## Steganography Primer

I)ruid <druid@caughq.org>

#### What is Steganography?

Steganos (covered) graphein (writing)

Hiding a secret message within a covermedium in such a way that others can not discern the presence of the hidden message

Hiding one piece of data within another



#### Steganography vs. Cryptography

Steganography's goal is to keep the presence of a message secret, or hide the fact that communication is taking place

- Cryptography's goal is to obscure a message or communication so that it cannot be understood
- Steganography and Cryptography make great partners. It is common practice to use cryptography with steganography



#### **Steganography Terms**

- <u>Cover-Medium</u> The medium in which information is to be hidden. Also sometimes called "cover-image/data/etc."
- Stego-Medium A medium within which information is hidden
- Message The data to be hidden or extracted
- Redundant Bits Bits of data in a covermedium that can be modified without compromising that medium's integrity



#### Simple Steganography Example

A simple example of a steganographic system would be to use a given letter of each word in the covermedium to convey your s33kr3t message:

Susan eats truffles. Under pressure, that helps everything before Owning Major Bullwinkle.

"Set Up the b0MB"



#### **Traditional Methods of Steganography**

Concealed Tattoos (under body hair)<sup>1</sup>

- Using newspaper articles / want-ads (with methods like our previous example)
- Invisible / disappearing ink on the back of other script or in-between lines

#### **Microdots**

Spread-spectrum radio communications



Modern Methods of Steganography

Use the properties of the media itself to convey a message

### Digitally embedding messages in other media, such as:

- 🕾 Plain Text
- B Hypertext
- 🕾 Audio / Video
- Still Imagery
- **Network Traffic**



#### Plain Text

Steganography with plain text can be done a number of different ways:

Solution Selected characters or words from a specially-crafted cover-text (like our example)

Introducing white-space characters that a text viewer won't display



#### Tool: snow

snow is used to conceal messages in ASCII text by appending white-space to the end of lines. Because spaces and tabs are generally not visible in many text viewers, the message is effectively hidden from casual observers.

- Snow exploits the steganographic nature of whitespace
- **W**Uses the ICE encryption algorithm



#### Hypertext

Steganography with hypertext can also be done a variety of different ways:

Similar methods as Plain Text
 Hypertext comment notation (view-source)
 Arrangement of content on a given page
 Presence or absence of content elements (images, phrases, etc.)



#### Audio

Messages can be hidden in common audio formatted files or the audio itself. Some methods include:

- Transmitting a message in the human-inaudible audio spectrum
- Assigning musical notes values and then creating or adding to a musical score, either played or on sheet music
- Digitally embedding a message into an audio file



#### **Digitally Embedding**

 Digitally embedding a message in a covermedium usually involves two steps:
 Identify the redundant bits of a cover-medium
 Deciding which redundant bits to use and then modifying them
 Generally, redundant bits are likely to be

the least-significant bits of each byte of the cover-medium



# Digitally Embedding in Audio Audio is a very inaccurate data format Slight changes will be indistinguishable from the original to the human ear

- In Audio, you can use the least-significant bits of each byte as redundant bits
- Use the redundant bits to minimize the impact of changes



#### **Example: Audio Embedding**

Let's assume an audio file had the following 8 bytes of data in it somewhere: 180, 229, 139, 172, 209, 151, 21, 104

In binary, this would be:

- 10110100-11100101-10001011-10101100-11010001-10010111-00010101-01101000
- If we wanted to hide the byte value '214' (11010110), we use the least significant bit from each byte to hide our byte: 10110101-11100101-10001010-10101101-11010000-10010111-00010101-01101000

The changes result in the following bytes, which are so close to the originals that the difference will be inaudible:

Modified:181, 229, 138, 173, 208, 151, 21, 104Original:180, 229, 139, 172, 209, 151, 21, 104



#### **Tool: S-Tools 4**

Steganography Tools 4 can operate on the following file types:

## ₩AV files using the method discussed Also operates on BMP & GIF files

## We'll use S-Tools to demonstrate hiding a message in a WAV file



#### **Embedding a message with S-Tools 4**

Using S-Tools is literally a drag-and-drop affair:





