

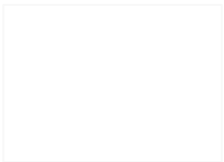
In-depth Anti-Forensics Challenges of Steganography & Discovering Hidden Data

EkoParty 2008

Domingo Montanaro

<conferences@montanaro.org>

Buenos Aires, October 2nd, 2008



Who am I ?

2007-2008

- **Manager of Research & Development for the Information Security and Computer Forensics Labs @ ScanIT – Oger Systems – Dubai - UAE**

2003-2007:

- **Security Coordinator at Unibanco S/A – 3rd Brazilian largest bank – Incident Response, Computer Forensics, Internal Investigations, Data Recovery, coordination of Pen-Test team**
- **Speaker at Security Conferences (Brazil, Argentina, USA, Malaysia, Vietnam, China, UAE, Saudi Arabia)**
- **Instructor in Information Security post-graduation courses and for special trainings of government agencies in Brazil**

2000-2008

- **Computer Forensics Connoisseur, working with Law Enforcement Agencies in Brazil**

**My Areas of Research: Computer Forensics & Anti-Forensics –
New techniques to improve and subvert analyses**

Agenda

- - **Steganography – wtf? (Where, To, From)**
- - **General Concepts & History (Application)**
- - **Steganography & Anti-Forensics**
- • **Differences between current approaches (more covered and less covered)**
 - o **“Front” Approach**
 - o **“Back” Approach and improvements**
- - **Detection**

What is Steganography?

According to Wikipedia:

Steganography is the art and science of writing hidden messages in such a way that no one apart from the sender and intended recipient even realizes there *is* a hidden message

Word: Comes from Greek and means “covered, or hidden writing”.

Two examples of how it appeared:

Demaratus sent a warning about a forthcoming attack to Greece by writing it on a wooden panel and covering it in wax

Another ancient example is that of Histiaeus, who shaved the head of his most trusted slave and tattooed a message on it. After his hair had grown the message was hidden

Real World Cases

Steganography is being used as a very powerful technique, even to Criminal Organizations. Some examples:

From: <http://www.wired.com/politics/law/news/2001/02/41658>

Bin Laden: Steganography Master?

Declan McCullagh 02.07.01 | 2:00 AM

USA Today reported on Tuesday that bin Laden and others "are hiding maps and photographs of terrorist targets and posting instructions for terrorist activities on sports chat rooms, pornographic bulletin boards and other websites, U.S. and foreign officials say."


The technique, known as steganography, is the practice of embedding secret messages in other messages -- in a way that prevents an observer from learning that anything unusual is taking place. Encryption, by contrast, relies on ciphers or codes to scramble a message.

Real World Cases (2)

Hello Kitty images used by Drug Lord as steganography vector

Hello Kitty was used by a notorious Colombian drug lord Juan Carlos Ramirez Abadia to hide messages to his minions. Badia apparently picked Hello Kitty as his courier because his wife was a big fan of the Japanese icon -- she had even decorated one of her rooms in a Brazilian house with Hello Kitty-themed chairs, watches and wallpaper

It's getting more light



HOME • CVPR • SUBMISSION • CONTACT

WVU'08, Anchorage - Alaska, US
June 23rd, 2008

First IEEE International
Workitorial on Vision of the Unseen

From the website:

...This unique event will engage the Vision Community in this challenging area. WVU'08 is a combined tutorial and a workshop exploring the many facets of vision and pattern recognition to 'see' what humans cannot. It will be held in conjunction with CVPR in Anchorage, Alaska on June 23rd, 2008.

Why now?

From:

http://www.darkreading.com/document.asp?doc_id=136702&WT.svl=news1_1

Research Shows Image-Based Threat on the Rise

New Purdue University research shows steganography, long considered a minor threat, may be on the rise

OCTOBER 18, 2007 | 6:00 PM

By Kelly Jackson Higgins

There are an **estimated 800 or so steganography tools available online**, many of them free and with user-friendly graphical user interfaces and point-and-click features. This broad availability making steganography more accessible and easier to use for hiding and moving stolen or illicit payloads, experts say.

Digital Steganography 101

Commonly known by the possibility of hiding “messages” inside graphics files, such as jpeg, gif, bmp and alike.

Techniques:

Least Significant Bit (LSB) insertion, masking and filtering, algorithms and transformations, noise and color reduction, etc.

