Global Information Assurance Certification Paper

Thomas M. Blake

Steganalysis

Or

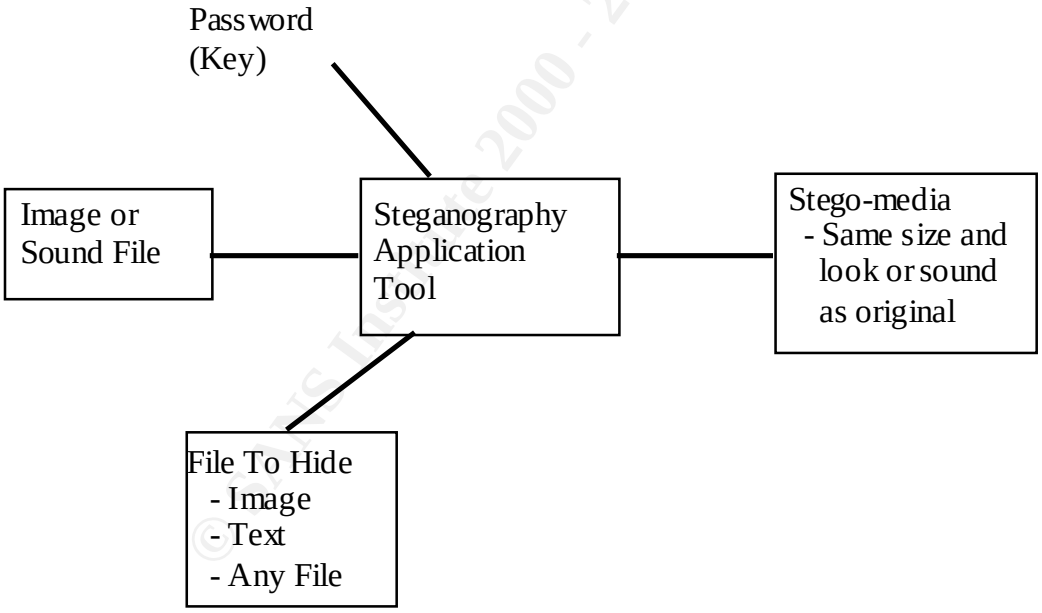
"Is Ralph In Marketing Selling Company Secrets on Our Web Page?"

Abstract

This paper will provide a top-level overview of steganography detection. Steganalysis techniques and steganography signature patterns will be discussed at an overview level. It will discuss why steganalysis is desired and provide some insight into some basic detection and protection techniques that can help defend your information and networks.

Steganography

Steganography is the science of hiding information within other information. There are many techniques and applications that the term applies to, but for this topic, steganography means hiding a data file within an image or sound file. This "hiding" is done in a manner that does not alter a human's perception of the picture or sound. There is much in current literature about steganography tools and how they work (1). The basic methodology behind steganography is depicted here:



The simple description is that an image or sound file is passed through a steganography application program, a secret password or key is used to embed another file into the original, and a new file is created that is indistinguishable from the original. At least a human cannot see or hear the difference. An important point here is that not only is data hidden within the new file, it is very difficult to even determine whether or not there is any hidden information in the file. Another important point to be made is that

another step can be added to encrypt the file to be hidden before it is merged into the cover file.

There are two different classes of steganography techniques commonly available (2). The first and simplest is image domain. This category of tool does bit manipulation on the carrier image, usually embedding data in the least significant bit (LSB) or in the noise. The simplest tools replace the LSB of each pixel with one bit of the data to be hidden. This method is usually used with a Bitmap (BMP), or GIF format cover image. The drawback here is with the large size of the cover file required compared to the hidden file. For a simple case, you could only hide a file $1/24^{th}$ the size of the cover image if there are 24 bits/pixel. It is also common to use images with 8 bits/pixel. This allows smaller cover file sizes. Software applications to do this simple steganography are readily available and free. I quickly downloaded S-Tools, a Windows based application with an easy to use drag and drop user interface (3). It was very easy to drag a cover BMP file to the S-Tools window, then drag a smaller JPEG file on top of it, type in a password and create a new BMP of the same filesize, with the JPEG hidden file inside. A choice of encryption algorithms (IDEA, DES, and Triple DES) was also available. I compared the original BMP with the new BMP at high magnification levels and was not able to discern any difference.

