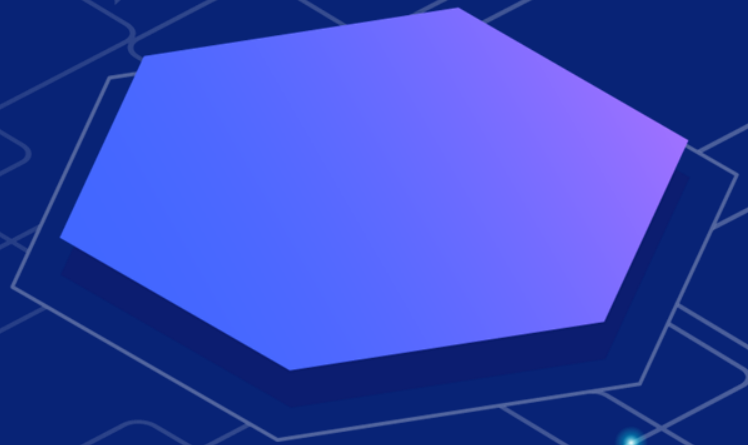
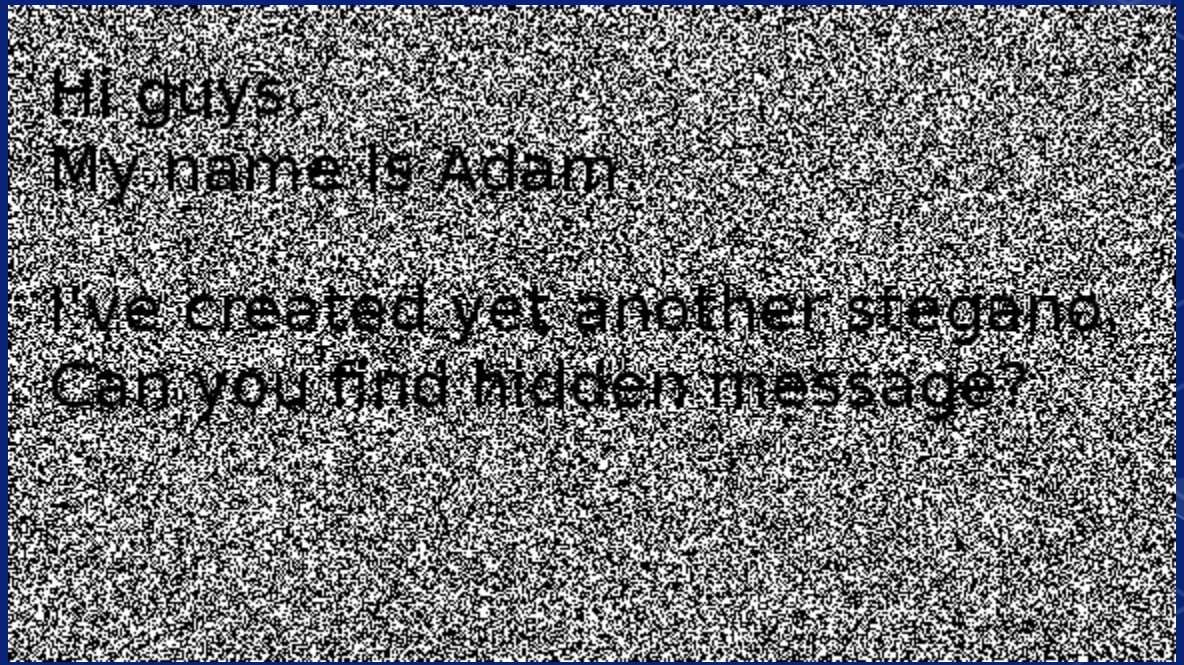
SECCON19 SANDSTORM



> Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video.

Wikipedia

I've received a letter... Uh, Mr. Smith?



Approaching such challenges

- Image metadata
- Text in the raw bytes of the image
- Hidden pixels in a single color plane
- Hidden data in the bits of the pixel
- Embedded file in the image itself
- Something else?