How about other methods

There are other methods to hide data from “Investigators” (Not deeply covered in this presentation), such as:

- [ADS](#)
- Slack Space
- StegFS - www.mcdonald.org.uk/StegFS/
- Intrinsic file system tricks using “unused” space – Unix & Windows – grugq’s cool stuff, WaffensFS- Hide data in spoofed journals, KYfs- hide in Null directory entries, DataMuleFs – Hide in reserved space
- Plausible Deniability – Super 31337 TrueCrypt! :D

How about other methods

There are other methods to hide data from “Investigators”, such as:

- ADS
- [Slack Space](#)
- StegFS - www.mcdonald.org.uk/StegFS/
- Intrinsic file system tricks using “unused” space – Unix & Windows – grugq’s cool stuff, WaffenFS- Hide data in spoofed journals, KYfs- hide in Null directory entries, DataMuleFs – Hide in reserved space
- Plausible Deniability – Super 31337 TrueCrypt! :D

How about other methods

There are other methods to hide data from “Investigators”, such as:

- ADS
- Slack Space
- StegFS - www.mcdonald.org.uk/StegFS/
- Intrinsic file system tricks using “unused” space – Unix & Windows – grugq’s cool stuff, WaffFS- Hide data in spoofed journals, KYfs- hide in Null directory entries, DataMuleFs – Hide in reserved space
- Plausible Deniability – Super 31337 TrueCrypt! :D

Well known Method

Original Image + “Message” = Image (Cover) with hidden text
Optional Encryption

Softwares that currently perform this task: From <http://www.jjtc.com/Steganography/toolmatrix.htm>

*Blindside *BMP Secrets *BMPEmbed
*Contraband Hell Edition *Hermetic Stego *Data Stash *DCT-Steg
*EzStego *F5 *Giovanni *Hide and Seek *Outguess *S-Tools

Well known Method - Example

Image

CutePlayerFromBestTeam.jpg

Size: 31.020 Bytes



Well known Method (2)

Run of OutGuess tool, by Niels Provos

```
vmdebian:/mnt/srv# outguess -k 123teste -d message.txt CutePlayerFromBestTeam.jpg
CutePlayerFromBestTeam2.jpg
Reading CutePlayerFromBestTeam.jpg....
JPEG compression quality set to 75
Extracting usable bits: 27653 bits
Correctable message size: 6491 bits, 23.47%
Encoded 'message.txt': 2032 bits, 254 bytes
Finding best embedding...
  0: 1042(50.5%)[51.3%], bias 1270(1.22), saved: -3, total: 3.77%
  1: 1037(50.2%)[51.0%], bias 1237(1.19), saved: -2, total: 3.75%
  2: 1048(50.8%)[51.6%], bias 1130(1.08), saved: -4, total: 3.79%
 22: 1009(48.9%)[49.7%], bias 1132(1.12), saved: 0, total: 3.65%
 154: 998(48.4%)[49.1%], bias 1138(1.14), saved: 2, total: 3.61%
 187: 985(47.7%)[48.5%], bias 1140(1.16), saved: 3, total: 3.56%
187, 2125: Embedding data: 2032 in 27653
Bits embedded: 2064, changed: 985(47.7%)[48.5%], bias: 1140, tot: 27588, skip: 25524
Foiling statistics: corrections: 581, failed: 0, offset: 61.261317 +- 124.089806
Total bits changed: 2125 (change 985 + bias 1140)
Storing bitmap into data...
Writing CutePlayerFromBestTeam2.jpg....
```

Well known Method (3)

Image

CutePlayerFromBestTeamWTF?

Size: 2008BeijingChampion



Ouch ... Magica? :P

Well known Method (4)

Image

NewPlayer.jpg

Size: 69 Tons of times



Oops!