The second class of steganography tool operates in the transform domain. Instead of replacing bits such as the LSB, these methods apply transform algorithms to complex parts of an image. These more complex techniques apply things like a discrete cosine or wavelet transforms and will act upon properties of the image such as luminance or the color palette. These methods will allow more hidden data in a carrier file. Many of the tools available can hide a file approximately 30% the size of the carrier.

The techniques are closely related to digital watermarking. In this technology, similar steganographic techniques are used to embed hidden data into a file for copyright or identification purposes. The main difference with watermarking is that it is designed to be found. This paper deals with the type of steganography that is meant to remain covert.

What, Me Worry?

So why do I worry about this? I don't see any problem. Like "Who's on first?" that is the problem. The biggest problem is that the very existence of the hidden data is hidden. Steganography can be used as a covert communication channel into or out of your network, and it is very difficult to determine that this channel exists. The steganography tools are out there, easily and inexpensively obtained, and very simple to use. A search showed 26 steganography applications available running on DOS, Win 9x, UNIX, Linux, and JAVA. Carrier files include PCX, BMP, GIF, JPEG, PICT, WAV, MP3, and PDF formats. Many were available for immediate download as freeware or shareware. The German firm Demcon sells a software package called Steganos Security Suite and has sold 100,000 copies (4). Steganos applies encryption to the file to be hidden and is touted as a privacy tool. It is marketed as a way to restrict access to your data, used by businessmen with notebook computers, students with shared computers, and even as protection for the "always-on" home computer. I find this a little hard to buy into. Certainly there is a need for privacy, easily provided by Pretty Good Privacy (PGP) and other encryption tools, but the need to hide the fact that the data exists? I do not

believe that in a business or government setting, there is much legitimate need for this type of steganography.

The technology is just recently moving to the attention of the mainstream media. USA Today (5) reported that nefarious terrorist Osama Bin Laden and his organization have used steganography to pass bombing plans, hidden in email attachments. Bill Hancock, chief of security for the world's largest internet hosting company, Exodus Communications, and former National Security Agency (NSA) computer scientist says he has been involved in six steganography related cases (4). One dealt with stealing airplane plans from a French aerospace company and the other five are classified. In another interesting report (6), Wetstone Technologies, while involved in steganography research for the Air Force Research Laboratory (AFRL) studied random images on the web. They watched a picture of a sewing machine being auctioned on ebay as pixel patterns changed every few days. It is not proof of anything, but is certainly a strange thing.

So we now know the tools are available, easy to use, free in most cases, and at least 100,00 people felt the need to spend money for a better version. It is easy to post the pictures on a web server or email pictures or sound files as attachments. But are people really using this method for espionage or even to trade pornographic images? One reason a person or group might use steganography is the relative obscurity provided to the receiver and sender of the information. If you suspect an image server of displaying steganographic images and lots of people download, most innocent web users, it will be very hard to determine who is getting and decoding the hidden files. Suppose you suspect a person of being the receiver of hidden files; if he is diligent about surfing lots of web sites, it will be very difficult to figure out what site is displaying the hidden information. So now as I look at the corporate web server and all of our email attachments, maybe I am a little worried.

Steganalysis

Steganalysis is the relatively new science of discovering, decoding, and/or rendering useless covert messages hidden in a carrier file (7). Since one of the main reasons to use steganography is to conceal the fact that a message is being hidden, just being able to figure out if steganography is being used is an important part of defeating it. Decoding a hidden message will be a very tough problem with analogies to cryptanalysis. A brute force method to decode a covert message where you don't know the encoding method, the format of the hidden message, and don't have access to an original version of the carrier file will be very complex and resource intensive. Even if decoded, the hidden message is probably again hidden by an encryption algorithm. It may be possible though, to use encryption as a type of signature to detect hidden data. Looking for the spectrum of encrypted data, which should appear random for a good algorithm, might provide information. The destruction of the hidden data is the easiest part of this problem. Image processing techniques can destroy many types of steganography coding.

