each image. This process would allow us to expand this analysis to even larger population of files if needed.

**Objectives**

The goal of this analysis is to identify if there is any encrypted data within the population of 18 digital images. If any of the images provided have encrypted data, the digital forensic team must extract these messages, document each image and its data and provide the methodologies and practices used to obtain and validate the information for the investigation.

**Chain of Custody**



**Methods and Tools Used**

When identifying the best approach to tackle this investigation. Our digital forensics team decided to use a widely used programming language called Python. Python has a massive library of modules such as Stegano which is used solely for the purpose of providing steganalysis. The Python code is not only efficient, accurate, widely adopted but also scalable to further expand the investigation population if needed. It was identified through the digital forensic analysis that all photographs had hidden messages embedded within them.

Python Code Used for Investigation

!pip install stegano

from stegano import lsb

image\_paths = [

'/content/Knight\_modified\_Tarrant.png',

'/content/LanaBracken\_buffalo\_modified.png',

'/content/Palace\_Modified\_Collier.png',

'/content/cornfield\_kuehler\_modified.png',

'/content/Sanchez.png',

'/content/Sunset-Mountains\_modified.png',

'/content/PointMuguBurkett\_modified.png',

'/content/correa.png',

'/content/f16\_modified\_Yang.png',

'/content/fox\_modified\_tanquerido.png',

'/content/iron\_FUDALA\_modified.png',

'/content/maroon\_bells\_modified\_by\_wang.png',

'/content/sadcat\_modified.png',

'/content/secret\_sunset\_JAGDALE.png',

'/content/simplyhired\_modified\_Kennady.png',

'/content/sunset\_Dupree.png',

'/content/universe\_modified - Noroozi.png',

'/content/stars\_modified\_Mayilsamy.png'

]

for i, image\_path in enumerate(image\_paths, start=1):

secret\_message = lsb.reveal(image\_path)

print(f"The secret message in image {i} is: {secret\_message}")

The above code has each individual image file that was uploaded into the Steganographic python code which further extracts each individual hidden message that is encrypted within each one.

Link to Google Colab notebook - https://colab.research.google.com/drive/1T\_Pf6pfrgqxj1gUMC3wW5e2ULIYoIBeK?usp=sharing

**Relevant Findings**

By using steganographic software through python coding we identified that each image file had been manipulated to include hidden messages. Each file was meticulously analyzed through the python code while leveraging the Stegano import lsb function. This method of steganography has been widely adopted by digital forensic teams and professionals throughout the world.

A list of information on a computer

Description automatically generated with medium confidence