Well known Method (5) - Example

Image

CutePlayerFromBestTeam2.jpg

Size: 27.828 Bytes

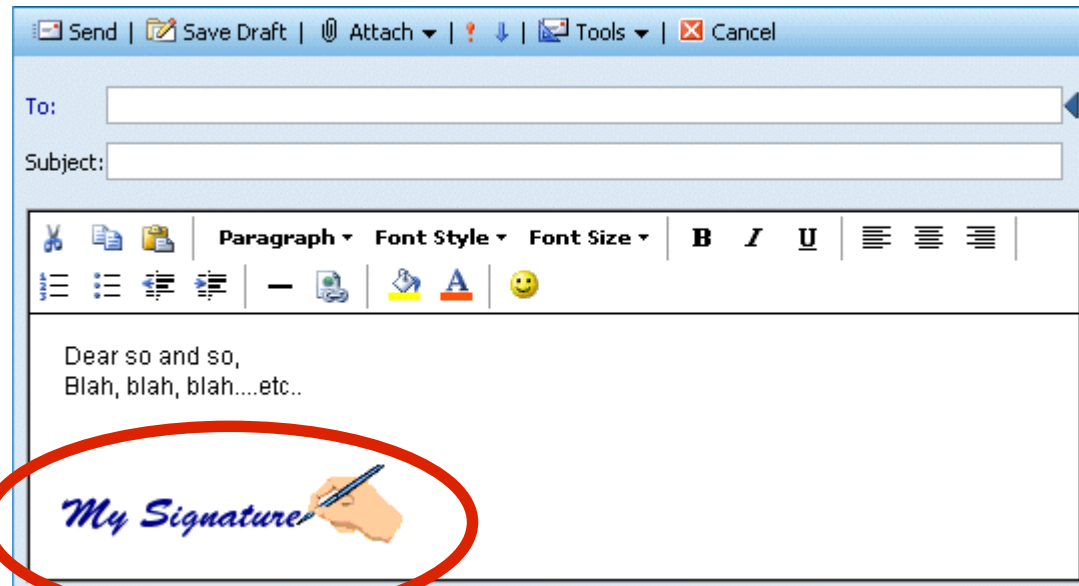


Possible Scenarios for Old Technique

Previous documentation suggests “messages” + Images to implement Digital Steganography

This method is known since 90s and still very efficient.

Single Harmful scenario:



Signature + Embedded Evil Data ←

Simplistic Image Trickery

- Image files follow their layout standards, as of any other kind of file
- Each standard has it's own data hiding capabilities (GIF, BMP, TIFF, etc) – of course, not the original purpose

Simplistic Image Trickery (2)

Nome	Tamanho	Tipo	Data de modificação
logo_h2hc	8 KB	Imagem no formato...	15/2/2006 18:44
trecho	585 KB	Winamp media file	15/2/2006 18:54

Two simple files

```
F:\Estudos\StegTest>copy logo_h2hc.gif /b + trecho.mp3
logo_h2hc.gif
trecho.mp3
1 arquivo(s) copiado(s).
F:\Estudos\StegTest>
```

Simply copy command

Nome	Tamanho	Tipo	Data de modificação
logo_h2hc	592 KB	Imagem no formato...	16/2/2007 15:05
trecho	585 KB	Winamp media file	15/2/2006 18:54

The 2 files continue, but notice the size of "logo_h2hc.gif"



Opening the file on the standard Image Visualization app, it comes up what was expected



Dragging and dropping the same GIF file on a winamp's window, we have 37 seconds of sound.

Different Approaches between previous 2 methods

“Front” Approach which messes with the “Rendering” Data

- **Transform Domain**
- **Spread Spectrum**
- **Statistical method**
- **Distortion, Noise and color reduction**
- **LSB**

“Back” Approach (BIA), which doesn’t touch the “Rendering” or “Interpreting” Data

- **Appending (last example)**
- **Comments Injection**
- **Other file formats characteristics (a.k.a. Insertion)**
- **Use of “Reserved” fields -> For Hacker use?**

LSB – Least Significant Bit

little endian study; 24 bit image

Example:

