THE JOURNAL OF
DIGITAL FORENSICS,
SECURITY AND LAW

Journal of Digital Forensics,
Security and Law

Volume 4 | Number 2 Article 1

2009

Detection of Steganography-Producing Software Artifacts on Crime-Related Seized Computers

Asawaree Kulkarni
Purdue University

James Goldman
Purdue University

Brad Nabholz
Purdue University

William Eyre
Purdue University

Recommended Citation

Kulkarni, Asawaree; Goldman, James; Nabholz, Brad; and Eyre, William (2009) "Detection of Steganography-Producing Software Artifacts on Crime-Related Seized Computers," *Journal of Digital Forensics, Security and Law*: Vol. 4 : No. 2 , Article 1.
DOI: https://doi.org/10.15394/jdfsl.2009.1055
Available at: https://commons.erau.edu/jdfsl/vol4/iss2/1

Detection of Steganography-Producing Software Artifacts on Crime-Related Seized Computers

Asawaree Kulkarni, James Goldman, Brad Nabholz, William Eyre
Department of Computer & Information Technology
Purdue University
Knoy 255, 401 N. Grant St.
W. Lafayette, IN 47907
{akulkarn, jgoldman, bnabholz, weyre} @ purdue.edu

ABSTRACT

Steganography is the art and science of hiding information within information so that an observer does not know that communication is taking place. Bad actors passing information using steganography are of concern to the national security establishment and law enforcement. An attempt was made to determine if steganography was being used by criminals to communicate information. Web crawling technology was used and images were downloaded from Web sites that were considered as likely candidates for containing information hidden using steganographic techniques. A detection tool was used to analyze these images. The research failed to demonstrate that steganography was prevalent on the public Internet. The probable reasons included the growth and availability of large number of steganography-producing tools and the limited capacity of the detection tools to cope with them. Thus, a redirection was introduced in the methodology and the detection focus was shifted from the analysis of the 'product' of the steganography-producing software; viz. the images, to the 'artifacts' left by the steganography-producing software while it is being used to generate steganographic images. This approach was based on the concept of 'Stego-Usage Timeline'. As a proof of concept, a sample set of criminal computers was scanned for the remnants of steganography-producing software. The results demonstrated that the problem of 'the detection of the usage of steganography' could be addressed by the approach adopted after the research redirection and that certain steganographic software was popular among the criminals. Thus, the contribution of the research was in demonstrating that the limitations of the tools based on the signature detection of steganographically altered images can be overcome by focusing the detection effort on detecting the artifacts of the steganography-producing tools.

Keywords: steganography, signature detection, file artifact detection.

1. INTRODUCTION TO STEGANOGRAPHY

The term steganography is derived from the Greek words steganos, which means 'covered', and graphein, which means 'to write' (Singh, 1999).

Steganography is the art and science of hiding information. The term steganography can also be used to refer to the hidden information itself. Steganography facilitates secret, undetected communication and refers to hiding information in information (Katzenbeisser and Petitcolas, 2000).

1.1. Steganographic Techniques

Steganography can be hidden within numerous types of files, most commonly image files. The images in which the secret information is hidden are called carrier files or cover images. The resultant files which contain the hidden information are referred to as stegoed files. Various techniques of steganography include LSB (least significant bit) steganography (Wayner, 2002), manipulation of the Discrete Cosine Transform (DCT) function (Acharya and Tsai, 2005), and the append technique (Goudy, 2007). The algorithms which are used for hiding information in the carrier files use different techniques to hide information in different types of files. These algorithms act in known and predictable ways. The embedding action of the algorithms often leaves image artifacts in the cover images ("Stego Suite," 2006).

The artifacts discussed in the first research project are specific only to the images. The term ‘file artifact’ is used to refer to the evidence left by the steganography-producing software application on the host system that generated the stegoed files and is discussed in the second research project. Image artifacts are observable anomalies in various characteristics of the image which indicate action of steganographic embedding software. Artifacts consist of changes to associated information and are not necessarily detectable by analyzing only the pixilated composition of the image (Wayner, 2002).

Embedding applications employ these steganographic algorithms in various ways and to varying degrees of effectiveness and stealth (Wayner, 2002). Some applications leave signatures, in the images in which they have hidden information. Signatures are means of associating the image with a specific steganographic application in order to identify which steganographic application must be employed for image extraction. Signatures are also detectable by steganalytic software. A detected signature almost always indicates the presence of steganography ("StegAlyzerSS," 2006). Shorter signatures are more likely to be present in non-stegoed images by random chance, thereby producing false positives in signature-based detection software.

