Competence Centers for
Excellent Technologies

What Time Is It?

Steganography in File System Metadata

Sebastian Neuner, SBA Research

Bundesministerium für Verkehr, Innovation und Technologie

Bundesministerium für Wissenschaft, Forschung und Wirtschaft

FFG

wirtschafts agentur wien
Ein Fonds der Stadt Wien

Christian Doppler Forschungsgesellschaft

FWF Der Wissenschaftsfonds.

whoami

- Security Researcher at SBA–Research
- Bug Hunter / Pentester
- CTFs!!11elf

What to Expect Today

- What is steganography
- Examples
- File system metadata steganography
- Special case: Timestamps
- Demo

What Is Steganography?

- Conceal data in data
- Steganos στεγανός and graphein γράφειν
  → Air-tight writing (well...almostˆˆ)

The important thing: Hide data in data, so no-one knows that it is hidden

Stego Examples

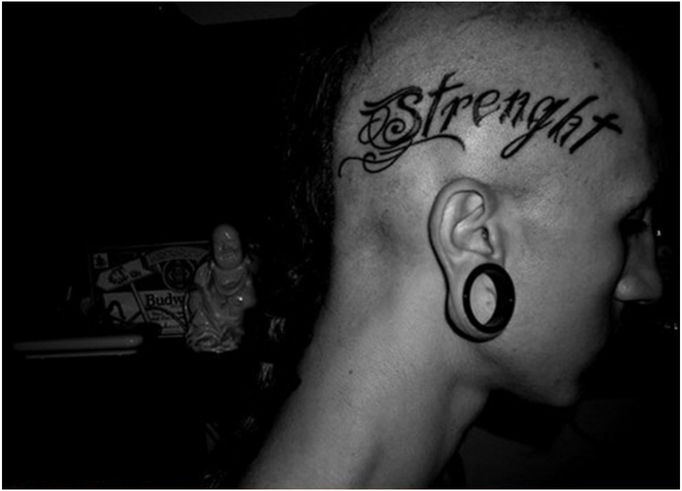
Historical Stego

- Transfer hidden messages to your allies through the enemy territory
- Ancient Greece: Tattoo the shaved head of a slave[1]

  → Hair needs to regrow (takes time)
- Having slaves with ”encoded” heads for a lot of possible use-cases???

[1] Slave of Histiaeus

Historical Stego



And take care of spelling errors :D

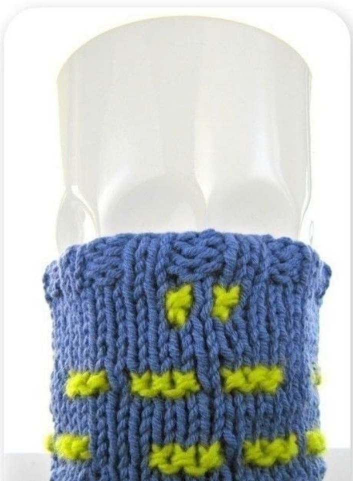
Historical Stego

- French Resistance sent couriers with invisible ink on back
- When: World War II

(Semi-) Historical Stego

One more example...

- Knitted Morse Code
- In carpets and tapestries



Modern Stego

A lot of stuff based on historical Stego...

- Morse Code while blinking eyes (American POW 1966)
- Historical tattoos → modern UV–pens

  (Would also work on skin...)

Digital Stego

Digital Stego

ISIS / Al-Qaeda use steganography over various channels...[2]

- Discovered by Mossad
- Messages encoded into ebay offers, Reddit messages and ”X-rated-pics”

  (Hard work, guys :D)

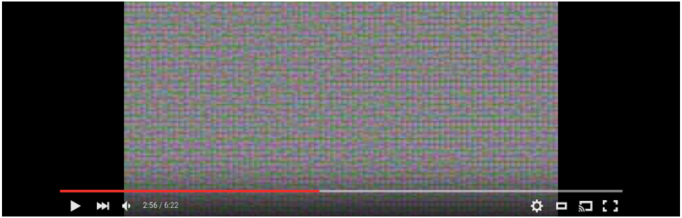
[2] http://nypost.com/2015/03/01/terrorists-using-ebay-and-reddit-to-send-coded-messages-mossad/, http://www.independent.co.uk/news/world/middle-east/isis-and-al-qaeda-sending-coded-messages-through-ebay-pornography-and-reddit-10081123.html

Digital Stego

Hide data in YouTube videos[3]

- Not really Stego
- ”For backup reasons”
- Discrete Cosine Transform
- Parameters for encoding have to be known

  (And maybe it’s encrypted?)

