

Lab 11

Chetan Sahrudhai Kimidi - ckimidi

Abstract—Learning about steganography and finding flags/secrets in images, using related tools.

I. INTRODUCTION

In this lab, I learnt what steganography is, how I can use it to encode or decode messages/secrets/flags into media like images, here, PNG files.

II. 1.SECRET

First, I followed the instructions as given in the assignment, and learnt about Dominic's stego toolkit, [1]. I used "**docker pull dominicbreuker/stego-toolkit**" and obtained the image into my VM. Then, I downloaded both 1.secret and 2.secret from the files on Canvas, and loaded/mounted them into a container, while starting/initializing it, as seen in Fig. 1.

Now, as suggested in the assignment hints, I learnt about various tools in the toolkit, such as file, binwalk, strings, exiftool, zsteg, foremost, stegdetect etc. I played with all these tools, as seen in Fig. 2, Fig. 3, Fig. 4 and Fig. 5. The file command pretty much gave me what type of the file 1.secret was, which was a PNG file, its dimensions and the bit color profile. The exiftool command gave me the whole metadata of the file. Binwalk, which I noticed being popularly used in online resources and steganography challenges, surprisingly gave me the output of a PNG image and just a copyright string. I even used the additional parameter of -e, which is for extracting any hidden files, but there was nothing interesting in the output. Strings gave out bunch of characters, and there were no interesting/readable characters. Foremost command turned out a bit interesting, since it gave out a directory called "output". This contained an audit.txt and another directory called "png", which had an image named 00000000.png, so I was pursuing this further. I used diff between both files and there was no difference. I even used zsteg for both, but it gave out nothing. Then, I used identify command, but only to know that there was an incorrect sRGB profile, which I later learnt through the web, was a pretty common and negligible issue in PNG files.

But, then, I used the argument -a, in zsteg, which stands for all checks. Then, I got a particular output called "**b1,r,lsb,YX,prime .. text: "gc7W[*C"**". I searched regarding this online and learnt that this indicates there is some bit value manipulations with the red color layer. This can be seen in Fig. 6.

Later, I used a command called stegoveritas, which basically checks the PNG with many tools, bruteforces LSB checks, outputs various transformed images into a directory. As seen in Fig. 7, it gave out some lines like "Found something worth

keeping!", and after the whole command finished running, it gave out a new directory called "results". In results, there were a bunch of various transformed images as seen in Fig. 8. I moved these images from the Docker container to my local VM, as seen in Fig. 9 and Fig. 10.

Out of the images, a few actually showed a flag on top of the image, as seen in Fig. 11.

Later, I also found some online tools, which were basically LSB type steganography decoders, which could decode the flag from a given image file. The results can be seen in Fig. 12 and Fig. 13.

The flag present in 1.secret is **w0ah 5uch 53cr3t m355ag3**. So, I found out the presence of the flag from the 1.secret file, in two ways. One using the stegoveritas command and other by using online web-based steganography tools, both which showed the result i.e. presence of flag, visually in images.

III. 2.SECRET - BONUS

Similar to the 1.secret file, I performed binwalk on this file, and found multiple MySQL ISAM files presence. Putting that aside, I directly used another web-based tool to find if there was any flag hiding in the image, just like the previous task. And luckily, I found a flag, when I used the Inverse(RGB) option in the website.

The flag present in 2.secret is **w3_s3t_up_ambush_0n_l3dg3**. There is a place circled Ledge, on the map. So, I understood the meaning of the flag in this way.

These can be seen in Fig. 14 and Fig. 15.

IV. CONCLUSION

To conclude this lab exercise, I learnt steganography, got familiar with Dominic's stego toolkit, found the hidden flags in the given files successfully.

REFERENCES

- [1] DominicBreuker's Stego Toolkit
- [2] Docker container files mounting
- [3] Copying files from Docker container to local machine
- [4] Online tool 1 - Aperisolve
- [5] Online tool 2 - StegOnline
- [6] LSB Steganography

```
Activities Terminal Mon Dec 4 15:53:09
root@ckimidi-Standard-PC-Q35-ICH9-2009:~# docker run -it -v /home/ckimidi/Downloads:/data e23e447a4
root@ce86b1071af1:/data# ls
1.secret 2.secret
root@ce86b1071af1:/data#
```