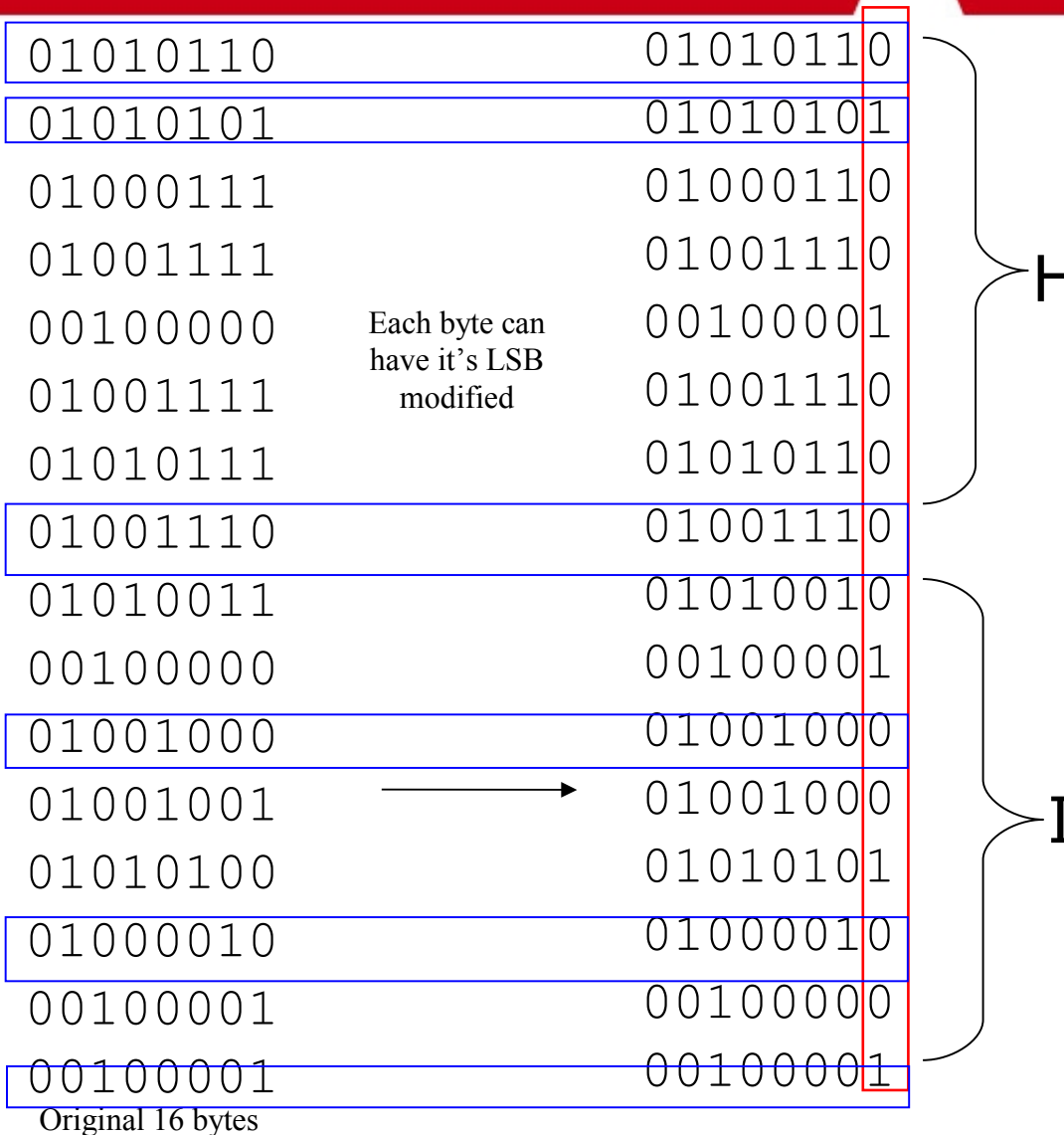
24-Bit Image

Each 8-bit
represents one
"channel"

R (8bit)

G (8bit)

B (8bit)



Going deeper

Every filetype has it's possibilities of storing “evil” data (not addressing compression formats here).

Harmful to think on all this knowledge about hiding information (stego) in files to come in a toolkit.

Scenario:

LibStego (ongoing project) – Supports data hiding on several file formats, parsing tons of file formats from wotsit.org

Supports 3 modes of operation

- 1) Modifying legitimate files – Ex: comments on graphic files**
- 2) Use redundant space on Multimedia formats (GIF, JPEG, AVI, MOV, etc), OLE formats (doc, xls, ppt, etc – not talking about compression here too) and others (DWG, CDR, etc)**
- 3) Use of alignment space on executable files (PE, ELF, etc)**

Use of redundant/Zero/Align spaces

Executables (ELF, Win32PE, etc) when compiled, depending on the compiler, most of the times need to have some space for alignment between subroutines.

Not a new idea in the IT field, since it's used by virii coders (injecting malware instructions into space used for alignment)

```
4AD051A5: C3 RETN ; end of subroutine
4AD051A6: 90 NOP ;
4AD051A7: 90 NOP ;
4AD051A8: 90 NOP ;
4AD051A9: 90 NOP ;
4AD051AA: 55 PUSH EBP ; begin of next subroutine
```

Alignment that can be used to store data
Can be 0x90, 0xCC or signature-based like GCC

On a 2GB “system” filesystem, it's possible to store nearly 1 MB on a “Second Filesystem” inside the “system” filesystem, only using alignment spaces (including DLLs) – Need to remember that relative (short) JMPs are needed to return in the program normal flow.

The libStego Project - Examples

Field “Comment Extension” in GIF89a from CompuServe Graphics Interchange Format

24. Comment Extension.

a. **Description.** The Comment Extension contains textual information which is not part of the actual graphics in the GIF Data Stream. It is suitable for including comments about the graphics, credits, descriptions or any other type of non-control and non-graphic data. The Comment Extension may be ignored by the decoder, or it may be saved for later processing; under no circumstances should a Comment Extension disrupt or interfere with the processing of the Data Stream. This block is **OPTIONAL**; any number of them may appear in the Data Stream.

b. Required Version. 89a.

c. Syntax.