Image metadata

- Basically text that *usually* contains information about the image
- No need to explain how obvious this is

```
$ exiftool sandstorm.png
ExifTool Version Number : 10.80
File Name : sandstorm.png
Directory : .
File Size : 62 kB
File Modification Date/Time : 2019:12:08 19:31:26+01:00
File Access Date/Time : 2019:12:08 19:32:11+01:00
File Inode Change Date/Time : 2019:12:08 19:31:45+01:00
File Permissions : rw-rw-r--
File Type : PNG
File Type Extension : png
MIME Type : image/png
Image Width : 584
Image Height : 328
Bit Depth : 8
Color Type : RGB with Alpha
Compression : Deflate/Inflate
Filter : Adaptive
Interlace : Adam7
Interlace Background Color : 255 255 255
Image Size : 584x328 Megapixels : 0.192
```

Printable text in image bytes

- Slightly more secure than hiding the secret in the image metadata
- Can be found using the `strings` command and a loop

Text after the end of the image

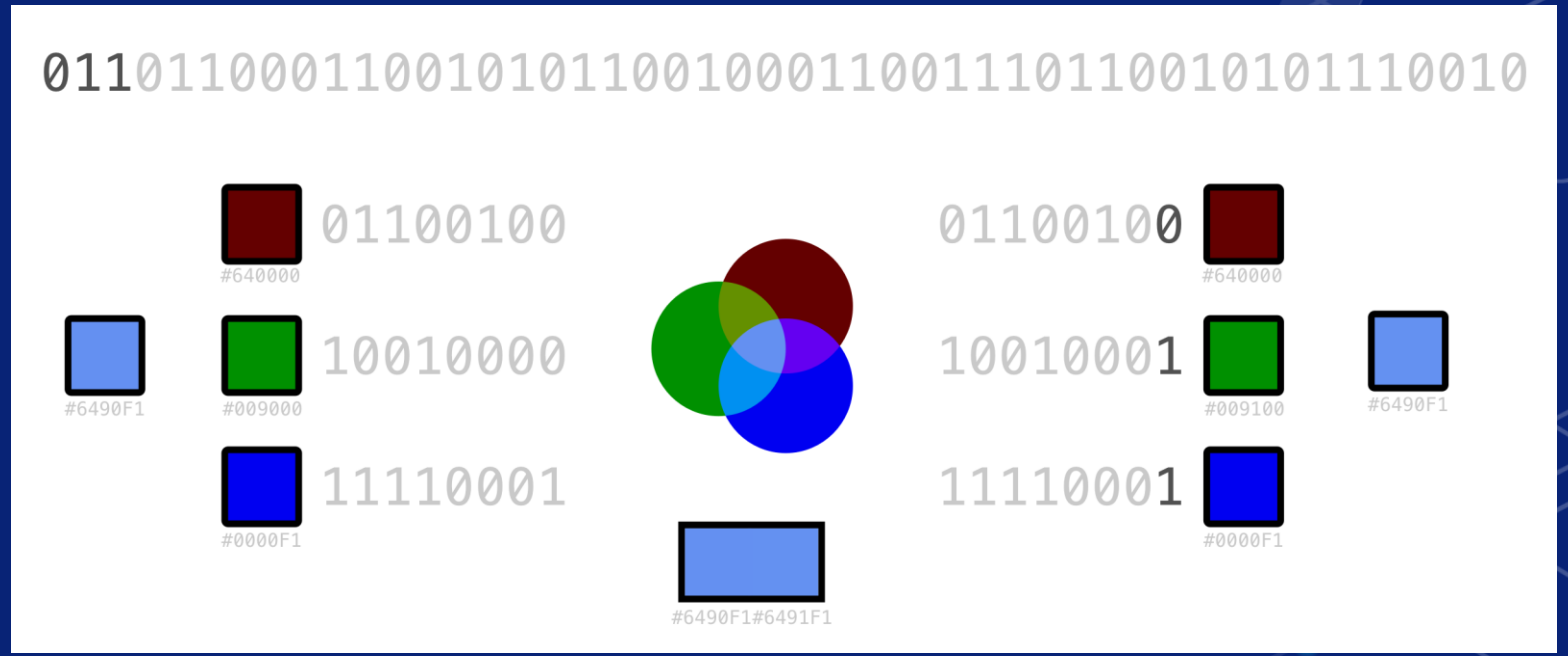
- This may be worth checking if the image is PNG and appears corrupted
- According to the PNG specification, the image file has special “chunks” of 8 bytes at the beginning and the end of the file
- **xxd** is your best friend here