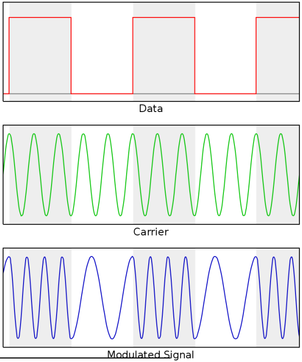


[3] https://hackaday.com/2015/08/23/transfer-data-via-youtube/

Digital Stego

Transmit information in the trilling of a referees whistle[4]

- I will stop after this example → I am going too far now :D
- Frequency shift key modulation (FSK)
- Perl script for encoding: 100 baud FSK

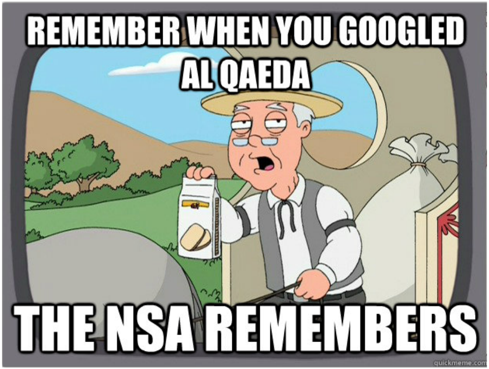


[4] http://www.windytan.com/2015/10/pea-whistle-steganography.html

Steganography in File System Metadata

Why Stego?

- As you have seen: Stego is almost everywhere
  (can be applied / injected almost everywhere)
- Advantage for the good guys (Snowden?)
- Another layer of abstraction to the bad guys (Agencies?)



Why FS Metadata Stego?

Because file systems are everywhere. And every filesystem needs metadata (in some form)

FS Metadata Stego

Requirements:

- Do not corrupt FS on modification
- Do not make files unreadable
- Be stealth
- Be robust
- Rely on Kerkhoffs Law

FS Metadata Stego

Feature	Resolution	suitable
File name	free text	∼
File created	1s-1ns	✓
File modified	1s-1ns	∼
File access	1s-1ns	∼/✓
File metadata modified	1s-1ns	∼/✓
File size	any size	∼
Fragmentation	arbitrary	∼
Permissions	r/w/x	✗
Owner, Group	user/group ID	✗
File type	soft-/hard link	✗
Data location	best fit	∼

Table: Suitability of file system metadata

FS Metadata Stego

- Permission, type and ownership modification would very likely make the file unreadable
- Data fragmentation, location of the file and file name are detectable

  → In case of fragmentation: statistical outlier detection of file fragmentation
- Creation and access timestamps are suitable

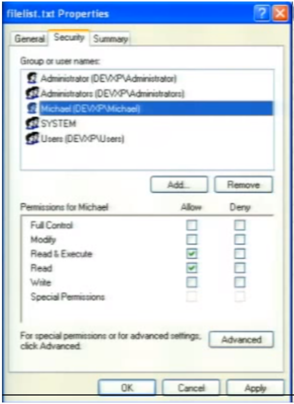
  → More later...

Examples

ACL Stego

Presented at BlackHat 2013 by Michael Perklin[5]

- Cool idea including a PoC
- Shown on Windows FSs
- Not totally stealth...