	7 6 5 4 3 2 1 0	Field Name	Type
0	+-----+ +-----+	Extension Introducer	Byte
1	+-----+ +-----+	Comment Label	Byte
N	+=====+ +=====+	Comment Data	Data Sub-blocks
0	+-----+ +-----+	Block Terminator	Byte

The libStego Project - Examples

Field `.comment` in ELF file format

Figure 1-14: Special Sections

Name	Type	Attributes
<code>.bss</code>	SHT_NOBITS	SHF_ALLOC + SHF_WRITE
<code>.comment</code>	SHT_PROGBITS	none
<code>.data</code>	SHT_PROGBITS	SHF_ALLOC + SHF_WRITE
<code>.data1</code>	SHT_PROGBITS	SHF_ALLOC + SHF_WRITE
<code>.debug</code>	SHT_PROGBITS	none
<code>.dynamic</code>	SHT_DYNAMIC	see below
<code>.dynstr</code>	SHT_STRTAB	SHF_ALLOC
<code>.dynsym</code>	SHT_DYNSYM	SHF_ALLOC
<code>.fini</code>	SHT_PROGBITS	SHF_ALLOC + SHF_EXECINSTR
<code>.got</code>	SHT_PROGBITS	see below
<code>.hash</code>	SHT_HASH	SHF_ALLOC
<code>.init</code>	SHT_PROGBITS	SHF_ALLOC + SHF_EXECINSTR
<code>.interp</code>	SHT_PROGBITS	see below
<code>.line</code>	SHT_PROGBITS	none
<code>.note</code>	SHT_NOTE	none
<code>.plt</code>	SHT_PROGBITS	see below
<code>.relname</code>	SHT_REL	see below

The libStego Project - Examples

Comments Chunk in Wave File Format

Comments Chunk Format

```
#define CommentID 'COMT' /* chunkID for Comments Chunk */

typedef struct {
    ID            chunkID;
    long          chunkSize;

    unsigned short numComments;
    char          comments[];
}CommentsChunk;
```

The ID is always COMT. chunkSize is the number of bytes in the chunk, not counting the 8 bytes used by ID and Size fields.

The numComments field contains the number of Comment structures in the chunk. This is followed by the Comment structures, one after the other. Comment structures are always even numbers of bytes in length, so there is no padding needed between structures.

The Comments Chunk is optional. No more than 1 Comments Chunk may appear in one FORM AIFF.

The libStego Project - Examples

Comments on PDF files

From the “Portable Document Format Reference Manual” Version 1.3:

5.14 Body

The body of a PDF file consists of a sequence of indirect objects representing a document. The objects, which are of the basic types described in Chapter 4, represent components of the document such as fonts, pages, and sampled images.

Comments can appear anywhere in the body section of a PDF file. Comments have the same syntax as those in the PostScript language; they begin with a % character and may start at any point on a line. All text between the % character and the end of the line is treated as a comment. Occurrences of the % character within strings or streams are not treated as comments.

& Anti-Forensics ?

Computer Forensics Investigations pretty much follow the same methodology

- ❖ Preparation
- ❖ Acquisition
- ❖ Preservation
- ❖ Examination and Analysis
- ❖ Reporting

& Anti-Forensics(2) ?

Common Anti-Forensics attacks target these steps:

- ❖ Preparation – Logical & Physical Bombs
- ❖ Acquisition – Subversion by rootkits
- ❖ Preservation
- ❖ Examination and Analysis – Data hiding (steganography, cryptography, etc)
- ❖ Reporting

& Anti-Forensics ?

How to deal with Steganography in common procedural forensics investigations?

1) File system analysis

	FILE ANALYSIS	KEYWORD SEARCH	FILE TYPE	IMAGE DETAILS	META DATA	DATA UNIT	HELP	CLOSE	
Directory Seek	r / r	\$Boot	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	8192	48	0	7-128-1
	d / d	\$Extend/	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	448	0	0	11-144-4
Enter the name of a directory that you want to view. C:/	r / r	\$LogFile	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	67108864	0	0	2-128-1
	r / r	\$MFT	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	127991808	0	0	0-128-2
	r / r	\$MFTMirr	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	4096	0	0	1-128-1
	r / r	\$Secure:\$SDH	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	56	0	0	9-144-16
VIEW	r / r	\$Secure:\$SDS	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	900724	0	0	9-128-0
	r / r	\$Secure:\$SII	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	56	0	0	9-144-17
	r / r	\$UpCase	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	131072	0	0	10-128-1
File Name Search	r / r	\$Volume	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	2006.09.29 01:28:24 (GST)	0	48	0	3-128-3
Enter a Perl regular expression for the file names you want to find.	d / d	./	2008.04.12 03:04:44 (GST)	2008.04.13 12:29:07 (GST)	2008.04.12 03:04:44 (GST)	160	0	0	5-144-21
	r / r	AUTOEXEC.BAT	2006.09.04 16:06:12 (GST)	2006.09.04 16:06:12 (GST)	2006.09.29 01:28:29 (GST)	0	0	0	27-128-1
	r / r	boot.ini	2008.06.01 21:38:25 (GST)	2008.04.13 10:44:11 (GST)	2008.06.01 21:38:25 (GST)	224	0	0	28-128-3
	✓ - / r	bootex.log	2008.01.07 09:55:31 (GST)	2008.01.07 09:55:31 (GST)	2008.01.07 09:55:31 (GST)	2892	48	0	17
	d / d	carbt/	2008.01.13 17:22:42 (GST)	2008.04.07 12:35:32 (GST)	2008.01.13 17:22:42 (GST)	272	0	0	66417-144-1
	d / d	COMPONENTS/	2006.09.29 01:29:48 (GST)	2008.04.07 12:35:33 (GST)	2006.09.29 01:29:48 (GST)	352	0	0	29-144-1
	r / r	CONFIG.SYS	2006.09.04 16:06:12 (GST)	2006.09.04 16:06:12 (GST)	2006.09.29 01:30:26 (GST)	0	0	0	131-128-1
SEARCH	r / r	Convidado58499.pgn	2008.03.24 11:37:43 (GST)	2008.03.24 11:37:43 (GST)	2008.03.24 11:37:43 (GST)	592	0	0	65848-128-1
	r / r	Convidado94905.pgn	2008.03.31 10:50:28 (GST)	2008.03.31 10:50:28 (GST)	2008.03.31 10:50:28 (GST)	499	0	0	65474-128-1
ALL DELETED FILES	d / d	Dev-Cpp/	2008.01.23 12:01:02 (GST)	2008.04.13 11:39:35 (GST)	2008.01.23 12:01:02 (GST)	56	0	0	78065-144-5
EXPAND DIRECTORIES									

