

Lab 9: Steganography

Thangamuthu Balaji

I. INTRODUCTION

THIS is the lab report for Lab 9: Steganography of the course, Cyber Defense Competitions where the students completed tasks that covered the topic of Image Steganography. This lab served to help students learn about various tools that could be used to detect steganography and how to use them to retrieve possible hidden information.

II. USING DOMINICBREUKER'S STEGO-TOOLKIT REPOSITORY TO ANALYZE IMAGE 1.SECRET AND IMAGE 2.SECRET

The process of hiding information in non-secret material, such as pictures or files, is called steganography. The goal of this lab is to utilize a variety of forensic and steganography methods to reveal concealed information in two supplied photographs (1.secret and 2.secret). The objective is to assess how well various instruments and techniques identify, extract, and decipher hidden material. Watermarking, espionage, and cybersecurity all make extensive use of steganography. Knowledge of its methods and defenses has practical uses in information security and digital forensics. Our first assignment was to use Dominic Breuker's stego-toolkit repository, a Docker image helpful for resolving steganography issues, to discover any secrets concealed in image 1.secret. Numerous well-known tools and screening scripts are pre-installed on the image, which you may use to verify basic items.

I started with cloning the DominicBreuker's stego-toolkit repository from git and then using the docker container to start the project. I uploaded the images that I wanted to analyze in the data folder in the repository.

I used the following tools for my analysis:

file - Check out what kind of file you have.

exiftool - Check out metadata of media files.

binwalk - Check out if other files are embedded/appended.

strings - Check out if there are interesting readable characters in the file.

foremost - Carve out embedded/appended files.

pngcheck - Get details on a PNG file (or find out if it is actually something else).

identify - GraphicMagick tool to check what kind of image a file is. Checks also if image is corrupted.

zsteg - Detects various LSB stego, also openstego and the Camouflage tool. LSB stands for Least Significant Bit. The idea behind LSB embedding is that if we change the last bit value of a pixel, there won't be much visible change in the color. For example, 0 is black. Changing the value to 1 won't make much of a difference since it is still black, just a lighter shade. [2]

III. FINDINGS

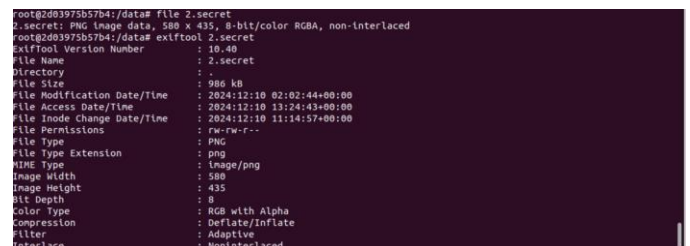
- 1) **file:** The files 1.secret and 2.secret were identified to be PNG images.



```

5(pwd)/data:/data_dominicbreuker/stego-toolkit/bin/bash
root@2d03975b57b4:/data# ls
1.secret 2.secret
root@2d03975b57b4:/data# file 1.secret
1.secret: PNG image data, 855 x 502, 8-bit/color RGBA, non-interlaced
root@2d03975b57b4:/data#
  
```

Fig. 2. 1.secret file



```

root@2d03975b57b4:/data# file 2.secret
2.secret: PNG image data, 580 x 435, 8-bit/color RGBA, non-interlaced
root@2d03975b57b4:/data# exiftool 2.secret
Exiftool Version Number       : 10.40
File Name                     : 2.secret
File Size                     : 986 KB
Directory                     :
File Modification Date/Time    : 2024:12:10 02:02:44+00:00
File Access Date/Time         : 2024:12:10 13:24:43+00:00
File Inode Change Date/Time   : 2024:12:10 11:14:57+00:00
File Permissions               : -rwxr-x--
File Type                     : PNG
File Type Extension           : png
MIME Type                     : image/png
Image Width                   : 580
Image Height                  : 435
Bit Depth                     : 8
Color Type                    : RGB with Alpha
Compression                   : Deflate/Inflate
Filter                        : Adaptive
Interlace                     : Noninterlaced
  
```

Fig. 3. 2.secret file

- 2) exiftool: This tool provided with a variety of image-related metadata for both photos, including profile details, color qualities, and dimensions (855 x 502).

3) binwalk: The image 1.secret's copyright was identified as "Copyright (c) 1998 Hewlett-Packard Company" by the binwalk utility. Nevertheless, it also exposed several MySQL files that were contained in the picture 2.secret.