#### **Embedding a message with S-Tools 4**

Once a cover-medium is selected, you then drag your message file directly onto it to produce your stego-medium:







**Embedding a message with S-Tools 4** 

The waveform of each audio file is nearly identical, and there is no audible difference:



🀠 🗄







**Extracting a message with S-Tools 4** 

To extract a message from a stego-medium, drag it into S-Tools and right click on it:

🔒 anomaly2.wav			
Save Save Save Save Save Save Save Save			
	Revealing from anomaly2.wav	×	
Reveal	Passphrase: *****	OK	
	Verify passphrase: *****	Cancel	
	Encryption algorithm: Triple DES	Help	🚬 Revealed Archive 📃 🗆 🗙
			Revealed files:
			Name Size Message.txt 139



#### Video

Like Audio, messages can be hidden in common video formatted files or the video itself. Some methods include:

The presence or absence of objects in the recorded environment, similar to the method described in Hypertext

- ⊠Visual Clues such as:
  - $\mathbb{R}$  Hand or foot positions
  - Eye-blink code

Digitally embedding a message into a video file



#### **Still Imagery**

Like Audio and Video, data can not only be hidden in the bits of the file, but in the visual itself. Some methods include:

Using slightly different colors to hide a message
 Digital watermarking
 Digitally embedding a message into an image file



#### **Using Slightly Different Colors**

By using nearly identical colors, an image or message can be hidden in the visual of the imagery. A good example is the logo from the SNOW tool mentioned earlier:







#### Image Revealed!

By changing the color value for the second white value to something greater in contrast like dark green, the hidden image is revealed.







#### **Digital Imagery Basics**

Images vary between resolutions and size of color palette

The number of unique colors an image can display is represented in it's bits-per-pixel (BPP) value

 $\mathbb{Z}$  8 bits per pixel == 256 colors available

24 bits per pixel == 16,777,216 colors available



#### **Digital Imagery Basics**

Steganography using 8-bit images have a lot of hurdles to overcome

- Due to the limited number of color values a single byte can represent, a color-map is generally employed
- Because pixel values are mapped to colors in a table, a single bit change in the byte representing the pixel could have drastic visible effects in the image



#### **Digital Imagery Basics**

- 24-bit images inherently provide more space for embedding a message than an 8-bit image
- Each pixel is represented by three bytes, one byte for Red, Green, and Blue (RGB) values
- Changing one bit of a color value stored this way will result in a color who's value is extremely close to the original
- A 1024x768 24-bit image provides over 2 million pixels, three bytes each



#### **A Quick Note on Image Compression**

- Types of image compression are categorized as 'lossy' and 'lossless'
- Lossless compression allows the user to reconstruct the original image upon decompression
- Lossy compression, as the name implies, will loose some of the original images data



#### **Examples of Poor Cover-Images**

- Images with small colorpalettes
- ⊠ Large areas of solid colors









#### **Examples of Good Cover-Images**

Landscapes & Portraits
 Subtle color variations
 Rich, contrasting, variety of colors



© 2004 Computer Academic Underground

punicht 1995 philoPmit.ed

#### The Best Type of Cover-Image

h4wt n3kk1d chix covered in video game gear.





#### **Using Least-Significant Bits**

Similar to the audio method discussed earlier, you can use the least-significant bits of each byte to embed a message

- Solution States and the set of th
- ✓ Using a 1024x768 pixel image, you can hide up to 2,359,296 bits (or 294,912 bytes)

Compressing your message before embedding allows for a relatively large message **Example of Embedding in Imagery** 

An image may have the following three pixels (9 bytes) in it somewhere:

(01010010, 10010110, 10100100) (10110100, 10010001, 01001110) (10110110, 00101110, 11010001)

If we wanted to hide the letter "A" (131 or 10000011), we would use the leastsignificant bits of each byte: (01010011, 10010110, 10100100) (10110100, 10010000, 01001110) (10110111, 00101111, 11010001)



#### **Identifying Redundant Bits**

Many different techniques exist for identifying which redundant bits to use for embedding a message:

- Using a pre-determined formula or key to use some of them
- Random distribution
- Section 20 Section 20
- Cover-image analysis in an attempt to defeat Steganalysis



#### **Tool: Outguess**

Outguess is a universal steganographic tool that allows the insertion of hidden information into the redundant bits of data sources.



