# Forensic Activity #1.7 – Steganalysis

The purpose of this activity is to get you familiarized and experienced with a couple of Steganography tools and techniques. The first part of the activity is to be completed during the demonstration and the second part, in your Zoom breakout session.

## Steganalysis Techniques

* *Known Message Attack*
  + Uses comparative analysis to decipher the message when the secret message is already known.
* *Chosen Stego Attack*
  + Used when the steganography tool and stego-media were used to hide content. The steganography tool is known and a password/passphrase is either known or password recovery techniques are used.
* Known Cover Attack
  + Used when the cover-media (i.e. original file) with no hidden message is available and the stego-media (i.e. steogified) file is available for comparison.

## Steganalysis Tools

* [Stegosuite](https://stegosuite.org/) used to hide information in an image file
  + Use the ‘Show Appliations’ button in the left menu bar to list programs on SIFT
  + Type stegosuite in the filter box and double-click the application icon
* [Steg](http://www.fabionet.org/content/description-and-features) used to hide and recover information in image files.
  + Requires extracted data to be saved in a new location. Use the folder ‘StegData’ in the ‘SuspectFiles’ folder.
  + Access the program with the following commands. *Note: the terminal running Steg must remain open.*
    - **cd Downloads/Steg-1.1.0.0-Linux64**
    - **./Steg-1.1.0.0-x64.AppImage**
* [Steghide](http://steghide.sourceforge.net/documentation.php) – command line tool used to hide and extract hidden data from various file types

### Getting Started

* From Canvas Download the StegDemoInClass folder
* Use the ‘Show Applications’ button in the left menu bar to list programs on SIFT
* Type stegosuite in the filter box and double-click the application icon

Stegosuite: GUI tool for hiding and extracting text from an image.

1. As the malicious user open the **Linux3 file** from **StegDemoInClass folder**
2. Then type in a message, then click embed.
3. Back in the StegDemo folder, find the new file and rename it.
4. Take note of the files properties
5. Open the **LinuxOriginal.jpg** file

## Stegosuite Analysis

On your own, you’ll perform a chosen stego attack, based on information in Table A to provide a description of your steganalysis results. Use Google to find the correct password to extract the embedded data.

*Make take a moment for stegosuite to fully load the image, before the buttons become available.*

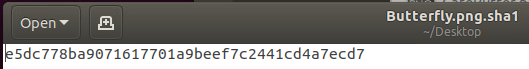
|  |  |
| --- | --- |
| **#A** | |
| **Steganalysis Technique** | **Chosen Stego Analysis** |
| **Tool:** | **Stegosuite** |
| **Password to supply:** | *“What year did Northern Kentucky State College become Northern Kentucky University?”* |
| **Stego Media File:** | **HACKED.png** |
| **Visual or Brief description of results** | **Graphical user interface  Description automatically generated** |

Steg: GUI tool for hiding and extracting embedded content

1. Access the program with the following commands. *Note: the terminal running Steg must remain open.*

* **cd Downloads/Steg-1.1.0.0-Linux64**
* **./Steg-1.1.0.0-x64.AppImage**

1. In Steg
   1. Select File, then open generic image à
   2. Navigate to **CatepillarOriginal.png** in the StegDemoInClass folder
2. Take of the space Steg calculated
3. In Steg
   1. Select Hide, “hide data”
   2. Navigate to **butterfly.png** in the StegDemoInClass folder
4. In Steg
   1. Select Hide, the “Save” with a new file name in the default location or in the Demo folder
5. Close the CatepillarOriginal file, then open the new stegofied file created in step 5.
6. Select “Extract”, then “Extract Data”
7. Save the extracted file to the Desktop
   1. ~/Desktop
8. View the extracted image file and the associated sha1 hash.