1.2 Detection

Various techniques are employed to detect the presence of steganography. Many detection applications are written to detect steganography based on the knowledge of the steganographic application which embedded the information. These are signature-based detection schemes (Jackson, Gunsch, Claypoole, and

Lamont, 2003). A signature-based detection scheme uses knowledge of the signatures to identify suspect images. A steganographic embedding application implements an embedding algorithm. The application may leave a signature, however the algorithm may create one or more image artifacts. These artifacts can then be detected in the image regardless of the application which employed the method that the algorithm is written to implement ("Stego Suite," 2006).

2. DESCRIPTION OF THE RESEARCH

The question of concern is whether steganography is being used by criminals? The research targets the population of steganography users with a criminal background (represented by the shaded area in figure 1). The usage of steganography may or may not be made for committing or providing assistance to crime. As shown in figure 2, the research consisted of two projects. The first project consisted of the detection of the presence of steganography in carrier images found on the Web and the second project consisted of detecting if steganographic applications had been installed (and/or used) and whether they left behind file artifacts on computers that were seized during criminal investigations of various types.

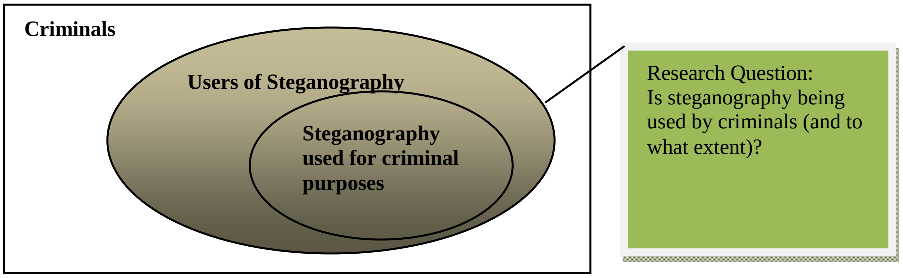
Criminals
Users of Steganography
Steganography used for criminal purposes
Research Question:
Is steganography being used by criminals (and to what extent)?

Figure 1: Target population of the Research

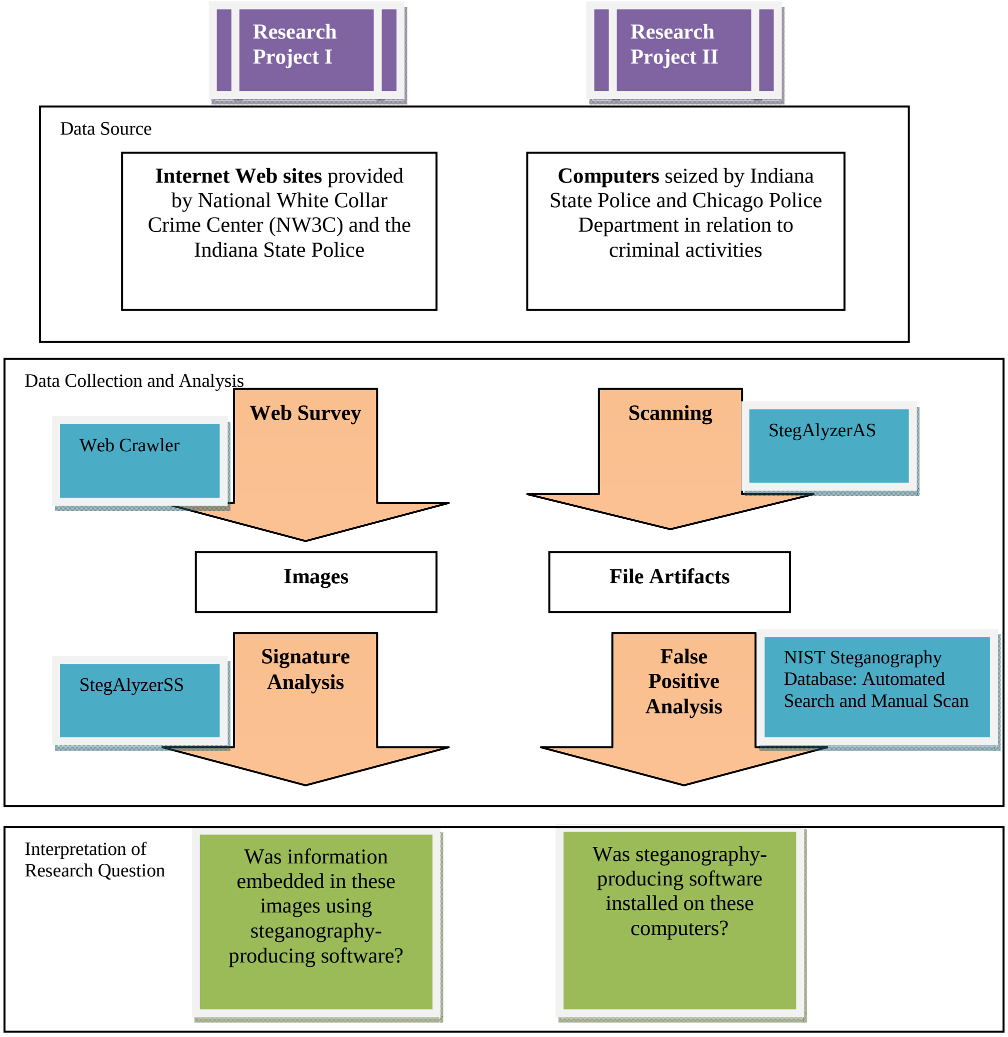
Research Project I
Research Project II
Data Source
Internet Web sites provided by National White Collar Crime Center (NW3C) and the Indiana State Police
Computers seized by Indiana State Police and Chicago Police Department in relation to criminal activities
Data Collection and Analysis
Web Crawler
Web Survey
Scanning
StegAlyzerAS
Images
File Artifacts
StegAlyzerSS
Signature Analysis
False Positive Analysis
NIST Steganography Database: Automated Search and Manual Scan
Interpretation of Research Question
Was information embedded in these images using steganography-producing software?
Was steganography-producing software installed on these computers?