Detection is usually going to be the first step. Just knowing that someone is using a covert communications channel into or out of your network is significant. For a corporate environment, the person's access can be terminated for a permanent solution. For a military situation, knowing that a covert channel exists can be used as part of an Indications & Warnings (I&W) process. A steganography detector can function as

another network information warfare sensor. It is always good to know things that your adversary doesn't think you know. Detection is usually broken down into two areas: signature detection and blind detection (2,6). Work at George Mason University (GMU) has concentrated on detection signatures for the various steganography algorithms. There has been a lot of research in using the known steganography tools and studying the resulting altered images for noticeable changes. Many of the tools do produce a recognizable change in the altered file. GMU is working to automate the process so a large number of files can be worked on electronically. For example, the S-Tools image domain steganography tool that I did some quick experimenting with, does leave a recognizable signature. S-Tools works by reducing the number of colors of the cover image to 32, but expands them over several color palette entries. If the palette is then sorted by luminance, blocks of colors appear to be the same, but actually have a one bit variance. This type of variance pattern would be extremely rare in a non-altered image. So if an image contains this pattern, it is fairly certain that it contains covert data. GMU has demonstrated that many image domain as well as transform domain steganography tools have similar signature characteristics. However, the "security through obscurity" principle applies here. If an unknown steganography algorithm with unknown signature were used, signature detection would not catch it. As of now, automated scanning tools are not mature.

The second method, blind detection focuses not on the steganography algorithm, but on the patterns normally occurring in digital images. This is sort of a reverse signature algorithm. In this area, the AFRL Information Directorate has funded research to develop these types of algorithms (8). In this research, Wetstone Technologies created a large Steganography Index Library (SIL) from various image types and steganography algorithms. The image formats covered were:

- Unmodified raw image data (BMP, PGM, RAS, TIFF)
- Image data stored as pointers to a finite color palette (GIF, PNG)
- Lossy compressed image data (JPEG, Wavelet, Fractal compression)

A large number of these types of images were used to embed hidden files with four different steganography tools, both image and transform domain, to create the SIL. The SIL was then used as a testbed to study both the "clean" images and the images with hidden data. A number of unique characteristics of the clean images were discovered that do not occur in the embedded data images. A number of proprietary algorithms were developed and run against the SIL. The work looks promising and the algorithms have a very high probability of detecting an image that could not have been created by a normal digital image capture process (CCD camera, scanner). These images would then be classified as probably steganographically altered. The Air Force has not yet released the report on the performance of this prototype tool.

The next step in steganalysis would be decoding the detected file to see what is hidden inside. This is the hardest step, and no open literature was found pointing to any tools or real research in this area. There is one report in the USA Today article (5) about the NSA taking an alleged terrorist's computer and using supercomputers for a year to decode encrypted files. While the article is about steganography, it is unclear if they really mean that here, or are just mixing steganography and encryption interchangeably.

The last step in steganalysis, destruction, is actually the easiest. Most steganography altered files are destroyed if they are changed any further. For example, I took my image domain S-Tools created BMP files and experimented with some compression and image processing techniques. I converted the file to a JPEG and back to BMP. S-Tools now could not recover the hidden data. I opened the BMP with Adobe PhotoShop Pro and tried some simple image manipulation (stretching, twisting, re-sizing). While the visible impact was negligible, S-Tools could not recover after each operation. The image domain tools are not very robust to these techniques but the transform domain tools that merge the hidden information with integral properties of the carrier image are more robust (2). It would be an inefficient steganalysis technique to attempt some image processing on every image into or out of your network, so detection is important.

There may be another twist to steganalysis techniques. Privacy rights groups would argue that steganography is important for anti-censorship and free speech reasons. Some steganography advocates have discussed a developing a virus that could infect image files and make subtle changes that might be detected as a steganographically altered image (9). This would protect people using steganography as all or a large number images out there would be indistinguishable from the ones with covert data.

What Can Be Done Now

In the unclassified world, there are not yet automated tools available for you to protect your networks and information. The Air Force and Wetstone Technologies work may eventually lead to a commercial product and work in digital watermarking technology is furthering the science. Some watermark identification tools already have automated software robots or "bots" scanning the web looking for copyright infringement. These may hold promise for some relatively near term developments.

But what should we be doing right now? Notwithstanding the lack of automated steganalysis tools, there are some common sense things that can be done to protect ourselves. The first is policy and enforcement. Steganography software should not be allowed on the company computers. It should be relatively simple to check computers for the common, widely available steganography tools. Unless part of a person's job involves steganography, there are not many "good" things that can happen with these tools.

Another thing to do is some simple pattern analysis of users. A simple email analysis would look for the type of attachments being sent. Are there lots of large images? Does the email content match the image context? Is what looks to be the same image sent numerous times? This would be suspect behavior.

To monitor the web server to detect an inside person sending covert data out, the images on the server could be analyzed on a regular basis. Saving the images at different times and searching for differences. A poor steganography tool might change the file modification date/time stamp. Seeing what looks like the same image with changing dates would warrant a closer look.