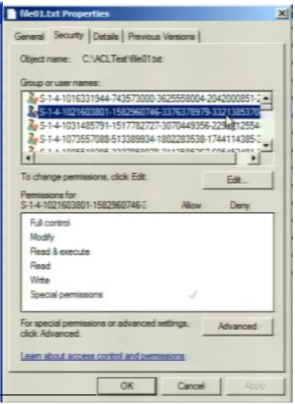


Group or user names:

- Administrator (DEVXP\Administrator)
- Administrators (DEVXP\Administrators)
- Michael (DEVXP\Michael)
- SYSTEM
- Users (DEVXP\Users)

Permissions for Michael	Allow	Deny
Full Control	□	□
Modify	□	□
Read & Execute	☑	□
Read	☑	□
Write	□	□
Special Permissions	□	□

For special permissions or for advanced settings, click Advanced.



Object name: C:\ACLTest\file01.txt

Group or user names:

- S-1-4-1016331944-743573000-3625558004-2042000851-2
- S-1-4-1021603801-1582960746-3376378979-332138537(
- S-1-4-1031485791-1517782727-3070449356-2295212554
- S-1-4-1073557088-513389934-1802283538-1744114385-3

To change permissions, click Edit.

Permissions for S-1-4-1021603801-1582960746-3	Allow	Deny
Full control		
Modify		
Read & execute		
Read		
Write		
Special permissions	✓	

For special permissions or advanced settings, click Advanced.

[5] https://www.youtube.com/watch?v=J4x8Hz6_hq0

Fragmentation Steganography

Fragmentation patterns in the cluster distribution of an existing file[6]

- Up to 24bits per cluster (2KB cluster size) on a half empty disk
- Encrypted data embedding
- Stated as ”statistically undetectable”
- Shown on Windows’ FAT FS
- Defragmentation will (most likely) kill all the information

[6] http://www.sciencedirect.com/science/article/pii/S016740481000088X

Permutation Steganography

Permutation of file ordering in FAT[7]

- Based on: Files are differently ordered by FAT and displayed by a GUI
- 15bytes to embed require 33 files
- On file deletion, the embedded data is killed (or relying on FATs undeletion)
- On file insertion, the order could be disrupted

[7] http://link.springer.com/chapter/10.1007/978-3-662-46739-8_6

Timestamp Steganography

Timestamp–Basics NTFS

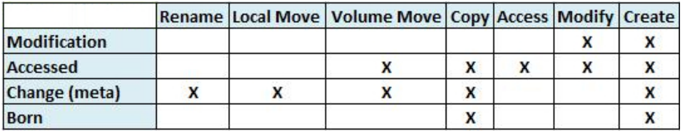
(Our PoCs target NTFS from Win Vista on → later...)

- MACE (Modified, Access, Creation, Modified MFT entry)
- Each 64bits

  → 24bits of that describe the nano seconds
- Number of 100 nano seconds since 1.1.1601

Timestamp–Basics NTFS

Before Vista (XP...):



	Rename	Local Move	Volume Move	Copy	Access	Modify	Create
Modification						X	X
Accessed			X	X	X	X	X
Change (meta)	X	X	X	X			X
Born				X			X

Timestamp–Basics NTFS

Vista++

- By default: NtfsDisableLastAccessUpdate set to 1
  → Immutable access time
- (ext4 mount option ”noatime”)

Timestamp Stego–Idea

Take the nano-second-part of timestamps

- Normally not presented to the user
- Suitable FSs: NTFS, ext4, btrfs, ZFS, XFS, and JFS
- Non-suitable FSs: FAT32, HFS+, ext3, ext2 and ReiserFS

Timestamp Stego–PoC *

Embed information in the creation (C) and access (A) nano-timestamp-parts of files’ metadata

- Python
- NTFS
- Error correction and encryption
- Kerkhoffs Principle!

Timestamp Stego–PoC 1

Save a metadata file

- Produce a metadata file, containing the location of all modified files
- Error corrected payload is encrypted
- Metadata file is encrypted also (different algorithm)
- Drawback: Obviously a file with random data is lying around

Timestamp Stego–PoC 2

Oblivious Replacement

- Take the data
- Produce error correcting codes
- Hide a canary byte in the creation timestamp
- Hide the length indicators
- Encrypt the stuff
- Embed it