Fig. 1. Container started with data mounted

```
root@ce86b1071af1:/data# exiftool 1.secret
File Name                           : 1.secret
File Size                            : 1683 kB
File Modification Date/Time         : 2023:11:28 10:24:28+00:00
File Access Date/Time               : 2023:12:04 09:11:19+00:00
File Last Change Date/Time          : 2023:12:04 09:10:30+00:00
File Permissions                   : rw-r--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                         : Deflated/Inflate
Filter                             : Adaptive
Interlace                           : Noninterlaced
Gamma                             : 2.2
ICC Profile                        : ICC profile
Profile Version                   : 2.1.0
Profile Class                      : Display Device Profile
Color Space Data                  : RGB
Profile Connection Space          : XYZ
Profile Creation Date/Time        : 1998:02:09 06:49:00
Profile File Signature             : acsp
Primary Platform                  : Microsoft Corporation
CMW Flags                          : Not Embedded, Independent
Device Manufacturer                : sRGB
Device Model                       : sRGB
Device Attributes                 : Reflective, Glossy, Positive, Color
Rendering Intent                  : Media-Relative Colorimetric
Connection Space Illuminant       : 0.9642 1 0.82491
Profile Creator                    : HP
```

Fig. 2. File, Exiftool

```
root@ce86b1071af1:/data# binwalk 1.secret
DECIMAL      HEXADECIMAL      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@ce86b1071af1:/data# binwalk 1.secret
DECIMAL      HEXADECIMAL      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@ce86b1071af1:/data# ls
1.secret  2.secret
root@ce86b1071af1:/data#
```

Fig. 3. Binwalk

```
root@ce86b1071af1:/data# strings 1.secret > 1.txt
root@ce86b1071af1:/data# ls
1.secret  1.txt  2.secret
root@ce86b1071af1:/data# head 1.txt
IHDR
g3P
LCPICC profile
Hlno
mntrRGB XYZ
acsppSFT
IEC sRGB
-HR
3desc
root@ce86b1071af1:/data# foremost 1.
1.secret 1.txt
root@ce86b1071af1:/data# foremost 1.secret
Processing: 1.secret
[=]
root@ce86b1071af1:/data# ls
1.secret 1.txt  2.secret  output
root@ce86b1071af1:/data# cd output/
root@ce86b1071af1:/data# output# ls
audit.txt  png
root@ce86b1071af1:/data# output# cd png
root@ce86b1071af1:/data# output# png ls
00000000.png
root@ce86b1071af1:/data# output# png open 00000000.png
```

Fig. 4. Strings, Foremost

```
root@ce86b1071af1:/data# cd output/png/
root@ce86b1071af1:/data# output/png ls
00000000.png
root@ce86b1071af1:/data# output/png zsteg 00000000.png
[=] nothing :
root@ce86b1071af1:/data# output/png cd ../..
root@ce86b1071af1:/data# zsteg 1.secret
[=] nothing :
root@ce86b1071af1:/data# pngcheck 1.secret
OK: 1.secret (855x502, 32-bit RGB+alpha, non-interlaced, -0.4%).
root@ce86b1071af1:/data# identify 1.secret
1.secret PNG 855x502+0+0 DirectClass 8-bit 1.0Mi 0.0000 0m:0.000000
identify identify: iCCP: known incorrect sRGB profile (1.secret).
root@ce86b1071af1:/data#
```

Fig. 5. Zsteg, PNGcheck, Identify

```
root@ce86b1071af1:/data# zsteg -a 1.secret
b1,r,lsb,YX,prime ... text: "gc7W[*('C"
root@ce86b1071af1:/data# cd output/png/
root@ce86b1071af1:/data/output/png zsteg -a 00000000.png
b1,r,lsb,YX,prime ... text: "gc7W[*('C"
root@ce86b1071af1:/data/output/png# cd ../..
root@ce86b1071af1:/data# ls
1.secret 1.txt 2.secret  output
root@ce86b1071af1:/data# cd output/
root@ce86b1071af1:/data/output# ls
audit.txt  png
root@ce86b1071af1:/data/output#
```

Fig. 6. zsteg -a output

```
root@ckimidi-Standard-PC-Q35-ICH9-2009: ~
root@ce86b1071af1:/data/output# clear
root@ce86b1071af1:/data/output# cd ..
root@ce86b1071af1:/data# stegoveritas.py 1.secret
bash: stegoveritas.py: command not found
root@ce86b1071af1:/data# stegoveritas 1.secret
Running Module: SVImage
+-----+
| Image Format | Mode |
+-----+
| Portable network graphics | RGBA |
+-----+
Found something worth keeping!
ISO-8859 text, with very long lines, with no line terminators
Found something worth keeping!
ISO-8859 text, with very long lines, with no line terminators
Found something worth keeping!
ISO-8859 text, with very long lines, with no line terminators
Found something worth keeping!
ISO-8859 text, with very long lines, with no line terminators
Found something worth keeping!
ISO-8859 text, with very long lines, with no line terminators
Running Module: MultiHandler
Exif
====
```