File Browsing Mode

Will not figure out Steganography. Not deleted, nor special files (or streams)

& Anti-Forensics ?

2) Keyword Search



Will not figure out steganography, only if the hidden content is in plain text or was not encrypted. Yet the investigator must know exactly what he is looking for.

& Anti-Forensics ?

3) Low level analysis regarding Slack Space

The screenshot displays a forensic analysis tool interface with a green header bar containing tabs: FILE ANALYSIS, KEYWORD SEARCH, FILE TYPE, IMAGE DETAILS, META DATA, DATA UNIT, HELP, and CLOSE. Below the header, there are navigation buttons: PREVIOUS, NEXT, EXPORT CONTENTS, and ADD NOTE. The main content area shows the following information:

Cluster Number: 31337
Number of Clusters: 69
Cluster Size: 4096
Address Type: Regular (dd)
Lazarus Addr:

Clusters: 31337-31405
Status: Allocated
[Find Meta Data Address](#)

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report)
File Type: data

ASCII Contents of Clusters 31337-31405 in sda1-0-0

```
m..2...S.v]...q.m...(.P..).#z...f.$.:C.z...{gU...as...a.F..MY%.v.a...y...y...Z.b...g.#...q.7...dV...HL.6...
...[L...C...M...GT...#...R...Bz]...*9z.r%...*I%[6...b1...$.*C...t>...f>.(W.W...?...[I.Q.4Z...p.6.b.1...
.S.c+xf/W...B...Zy...S:
...{...9...-2.A.7...eSh"o...V...o3...cCr...B.fc.); D...0...|...e...-DTG...T.9W...W...0...k-$..L.x5...[...j]...1b5...3..W.l.0"...
$.n.h.m.d)7...t2...6...m...uM...q...yBN.Y...{...UI...N)...1...I...M...+...G.GosX...Y...T...N...
S...7B...K.../c@...9...<.(M.z.s.u.S...M...l...l...).>...Q...z.C/U...p...z...*E]V[...42.&...96.Y...a...l.x7.%..4'.f]...?..kSF]..4"...6xS...pVe...$...C...
u:l.Zh.v...7...>...2...h.c...j...7...<b.N...e.h6.W.G...p...f...v..._...a)oSu...2Dw...{W...#PMS...2YO...EM.k&..h2...Yh...d[Ist...e...$
#...
.{.H[...6...wMB.U...g...).y.R2..W(...3...[O.<C...>...Xs...9K.../b...vuS...I...X)Ku6..I.4..J...JW.Z..q...8.9...D.S...9h]..4...
```

Because non-allocated space doesn't have Metadata, It's hard to tell if contents are from previous files or from "bad content".

& Anti-Forensics ?

So, the “bad” content is hidden inside the “good” files. Could it be even a whole another file system sitting on top of the “good” files.