- 4) strings: The strings tool assisted in determining if each image included any legible text. The picture

1.secret had a few strings that contained the data we gleaned from the picture and metadata. 2. Secret appeared to lack any logical strings.

pngcheck, foremost and identify: These were some more additional tools that I tested on the image 1.secret.

```

root@2d03975b57b4:/data# exiftool 1.secret
ExifTool Version Number      : 10.40
File Name                    : 1.secret
Directory                   : .
File Size                   : 1683 kB
File Modification Date/Time  : 2024:12:10 02:02:40+00:00
File Access Date/Time       : 2024:12:10 11:15:28+00:00
File Inode Change Date/Time  : 2024:12:10 11:14:11+00:00
File Permissions             : rw-rw-r--
File Type                   : PNG
File Type Extension         : png
MIME Type                   : image/png
Image Width                 : 855
Image Height                : 502
Bit Depth                   : 8
Color Type                  : RGB with Alpha
Compression                 : Deflate/Inflate
Filter                      : Adaptive
Interlace                   : NonInterlaced
Gamma                      : 2.2
Profile Name                : ICC profile
Profile CMH Type            : Lino
Profile Version              : 2.1.0

u have      file stego.jpg

```

Fig. 4. 1.secret exiftool

```

root@2d03975b57b4:/data# file 2.secret
2.secret: PNG image data, 580 x 435, 8-bit/color RGBA, non-interlaced
root@2d03975b57b4:/data# exiftool 2.secret
ExifTool Version Number      : 10.40
File Name                    : 2.secret
Directory                   : .
File Size                   : 986 kB
File Modification Date/Time  : 2024:12:10 02:02:44+00:00
File Access Date/Time       : 2024:12:10 13:24:43+00:00
File Inode Change Date/Time  : 2024:12:10 11:14:57+00:00
File Permissions             : rw-rw-r--
File Type                   : PNG
File Type Extension         : png
MIME Type                   : image/png
Image Width                 : 580
Image Height                : 435
Bit Depth                   : 8
Color Type                  : RGB with Alpha
Compression                 : Deflate/Inflate
Filter                      : Adaptive
Interlace                   : NonInterlaced

```

Fig. 5. 2.secret exiftool

```

root@2d03975b57b4:/data# binwalk 1.secret
DECIMAL      HEXADECEMAL  DESCRIPTION
-----
0            0x0      PNG image, 855 x 502, 8-bit/color RGBA, non-interlaced
421          0x1A5      Copyright string: "Copyright (c) 1998 Hewlett-Packard Company"
root@2d03975b57b4:/data#

```

Fig. 6. 1.secret binwalk

```

root@2d03975b57b4:/data# binwalk 2.secret
DECIMAL      HEXADECEMAL  DESCRIPTION
-----
0            0x0      PNG image, 580 x 435, 8-bit/color RGBA, non-interlaced
737          0x2E1      MySQL ISAM index file Version 1
897          0x381      MySQL ISAM index file Version 1
1409         0x581      MySQL ISAM index file Version 1
1425         0x591      MySQL ISAM index file Version 1
1441         0x5A1      MySQL ISAM index file Version 1
2114         0x5298      MySQL ISAM compressed data file Version 6
23449        0x5899      MySQL ISAM index file Version 3
23641        0x5C59      MySQL ISAM index file Version 5
25786        0x64BA      MySQL ISAM compressed data file Version 4
27498        0x686A      MySQL ISAM index file Version 2
44359        0xA047      MySQL ISAM index file Version 5
56252        0xDBBC      MySQL ISAM compressed data file Version 6
58605        0xE4ED      MySQL ISAM index file Version 3
73732        0x1204      MySQL ISAM index file Version 1
82808        0x14378     MySQL ISAM index file Version 5
103137       0x192E1     MySQL ISAM compressed data file Version 7

```

Fig. 7. 2.secret binwalk

```

g1RL
bTRC
text
Copyright (c) 1998 Hewlett-Packard Company
desc
sRGB IEC61966-2.1
sRGB IEC61966-2.1
KVZ
KVZ
KVZ
KVZ
desc
IEC http://www.iec.ch
IEC http://www.iec.ch
desc
-IEC 61966-2.1 Default RGB colour space - sRGB
-IEC 61966-2.1 Default RGB colour space - sRGB
desc
,Reference Viewing Condition in IEC61966-2.1
,Reference Viewing Condition in IEC61966-2.1