Steg Analysis: Use Steg to complete Table B with the image listed.

|  |  |
| --- | --- |
| **#B** |  |
| **Steganalysis Technique** | **Chosen Steg Attack** |
| **Tool:** | Steg |
| **Password:** | n/a |
| **Stego Media File:** | MicrosoftLogo.png |
| **Visual or Brief description of results** | **Graphical user interface, text, website  Description automatically generated** |

Steghide: command-line tool for hiding and extracting data

1. cd to the StegDemoInClass folder
2. Issue the following command to view file properties
   1. **ls -l Friend\***
3. Issue the following command to start the embedding process
   1. **steghid embed -cf Friendship1.jpg -ef TEST.png**
   2. When asked for a passphrase you can either hit Enter to leave blank or add a phrase
      1. *-cf = cover file*
      2. *-ef = embedded file*
4. Issue an ls -l to view the new stats of Friendship1.jpg
5. Extract the data just embed with the following command
   1. **steghid extract -sf – Frindship1.jpg**
      1. *Provide a passphrase if necessary*
      2. -sf = stegofile

### Steghide Analysis: Complete Table C after extracting and examining the data from above

|  |  |
| --- | --- |
| **#C** |  |
| **Steganalysis Technique** | Chosen Stego Attack |
| **Tool:** | steghide |
| **Password:** | enter |
| **Stego Media File:** | Friendship2.jpg |
| **Visual or Brief description of results** | **Birthday popper** |

# Teams

For the remaining part of the class, you’ll break up into teams and do some Steganalysis in your groups using the tools above on a set of files.

**Some things to be aware of:**

* Not every file will have embedded data
* Not every file is password protected, but those that are use the same password as HACKED.png
* Not every file is embedded with an image. Some programs allow non-image file types
* You’ll end up with 3 files that have been stegofied.

Open the **SuspectImageFiles** folder to get started and enteryour results in the tables below:

|  |  |
| --- | --- |
| **#1** |  |
| **Stego media File Name** | **NKU\_CyberLab.jpg** |
| **Stego media File Type** | **.jpg** |
| **Is the original cover media file available?** | **no** |
| **Steganalysis Technique** | *Chosen Stego Attack* |
| **Tool:** | **steghide** |
| **Password:** | **N/A** |
| **Extracted data filename:** | **CCs.txt** |
| **Brief description of extracted data** | **Credit card name type, names, card number, SSN,** |
| How did you determine this file contained embedded data? Briefly, describe the Steganalysis process used. | **After trying the steg, stegosuite, the final option was to try steghide in which we were able to try different passwords to find the embedded data. The file has embedded info when the CCs.txt was extracted.** |

|  |  |
| --- | --- |
| **#2** |  |
| **Stego media File Name** | **Griffin\_hall\_NKU2.jpg** |
| **Stego media File Type** | **.jpg** |
| **Is the original cover media file available?** | **No** |
| **Steganalysis Technique** | *Chosen Stego Attack* |
| **Tool:** | **steghide** |
| **Password:** | **1976** |
| **Extracted data filename:** | **Rockyou2.txt** |
| **Brief description of extracted data** | **Passwords for brute force attacks** |
| How did you determine this file contained embedded data? Briefly, describe the Steganalysis process used. | **I tried using the steghide first since that was the last answer I found and then tried the password in which it resulted me with the rockyou2.txt file with the brute force passwords.** |

|  |  |
| --- | --- |
| **#3** |  |
| **Stego media File Name** | **Norse1.png** |
| **Stego media File Type** | **.png** |
| **Steganalysis Technique** | *Chosen Stego Attack* |
| **Tool:** | **steg** |
| **Password:** | **N/A** |
| **Extracted data filename:** | **Norse2.png** |
| **Brief description of extracted data** | **A new Norse background**  **Graphical user interface, application  Description automatically generated** |
| How did you determine this file contained embedded data? Briefly, describe the Steganalysis process used. | **We determined this by using the steg source and extracting this data. We were able to find a new image under Norse2.png** |