In comparison to Slack Space usage, hiding content in “good” files is better idea because of “instable” situation of the Slack Space.

r / r	EULA/	2006.09.29 01:33:15 (GST)	2008.04.07 12:35:32 (GST)	2006.09.29 01:33:15 (GST)	56	0	0	1015-144-5
d / d	GTK/	2008.02.25 14:50:04 (GST)	2008.04.07 12:35:32 (GST)	2008.02.25 14:50:04 (GST)	56	0	0	112982-144-5
r / r	hiberfil.sys	2008.04.10 08:39:12 (GST)	2008.04.10 08:39:12 (GST)	2008.04.10 08:39:12 (GST)	2146484224	0	0	31794-128-2
d / d	I386/	2006.09.29 01:36:48 (GST)	2008.04.07 12:35:32 (GST)	2006.09.29 01:36:48 (GST)	56	0	0	1052-144-7
r / r	Io.SYS	2006.09.04 16:06:12 (GST)	2006.09.05 09:42:32 (GST)	2006.09.29 01:36:48 (GST)	0	0	0	7797-128-1
r / r	MSDOS.SYS	2006.09.04 16:06:12 (GST)	2006.09.04 16:06:12 (GST)	2006.09.29 01:36:48 (GST)	0	0	0	7798-128-1
d / d	MSOCache/	2006.09.29 01:36:48 (GST)	2008.04.07 12:35:32 (GST)	2008.02.26 14:14:15 (GST)	256	0	0	7799-144-1
r / r	NTDETECT.COM	2004.08.10 16:00:00 (GST)	2006.09.29 02:03:43 (GST)	2006.09.29 01:37:26 (GST)	47564	0	0	7859-128-3
r / r	ntldr	2004.08.10 16:00:00 (GST)	2006.09.29 02:03:43 (GST)	2006.09.29 01:37:26 (GST)	250032	0	0	7860-128-3
r / r	pagefile.sys	2008.04.10 08:39:11 (GST)	2008.04.10 08:39:11 (GST)	2008.04.10 08:39:11 (GST)	2145386496	0	0	130-128-2
d / d	Program Files/	2008.04.12 10:26:28 (GST)	2008.04.13 12:43:33 (GST)	2008.04.12 10:26:28 (GST)	56	0	0	7861-144-6
r / r	putty.exe	2006.12.12 10:04:26 (GST)	2008.04.13 12:52:07 (GST)	2008.04.13 10:27:13 (GST)	421888	0	0	8841-128-3
d / d	Python24/	2008.02.25 14:55:22 (GST)	2008.04.07 12:35:32 (GST)	2008.02.25 14:55:22 (GST)	56	0	0	115728-144-5
d / d	Python25/	2008.02.27 14:50:31 (GST)	2008.04.13 10:26:39 (GST)	2008.02.27 14:50:31 (GST)	56	0	0	70456-144-5
d / d	raide/	2008.01.14 20:16:29 (GST)	2008.02.27 14:51:47 (GST)	2008.01.14 20:16:29 (GST)	488	0	0	83450-144-1
d / d	RECYCLER/	2008.01.06 20:23:28 (GST)	2008.04.09 14:39:15 (GST)	2008.01.07 06:10:30 (GST)	328	0	0	8743-144-1
r / r	statusclient.log	2008.01.09 20:56:50 (GST)	2008.04.12 22:30:25 (GST)	2008.01.09 20:56:50 (GST)	7	0	0	53664-128-1
d / d	SUPPORT/	2006.09.29 01:41:59 (GST)	2008.02.27 14:51:47 (GST)	2006.09.29 01:41:59 (GST)	144	0	0	15357-144-1
r / r	SWSTAMP.TXT	2006.09.27 00:07:39 (GST)	2008.02.08 06:25:56 (GST)	2006.09.29 01:42:05 (GST)	440	0	0	15370-128-1

Update: Tool: Slacker from the Metasploit project

How to deal with the visible filesystem

One Approach that boosts analysis: Hashes Database

Good (Ignore) Database: Will automatically exclude from the “analysis” all well known files that are equal in default Windows and other software's installation.

Bad (Alert) Database: Will point all the files already catalogued as malware (such as virus, trojans, spyware, rootkits, backdoors, whatever).

NIST NSRL Database: Collection of good and bad hashes.

500 MB of hashes

<http://www.nsrl.nist.gov/>

& Anti-Forensics ?

The more the PC is used, more we find “Unknown” files, either created or modified by the user.

These files most of the times follow well known file formats.

Interpretors who read and write these files implement this “file format standard”. The art of “hiding content” inside these files rely on finding “holes” on the file formats standard in order to achieve data storage without the interpreter's knowledge.

But, Still: It's only efficient for **known files** :-)

Detection

Approaches regarding detection are public, but normally they address the “rendering DATA” (tons of techniques and papers)

Previous installed tools like Tripwire would detect that monitored files are being “changed” by Steganography techniques that touch “visible files” (this presentation’s approach).