```

Fig. 8. 1.secret strings

```

root@2d03975b57b4:/data# strings 2.secret
IHDR
IDATx
xkLA
5675
/*/*
TJAncZ
oZWaVv
KGD+&
/+*2.-
[Xkee
9+(4.-
933824
2+-50-
-63:63
NIF[V$
-5/70(
<5+3*!
2+11*
!uOC<6
34/C66

```

Fig. 9. 2.secret strings

```

root@2d03975b57b4:/data# foremost 1.secret
Processing: 1.secret
[*]
root@2d03975b57b4:/data# pngcheck 1.secret
OK: 1.secret (855x502, 32-bit RGB+alpha, non-interlaced, -0.4%).
root@2d03975b57b4:/data# identify -verbose 1.secret
Image: 1.secret
Format: PNG (Portable Network Graphics)
Geometry: 855x502
Class: DirectClass
Type: grayscale
Depth: 8 bits-per-pixel component
Channel Depths:
  Gray: 8 bits
  Opacity: 1 bits
Channel Statistics:
  Gray:
    Minimum: 0.00 (0.0000)
    Maximum: 65535.00 (1.0000)
    Mean: 43199.70 (0.6592)
    Standard Deviation: 26255.19 (0.4006)
  Opacity:

```

Fig. 10. 1.secret pngcheck foremost identify tools

mainly does statistical tests, but only in straightforward situations can it disclose hidden meanings. However, if it discovers intriguing inconsistencies, it could offer clues about what to hunt for.

The text "gc7W[*](C)" was discovered in image 1.secret, while other patterns and files were discovered in image 2.secret.

```

root@2d03975b57b4:/data# zsteg -a 1.secret
b1,r,lsb,VX,prime ... text: "gc7W[*](C"
root@2d03975b57b4:/data#

```

Fig. 11. 1.secret zsteg

```

root@2d03975b57b4:/data# zsteg -a 2.secret
Image data:
  02,r,lsb,xy: text: "b1\rh\|SRU"
  02,r,lsb,xy: text: "ws3337ws?"
  02,g,lsb,xy: text: "((((____KXKK')/"
  02,g,lsb,xy: text: "))))----"
  02,g,msb,xy: text: "0000xxxx--"
  02,b,msb,xy: text: "----NN\\|\\r"
  02,bgr,lsb,xy: text: ["N" repeated 12 times]
  02,a,msb,xy: text: "wgvhwgC"
  02,bgr,lsb,xy: text: "b1RuRu1.-"
  02,g,msb,xy: text: "7675757?"
  04,a,msb,xy: text: "73737373"
  06,a,msb,xy: file: MPEG ADTS, layer III, v2, 112 kbps, Monaural
  06,abgr,msb,xy: file: MPEG ADTS, layer III, v2, 24 kbps, Monaural
  07,b,lsb,xy: file: AIX core file fulldump 32-bit, \002\375
  07,a,lsb,xy: file: MPEG ADTS, layer III, v2, 160 kbps, Monaural
  07,abgr,lsb,xy: file: MPEG ADTS, layer III, v2, 160 kbps, Monaural
  08,a,lsb,xy: file: MPEG ADTS, layer II, v1, Monaural
  08,abgr,lsb,xy: file: MPEG ADTS, layer II, v1, Monaural
  01,rgb,lsb,xy,prime: file: MPEG ADTS, AAC, v2 LC, 48 kHz, stereo + center
  01,rgb,msb,xy,prime: file: compacted data
  01,bgr,lsb,xy,prime: file: MPEG ADTS, layer III, v2, 16 kbps, 22.05 kHz, JntStereo

```

Fig. 12. 2.secret zsteg

IV. CONCLUSION

I can say that I've finished all of the lab assignments successfully. This lab gave participants practical experience with steganography tools, demonstrating both their advantages and disadvantages in terms of revealing secret data. For straightforward situations, technologies like Binwalk and Zsteg worked well, but advanced methods.

It confirmed if the image is a PNG image, checked for any embedded files and checked if the image was corrupted in any way.

- 2) **zsteg**: This was a tool designed to detect steganography

REFERENCES

- [1] "Wstego-toolkit" [github.com
https://github.com/DominicBreuker/wstego-toolkit](https://github.com/DominicBreuker/wstego-toolkit)
- [2] "LSB based Image steganography using
MAT- LAB" [geeksforgeeks.org
https://www.geeksforgeeks.org/lsb-based-
image-steganography-using-matlab/](https://www.geeksforgeeks.org/lsb-based-image-steganography-using-matlab/)