Fig. 7. Stegoveritas output

```
Activities   Terminal
root@ce86b1071af1:/data# cd results/keepers
root@ce86b1071af1:/data# ls
1.secret 1.txt  2.secret  output
root@ce86b1071af1:/data# cd results/
root@ce86b1071af1:/data# ls
1.secret_Alpha_0.png 1.secret_Alpha_5.png 1.secret_Green_6.png 1.secret_Red_7.png
1.secret_Alpha_1.png 1.secret_Alpha_6.png 1.secret_Green_7.png 1.secret_Sharpn.png
1.secret_Alpha_2.png 1.secret_Blue_7.png 1.secret_Max.png 1.secret_Smooth.png
1.secret_Alpha_3.png 1.secret_Edge-enhance.png 1.secret_Median.png 1.secret_alpha_plane.png
1.secret_Alpha_4.png 1.secret_Edge-enhance.More.png 1.secret_Min.png 1.secret_blue_plane.png
1.secret_Alpha_5.png 1.secret_Fin.Edges.png 1.secret_Mode.png 1.secret_green_plane.png
1.secret_Alpha_6.png 1.secret_GaussianBlur.png 1.secret_Red_0.png 1.secret_Inverted.png
1.secret_Alpha_7.png 1.secret_GaussianBlur.More.png 1.secret_Red_1.png 1.secret_red_plane.png
1.secret_Blue_0.png 1.secret_Green_1.png 1.secret_Red_2.png 1.exif
1.secret_Blue_1.png 1.secret_Green_2.png 1.secret_Red_3.png 1.secret_Red_4.png
1.secret_Blue_2.png 1.secret_Green_3.png 1.secret_Red_4.png 1.secret_Red_5.png
1.secret_Blue_3.png 1.secret_Green_4.png 1.secret_Red_5.png 1.secret_Red_6.png
1.secret_Blue_4.png 1.secret_Green_5.png 1.secret_Red_6.png
```

Fig. 8. Results directory

```
root@ckimldi-Standard-PC-Q35-ICH9-2009:~          root@ckimldi-Standard-PC-Q35-ICH9-2009:~ 
Use '-' as the source to read a tar archive from stdin
and extract it to a directory destination in a container.
Use '-' as the destination to stream a tar archive of a
container source to stdout.

Options:
-a, --archive      Archive mode (copy all uid/gid information)
-L, --follow-link  Always follow symbol link in SRC_PATH
root@ckimldi-Standard-PC-Q35-ICH9-2009:~# sudo docker cp -a recursing_difflie:/data/results .
root@ckimldi-Standard-PC-Q35-ICH9-2009:~ ls
results snap
root@ckimldi-Standard-PC-Q35-ICH9-2009:~# cd results/
root@ckimldi-Standard-PC-Q35-ICH9-2009:~/results# ls
1.secret_Alpha_0.png 1.secret_blue_plane.png 1.secret_Min_.png
1.secret_Alpha_1.png 1.secret_Edge-enhance_more.png 1.secret_Med_.png
1.secret_Alpha_2.png 1.secret_alpha_plane.png 1.secret_Red_1_.png
1.secret_Alpha_3.png 1.secret_Find_Edges.png 1.secret_Red_2_.png
1.secret_Alpha_4.png 1.secret_GaussianBlur.png 1.secret_Red_3_.png
1.secret_Alpha_5.png 1.secret_Green_0_.png 1.secret_Red_4_.png
1.secret_Alpha_6.png 1.secret_Green_1_.png 1.secret_Red_5_.png
1.secret_Alpha_7.png 1.secret_Green_2_.png 1.secret_Red_6_.png
1.secret_Alpha_plane.png 1.secret_Green_3_.png 1.secret_Red_7_.png
1.secret_Blue_0.png 1.secret_Green_4_.png 1.secret_red_plane.png
1.secret_Blue_1.png 1.secret_Green_5_.png 1.secret_Sharpener.png
1.secret_Blue_2.png 1.secret_Green_6_.png 1.secret_smooth.png
1.secret_Blue_3.png 1.secret_Green_7_.png
1.secret_Blue_4.png 1.secret_Inverted.png ext
1.secret_Blue_5.png 1.secret_Inverted.png keepers
1.secret_Blue_6.png 1.secret_Max_.png
1.secret_Blue_7.png 1.secret_Median.png
root@ckimldi-Standard-PC-Q35-ICH9-2009:~/results# cd ..
root@ckimldi-Standard-PC-Q35-ICH9-2009:~# mv results /home/ckimldi/Desktop/results
root@ckimldi-Standard-PC-Q35-ICH9-2009:~ ls
snap
root@ckimldi-Standard-PC-Q35-ICH9-2009:~#
```