#### **Tool: Outguess**

Designed so that the core of the tool is independent of data types

- Cover-data type is irrelevant, provided there is a "handler" for that type of data to identify the redundant bits
- When the redundant bits are identified, Outguess' core can do the rest



#### Why Outguess?

- Preserves cover-medium statistics in order to defeat detection by statistical analysis
- For JPEG images, Outguess preserves statistics based on frequency counts
- Before embedding data into an image, Outguess determines the maximum message size that can be hidden while still being able to maintain statistics
- Other technical cover-medium-analyzing goodness



#### **Outguess Demo**



#### Network Steganography

Modify Existing Network Traffic

- Create new traffic emulating legitimate traffic
- Make use of otherwise unused or unneeded fields in network protocol headers
- Modify protocol header values that are being used
- Depending on data type, even the payload of the network traffic could be used



Network Steganography Don'ts
Avoid using "optional" header fields
Avoid using headers that are likely to change in transit



#### **Embedding in Network Traffic**

Various types of network traffic provide for various types of hidden communications

- Embedding within TCP session can provide for two-way communications
- Multicast UDP or ICMP traffic could be used for a steganographic broadcast



#### **Examples of Usable Header Fields**

Using the IP header's packet ID field within a single session

 Using TCP SYN packet's initial sequence number (ISN) across multiple sessions
 Various types of ICMP have undefined header space between fields



Examples of Usable Payloads CMP Echo Request/Reply data Video or Audio traffic RTP Other VoIP protocols



#### **Tool: StegTunnel**

StegTunnel establishes a full-duplex steganographic communications tunnel using a legitimate TCP session generated by the client host

 Uses a TCP connection's handshake SYN and SYN/ACK packets to establish a "keyed" communication session
 Uses the IP header's IPID field to transmit it's payload



#### **Tool: hcovert**

Uses part of a GET request within HTTP's payload to convey it's message

- Obscures the message by converting it to hexadecimal values
- Sends the message via network socket to the web server
- Receives the message by parsing the web server's log files



#### hcovert Demo



#### **Steganography Tips and Tricks**

Always encrypt your message prior to using steganography to hide it

- Hide your stego-medium among other media of the same type, or in a unsuspicious location
- Destroy the original cover-medium so that the only version of it that remains is the stego-medium



#### **Defeating Steganography**

 Because of steganographic systems' invasive nature, they leave detectable traces in a cover-medium's characteristics
 This allows an eavesdropper to detect media that has been modified, revealing that secret communication is taking place

The integrity of the information is not degraded, however it's hidden nature is revealed, thus defeating the main purpose of steganography



#### **Steganalysis**

The processes and methods of attempting to defeat steganography through analyzing potential stego-mediums for the traces of steganographic modifications.

Steganalysis is the Yin to Steganography's Yang.



#### **Additional Reading**

```
Hide & Seek: An Introduction to Steganography:
  http://niels.xtdnet.nl/papers/practical.pdf
Exploring Steganography: Seeing the Unseen:
  http://www.jjtc.com/pub/r2026.pdf
Covert Channels in the TCP/IP Protocol Suite:
  http://www.firstmonday.dk/issues/issue2 5/rowland/
CAU E-Zine #22 – Stego/Crypto Hunt Challenge:
  http://www.caughq.org/zines/cau-0022.txt
RFC 791 – Internet Protocol:
  http://www.faqs.org/rfcs/rfc791.html
RFC 792 – Internet Control Message Protocol:
  http://www.faqs.org/rfcs/rfc792.html
RFC 793 – Transmission Control Protocol:
  http://www.faqs.org/rfcs/rfc793.html
```



#### Some Available Tools

 $\mathbb{Z}$  Tools we've discussed:

Snow: http://www.darkside.com.au/snow/

- S-Tools 4: http://www.spychecker.com/program/stools.html
- ⊠ Outguess: http://www.outguess.org
- StegTunnel: http://www.synacklabs.net/projects/stegtunnel/
- A hcovert: http://sourceforge.net/projects/hcovert/

Tools Lists:

Market Ma

Market Security/Stegtools.htm 🗠 🕅