Detection

When analyzing a “suspicious” file a good approach would be entropy analysis:

Ftimes' XMagic (useful to do block-based entropy calculations):

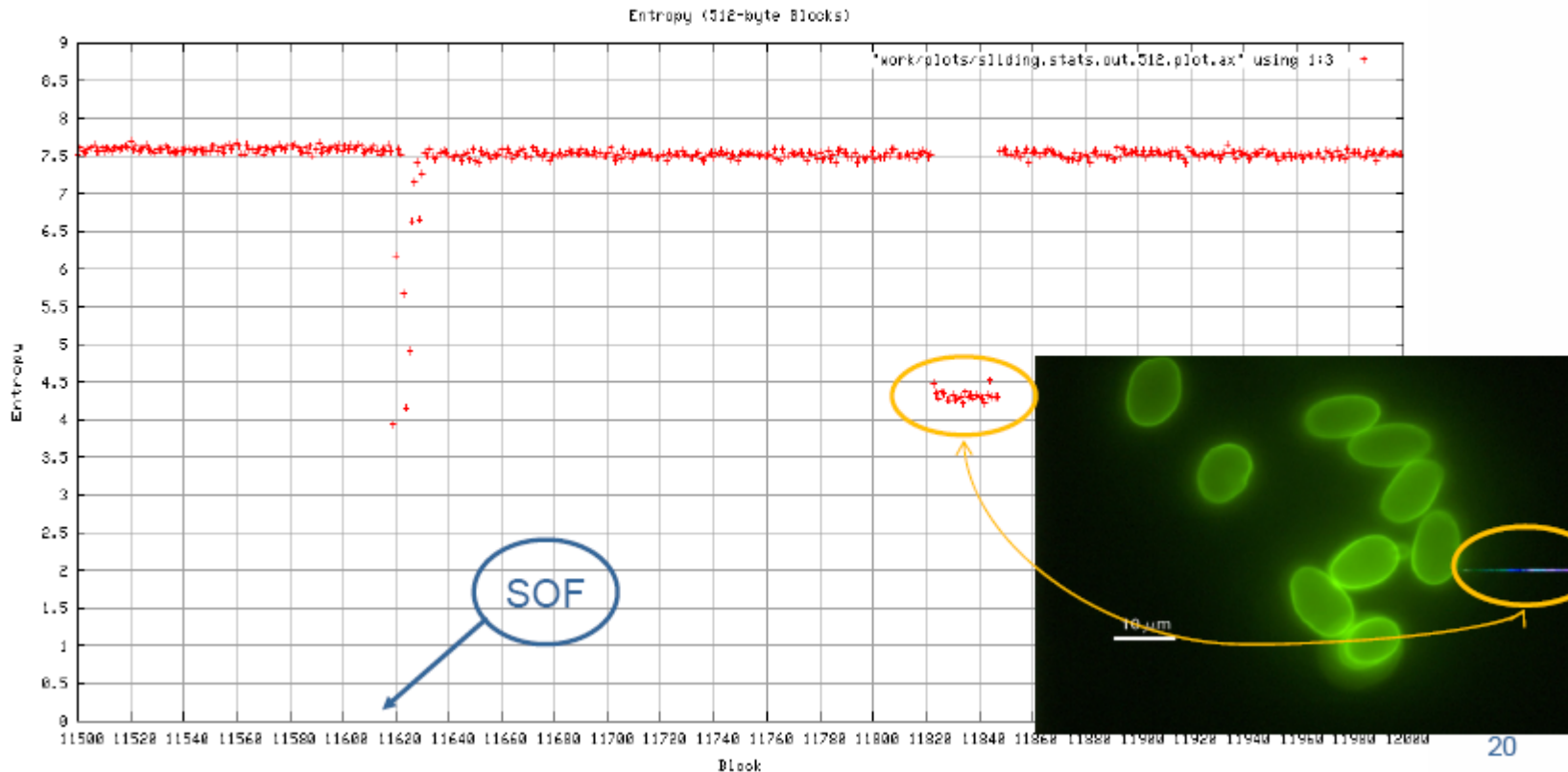
<http://ftimes.sourceforge.net/FTimes/XMagic.shtml>

Big “salts”, depending the offset that is being touched (checking it with the file format standard) should be suspicious

Detection

XMagic and FTimes: Same technique used for Advanced File Carving could be used to Stegdetection

This sliding entropy graph shows the start of the JPEG image at block 11619. The graph also reveals a drop in entropy at block 11820.



From Korelogic.com's presentation, winners of DFRWS(Digital Forensics Research Conference) 2006 File Carving Challenge

Conclusion

In real life cases, it's **very** hard to find previous installed tools that keep hashes of hard drive's files.

To the Computer Forensics Examiner, receiving evidence and "approaching" `_visible_` files is not easy, because most of the times he needs to rely on "non-forensics-driven" Interpreters.

There is a big lack in "File Formats" analyzers and awareness needs to be raised.

.... ?

Questions?

Run, run :P

Thanks a lot :)

Domingo Montanaro
<conferences@montanaro.org>