Fig. 9. Copying images from container to my VM

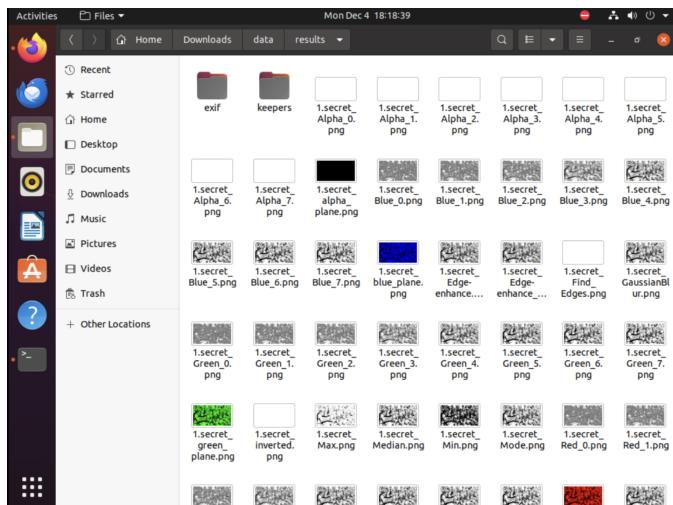


Fig. 10. List of images



Fig. 11. FLAG!!!



Fig. 12. Online tool - AperiSolve

flag(w0ah_Such_b3cr3t_m3bbad3)

Fig. 13. FLAG VISIBLE

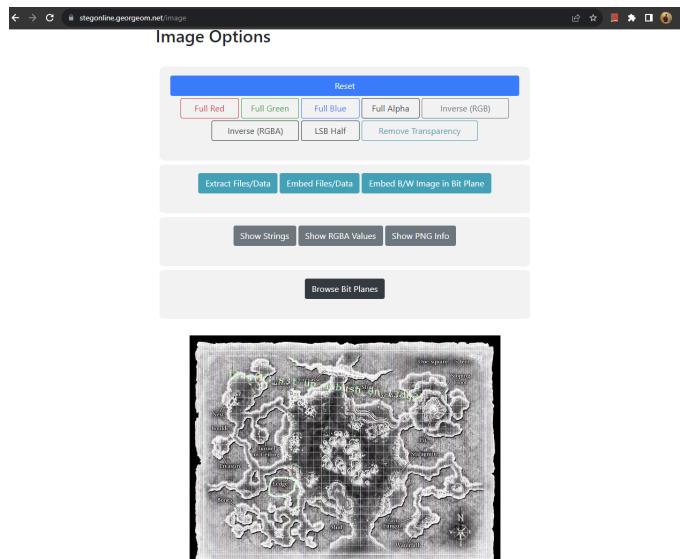


Fig. 14. Online tool - StegOnline by GeorgeOm

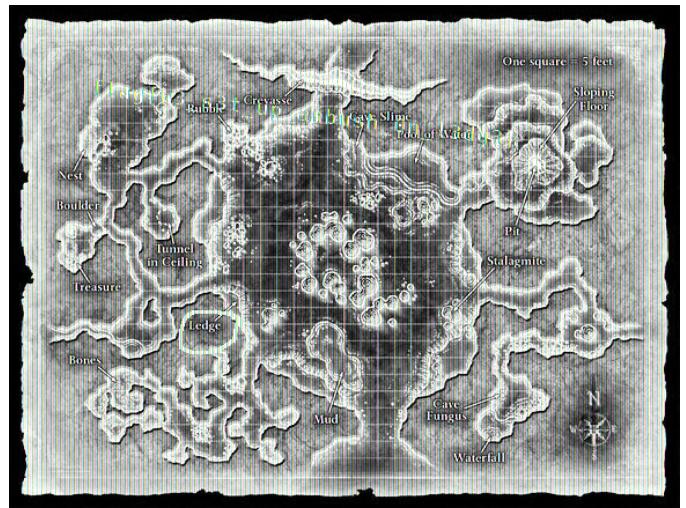


Fig. 15. FLAG VISIBLE