Steganography

Steganography is the practice of concealing a file, message, image, or video within another file, message, image, or video. The word steganography comes from New Latin steganographia, which combines the Greek words steganós, meaning "covered or concealed", and -graphia meaning "writing".

For CTF you would get a category of problems to work on stuff like images and blank text files etc.
To solve this type of challenges you need to know about steganography.

Some of the common challenges you would find are: -

I would specially thank john-hammond for the wonderful resource from his git hub.

Steganography

- StegCracker

  Don't ever forget about `steghide`! This tool can use a password list like `rockyou.txt` with steghide. SOME IMAGES CAN HAVE MULTIPLE FILED ENCODED WITH MULTIPLE PASSWORDS.

- Steganography Online

  A tool often used in CTFs for encoding messages into images.

- **`steg_brute.py`**

  This is similar to `stegcracker` above.

- **`openstego`**

  A Java `.JAR` tool, that can extract data from an image. A good tool to use on guessing challenges, when you don't have any other leads. We found this tool after the Misc50 challenge from HackIM 2018

- `Stegsolve.jar`

  A Java `.JAR` tool, that will open an image and let you as the user arrow through different renditions of the image (viewing color channels, inverted colors, and more). The tool is surprisingly useful.

- `steghide`

  A command-line tool typically used alongside a password or key, that could be uncovered some other way when solving a challenge.

- `stepic`

  Python image steganography. Stepic hides arbitrary data inside PIL images. Download it here: http://domnit.org/stepic/doc/

- Digital Invisible Ink Stego Tool

  A Java steganography tool that can hide any sort of file inside a digital image (regarding that the message will fit, and the image is 24 bit colour).

- ImageHide

  For PNG images (or BMP) images, there exists a Windows utility that can hide "ENCRYPTED" text within the LSB. If you also happen to have passwords, you can decrypt this and potentially find a flag. https://www.softpedia.com/get/Security/Encrypting/ImageHide.shtml

- stegoVeritas

  Another steganography tool. A simple command-line tool and super easy to use -- definitely one to at least try.

- Unicode Steganography / Zero-Width Space Characters

  Some text that may be trying to hide something, in a seemingly innocent way, like "Hmm, there may be something hiding here..." may include zero-width characters. This is a utility that might help: https://330k.github.io/misc_tools/unicode_steganography.html ... Other options are just gross find and replace operations in Python IDLE.

- Online LSB Tools

  There are many online LSB tools that work in different ways. If you are given a file that you know is part of a Least Significant Bit challenge, try these tools:

https://manytools.org/hacker-tools/steganography-encode-text-into-image/ Only supports PNG https://stylesuxx.github.io/steganography/

- Other stego tools:

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

- `zsteg`

  Command-line tool for use against Least Significant Bit steganography... unfortunately only works against PNG and BMP images.

- `jsteg`

  Another command-line tool to use against JPEG images. https://github.com/lukechampine/jsteg Handy for Hackerrank Codefest CTF 2018.

- Jstego

  A GUI tool for JPG steganography. https://sourceforge.net/projects/jstego/ It is a Java JAR file similar to stegsolve.jar

- Morse Code

  Always test for this if you are seeing two distinct values... *it may not always be binary!* Online decoders like so: https://morsecode.scphillips.com/translator.html. If you need to be case-sensistive or include a bit more stuff like numbers and punctuation, use this code: https://gist.github.com/JohnHammond/961acabfd85a8715220fa79492b25368

  If you find Morsecode in the "international written form", like "dah-dit-dit-dah" etcetera, you can use this code: https://gist.github.com/JohnHammond/7d3ddb167fa56f139dc4419091237b51 ... which was carved out of this resource: https://morsecode.scphillips.com/morse.html

- Whitespace

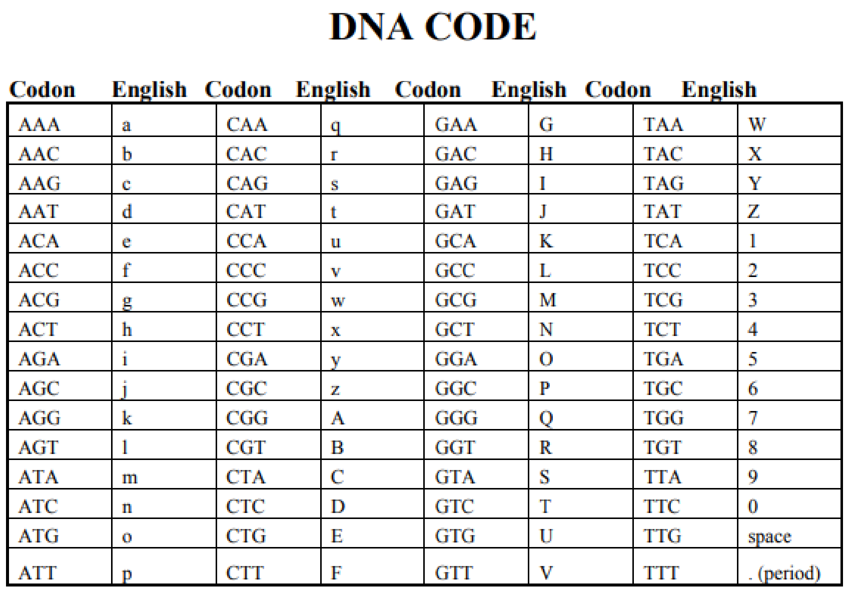
  Tabs and spaces could be representing 1's and 0's and treating them as a binary message... or, they could be whitespace done with `snow` or an esoteric programming language interpreter: https://tio.run/#whitespace