Hidden pixels in a color plane

- A color plane describes how a color is encoded using the primary colors
- Sometimes, isolating different color planes may reveal hidden pixels
- Stegsolve is the perfect tool for the job

Hidden data in the bits of a pixel

- Each pixel needs 8 bits to be represented
- Adding information to the file can be done by changing the LSB of the pixel
- A change of the color of the pixel will occur, but it will not be so easily visible
- Recognizing all changes will reveal the secret
- `zsteg` is the tool for detecting those


01101100110010101100100011001110110010101110010
01100100
#640000
10010000
#009000
11110001
#0000F1
#6490F1
#6490F1#6491F1
01100100
#640000
10010001
#009100
11110001
#0000F1
#6490F1

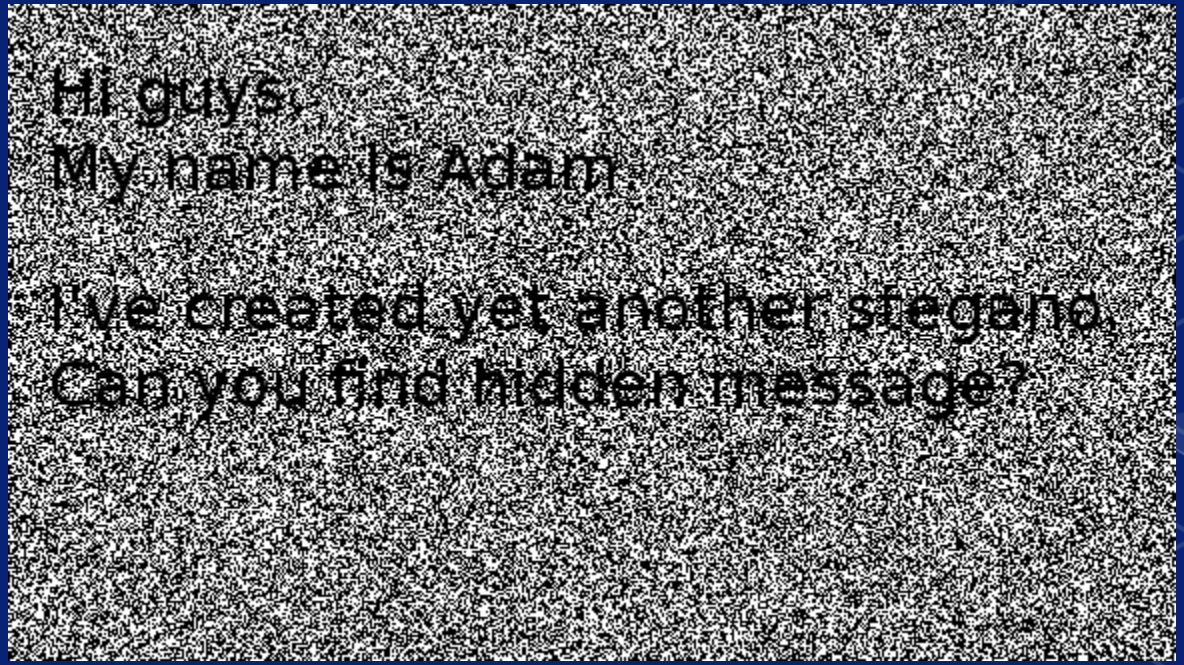
Embedded file in the image itself

- Common practice
- Not directly visible
- The file can be a text file, an archive or even another image
- Inspect the hex of the file and see if you can spot anything suspicious (e.g., PK near the end of the file means zip archive)
- Best friends include: `xxd`, `binwalk`, `dd`, `zsteg`

Solution

- Look more into the picture
- Do a bigger think
- ???
- Profit

Look more into the picture



Who is Adam?