Timestamp Stego–Thoughts

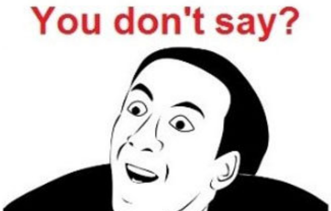
- The canary is needed to recover the correct order of the files
- The amount of error correction is variable but influences the possible capacity
- Speaking of capacity:

  → PoC 1 is able to use 48bits payload, where PoC 2 just 40 bits (canary byte)

  → The more error-correction, the more capacity is needed (the more errors are recoverable)

Timestamp Stego–Thoughts

- The canary is needed to recover the correct order of the files
- The amount of error correction is variable but influences the capacity
- Speaking of capacity:
  → PoC 1 is able to use 48bits payload, where PoC 2 just 40 bits (canary byte)
  → The more error-correction, the more capacity is needed (the more errors are recoverable)



Timestamp Stego–Capacity

Example for PoC2 (oblivious replacement)

- Creation: 3bytes / Access: 3bytes
    - Minus: 1byte per file (canary)
    - Minus: Every 255th file contains the length of the whole data
    - Minus: Error correction

Timestamp Stego–Capacity Win8

Freshly installed Win8 → roughly 160k files

- Theoretical payload: 48bits * 160k: 960KB
- Real payload: (40bits * 160k) - (160k / 255 * 5) - (15% error correction)

  → ∼ 680kb hard payload

Impressive?

Impressive?

BUT...

...we have encryption

...we have error correction

...we can recover order

...we are stealth

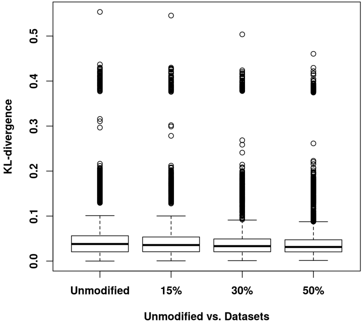
Stealth?

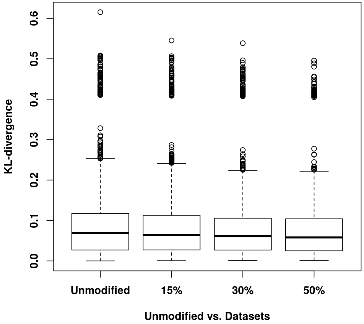
By relying on the requirement of encryption to look like random data, our embedded data looks like random data.

Stealth → statistically undetectable

Undetectable?

Measured with Kullback–Leibler divergence (”measure of the difference between two probability distributions”[8])





[8] https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence

DEMO

DEMO

Concluding

→ Publish paper in 2016

→ On date of publication: Source code on github (Twitter)

Thank you for your attention...

Sebastian Neuner

sebastian.neuner@gmail.com
PGP: 0x7864146D

sneuner@sba-research.org
PGP: 0x5E82F701

🐦 @sebastian9er

I has a question...

Image References

https://ctf.isis.poly.edu/static/archives/2013/about/ctf.jpg
http://tpj.videonativesltd.netdna-cdn.com/wp-content/uploads/2014/11/
strentgh-head-tattoo-fail.jpg
http://images.coplusk.net/project_images/116623/image/full_tumbler_cozy_full.jpg
http://www.the-scientist.com/wordpress/wp-content/uploads/2011/09/secret-cropped.jpg
https://hackadaycom.files.wordpress.com/2015/08/stegmain.png?w=800
https://upload.wikimedia.org/wikipedia/commons/thumb/3/39/Fsk.svg/800px-Fsk.svg.png
http://i2.kym-cdn.com/photos/images/original/000/558/887/01d.png
https://blogs.sans.org/computer-forensics/files/2010/10/ts_change_rules_gui1.jpg
https://i.imgur.com/L9cPO.png http://cdn.meme.am/instances/32090244.jpg
http://www.quickmeme.com/img/a6/
a6984aabbb5d3a2249abac266b44bd266214648332f0aeb5bdd8b4fdd9d00331.jpg
http://philbaumann.com/wp-content/uploads/2009/01/Twitter_bird_logo_2012.png
http://img4.wikia.nocookie.net/__cb20121008041422/thehungergames/images/b/bd/I_has_a_
question.jpg