- Audio Speed Change (also change pitch)

```
mplayer -af scaletempo -speed 64 flag.mp3
```

- DNA Codes

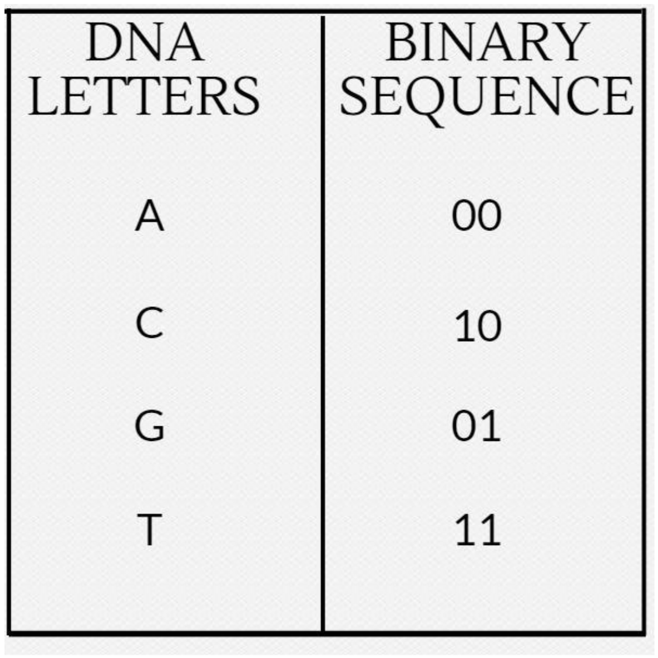
When given a sequence with only A, C, G, T , there is an online mapping for these. Try



DNA CODE

Codon	English	Codon	English	Codon	English	Codon	English
AAA	a	CAA	q	GAA	G	TAA	W
AAC	b	CAC	r	GAC	H	TAC	X
AAG	c	CAG	s	GAG	I	TAG	Y
AAT	d	CAT	t	GAT	J	TAT	Z
ACA	e	CCA	u	GCA	K	TCA	1
ACC	f	CCC	v	GCC	L	TCC	2
ACG	g	CCG	w	GCG	M	TCG	3
ACT	h	CCT	x	GCT	N	TCT	4
AGA	i	CGA	y	GGA	O	TGA	5
AGC	j	CGC	z	GGC	P	TGC	6
AGG	k	CGG	A	GGG	Q	TGG	7
AGT	l	CGT	B	GGT	R	TGT	8
ATA	m	CTA	C	GTA	S	TTA	9
ATC	n	CTC	D	GTC	T	TTC	0
ATG	o	CTG	E	GTG	U	TTG	space
ATT	p	CTT	F	GTT	V	TTT	. (period)

this:



DNA LETTERS	BINARY SEQUENCE
A	00
C	10
G	01
T	11

- Extract Thumbnail (data is covered in original image)

If you have an image where the data you need is covered, try viewing the thumbnail:

```
exiftool -b -ThumbnailImage my_image.jpg > my_thumbnail.jpg
```

- `snow`

  A command-line tool for whitespace steganography (see above).

- SONIC Visualizer (audio spectrum)

  Some classic challenges use an audio file to hide a flag or other sensitive stuff. SONIC visualizer easily shows you spectrogram. **If it sounds like there is random bleeps and bloops in the sound, try this tactic!**

- Detect DTMF Tones

  Audio frequencies common to a phone button, DTMF: https://en.wikipedia.org/wiki/Dual-tone_multi-frequency_signaling.

- Phone-Keypad

  Some messages may be hidden with a string of numbers, but really be encoded with old cell-phone keypads, like text messaging with numbers repeated:

- `hipshot`

  A Python module to compress a video into a single standalone image, simulating a long-exposure photograph. Was used to steal a QR code visible in a video, displayed through "Star Wars" style text motion.

- QR code

  A small square "barcode" image that holds data.

- `zbarimg`

  A command-line tool to quickly scan multiple forms of barcodes, QR codes included. Installed like so on a typical Ubuntu image:

```
sudo apt install zbar-tools
```

- Punctuation marks `!`, `.` and `?`

  I have seen some challenges use just the end of `.` or `?` or `!` to represent the Ook esoteric programming language. Don't forget that is a thing!

These are the some of the tools and in my next repo called tools I am going to add many tools for solving these challenges.