Remember the metadata?

```
$ exiftool sandstorm.png
ExifTool Version Number : 10.80
File Name : sandstorm.png
Directory : .
File Size : 62 kB
File Modification Date/Time : 2019:12:08 19:31:26+01:00
File Access Date/Time : 2019:12:08 19:32:11+01:00
File Inode Change Date/Time : 2019:12:08 19:31:45+01:00
File Permissions : rw-rw-r--
File Type : PNG
File Type Extension : png
MIME Type : image/png
Image Width : 584
Image Height : 328
Bit Depth : 8
Color Type : RGB with Alpha
Compression : Deflate/Inflate
Filter : Adaptive
Interlace : Adam7
Interlace Background Color : 255 255 255
Image Size : 584x328 Megapixels : 0.192
```

Adam7 Algorithm

- Interlacing algorithm used for PNG
- 7 passes needed for full interlacing, each pass creating a subimage
- Each pass replicates certain elements in an 8x8 pattern
- What if we separated the images from the 7 passes and examined them?

Extracting the flag

```python
from PIL import Image

img = Image.open('sandstorm.png')
W, H = img.size

img2 = Image.new('RGBA', (W//8, H//8))

for y in range(0,H//8):
    for x in range(0,W//8):
        img2.putpixel((x,y), img.getpixel((x*8,y*8)))

img2.save('img2.png')
```

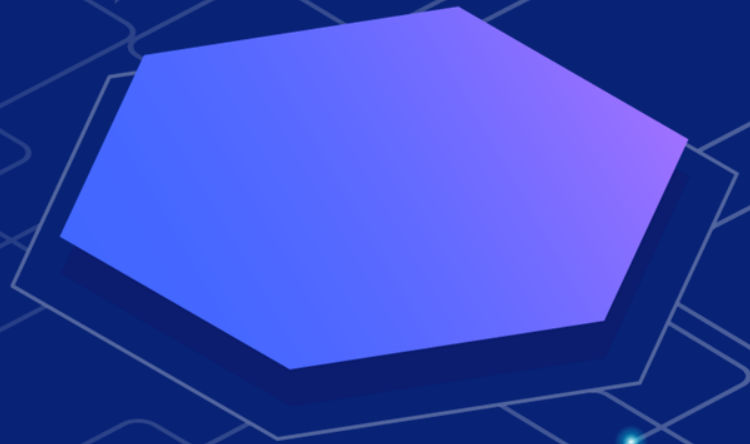
https://minaminao.com/post/2019-10-20-seccon-online-ctf/

Jackpot!

Steganography IRL

- Used for hiding stuff, NOT for encryption -if the medium is suspected, then the confidentiality of the data is compromised
- Often used to spread malware
- Recently used by Russian spies (ofc)

Questions?



Useful links

Steganography toolkit:

https://github.com/DominicBreuker/stego-toolkit

Using Stegsolve in Python (PNG only):

https://agsyndro.me/how-i-reverse-stegsolve-to-automate-it-temp-title/

Sources

https://github.com/10secTW/ctf-writeup/tree/master/2019/SECCON%20CTF%20quals/Sandstorm

https://minaminao.com/post/2019-10-20-seccon-online-ctf/

https://medium.com/@FourOctets/ctf-tidbits-part-1-steganography-ea76cc526b40

https://itnext.io/steganography-101-lsb-introduction-with-python-4c4803e08041

http://www.libpng.org/pub/png/spec/1.2/PNG-Structure.html

https://www.technologyreview.com/s/419833/russian-spies-use-of-steganography-is-just-the-beginning/

https://ctfs.github.io/resources/topics/steganography/file-in-image/README.html

https://en.wikipedia.org/wiki/Interlacing_(bitmaps)

https://en.wikipedia.org/wiki/Adam7_algorithm