To monitor your web server for outside users receiving covert data, how the server is being accessed could be monitored. If there are users downloading just images and not associated text, that could be a pointer to suspect activity. Are user activities within the context of what is considered normal use of your web server?

All the above things are sort of common sense policy or pattern analysis techniques. If you are operating in a very secure or paranoid (can't have too much paranoia!) environment, there are more drastic measures one could take. Disallowing any email attachments would close one channel. A resource intensive protection scheme could do some visually imperceptible processing (your own steganography?) on every image into or out of your network. This could stop a lot of steganography traffic. Some of the more robust algorithms would still survive. This is probably not practical unless combined with some type of detection scheme.

In conclusion, I hope I provided a better understanding of what your networks need to be protected against. Although there is currently no easy way to defend against steganography attack methods, the information here may help an administrator figure out if there is a problem.

Bibliography

1) Krinn, Jeremy. "Introduction to Steganography." 26 Jun 00.
URL: http://www.sans.org/infosecFAQ/covertchannels/steganography.htm (30 Jan 01).

2) Johnson, Neil F, and Jajodia, Sushil. "Steganalysis of Images Created Using Current Steganography Software." Lecture Notes on Computer Science, Vol. 1525, Springer-Verlag, 1998, pages 273-289. Second Information Hiding Workshop, Portland, Oregon. 15-17 Apr 98.

3) Unknown. "S-Tools Download." 8 Dec 99.
URL: http://mila.ljudmila.org/matej/pgp/stegodl.html (1 Feb 01).

4) Shannon, Elaine. "Hiding In Plain Sight." On Magazine. March 2001.

5) Kelley, Jack. "Terror Groups Hide Behind Web Encryption." USA Today. 6 Feb 01.
URL: http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen.htm (9 Feb 01).

6) Hosmer, Chet and Gordon, Gary. "Steganography Detection and Recovery Toolkit." Wetstone Technologies Inc. Air Force Research Laboratory Final Technical Report for Contract F30602-99-C-0210. March 2000.

7) Johnson, Neil F, Giordano, Joe, and Jajodia, Sushil. "Steganography and Computer Forensics: The investigation of Hidden Information." George Mason University, Center for Secure Information Systems. Technical Report CSIS-TR-99-10-NFJ. Oct 99.

8) Hosmer, Chet. "Securing Digital Integrity."
URL: http://www.wetstone.com/sdart.htm (1 Feb 01)

9) Hansmann, Fabian. "Steganalysis: Scanning the Web." 28 Aug 98.
URL: http://www.demcon.com/english/steganos/steganalysis.htm (30 Jan 01)

Last Updated: July 25th, 2017

Upcoming Training

SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
Community SANS Omaha SEC401*	Omaha, NE	Aug 14, 2017 - Aug 19, 2017	Community SANS
Virginia Beach 2017 - SEC401: Security Essentials Bootcamp Style	Virginia Beach, VA	Aug 21, 2017 - Aug 26, 2017	vLive
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
Community SANS Pasadena SEC401 @ NASA	Pasadena, CA	Aug 23, 2017 - Aug 30, 2017	Community SANS
Mentor Session - SEC401	Minneapolis, MN	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
Mentor Session - SEC401	Edmonton, AB	Sep 06, 2017 - Oct 18, 2017	Mentor
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Sep 11, 2017 - Oct 12, 2017	Mentor
Community SANS Albany SEC401	Albany, NY	Sep 11, 2017 - Sep 16, 2017	Community SANS
Community SANS Columbia SEC401	Columbia, MD	Sep 18, 2017 - Sep 23, 2017	Community SANS
Community SANS Dallas SEC401	Dallas, TX	Sep 18, 2017 - Sep 23, 2017	Community SANS
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS Copenhagen 2017	Copenhagen, Denmark	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Boise SEC401	Boise, ID	Sep 25, 2017 - Sep 30, 2017	Community SANS
Baltimore Fall 2017 - SEC401: Security Essentials Bootcamp Style	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	vLive
Community SANS New York SEC401	New York, NY	Sep 25, 2017 - Sep 30, 2017	Community SANS
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Sacramento SEC401	Sacramento, CA	Oct 02, 2017 - Oct 07, 2017	Community SANS
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Community SANS Charleston SEC401	Charleston, SC	Oct 02, 2017 - Oct 07, 2017	Community SANS
Mentor Session - SEC401	Arlington, VA	Oct 04, 2017 - Nov 15, 2017	Mentor
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event