Figure 2: Overview of the Research

2.1. Research Project I: Signature Detection in Carrier Images

This study attempted to determine to what extent steganography may have been used by criminals to communicate information over the Web. This project made the use of the signature (image artifact) detection technique explained in Section 1.

2.1.1. Motivation

Digital steganography represents a particularly significant threat today because of the large number of applications freely available on the Internet that are easy to find, download, install, and use to steal sensitive, classified, or proprietary information and conceal evidence of other types of criminal activity (Backbone Security). On the Web, there is evidence of terrorists using steganography to communicate covertly. "…individuals can use steganography…to embed messages into digital photographs or music clips. Posted on publicly available Web sites, the photos or clips are downloaded by collaborators as necessary. (This technique was reportedly used by recently arrested terrorists when they planned to blow up the US Embassy in Paris)" (Homer-Dixon, 2002, p. 54; Kolata, 2001). Terrorists would want to hide information in images which are posted to high traffic sites which have high turnover rates of images (Davidson and Goutam, 2004). Thus, steganography has the potential for harmful and dangerous applications.

2.1.2. Steganographic Web Survey

An attempt to determine the prevalence of steganography in carrier images on the Web was made by the research team in conjunction with Backbone Security, Inc. and law enforcement.

2.1.3. Comparison with related Prior Work: Survey conducted by University of Michigan

The research project described in this paper is distinct from the Web survey conducted by Neils Provos of University of Michigan in 2001. The latter used the tool Stegdetect to analyze over two million images from downloaded only from eBay auction sites. It was a survey of JPEG images only. The University of Michigan concluded that there was no steganography on the Web (Provos and Honeyman, 2002). Provos' survey used signature-based detection tools to detect the possible embedding of steganography only by three possible tools while as this research addresses as many as 16 information hiding tools.

2.1.4. Tools and Validation

The signature-based detection software selected for the survey proposed in this paper was StegAlyzerSS, version 1.1, named StegScan 1.1. The version of the software used to test the suspect images was ported to ANSI C from the proprietary version, and run under Debian LINUX. The version 2.0 of this software was rated effective by the Defense Cyber Crime Institute (DCCI) in tests conducted during October 2006. The tests conducted by the DCCI indicated the Backbone software was able to identify steganography from fourteen different algorithms with 99.6% certainty that steganography existed when an image was detected as containing steganography (Hirsh and Kong, 2006). The Backbone StegAlyzerSS was able to identify steganography embedded by the applications listed in table 1 (CyberScience Laboratory

Functional Analysis of StegAlyzerSS Version 1.1, 2005).

Table 1: List of Detectable Signatures

Steganography Embedding Program			
CryptArkan 1.0	InPlain View 1.0	wbStego 4.2	wbStego 2.0
JPegX 1.00.6	Camouflage 1.0.4	Camouflage 1.1.2	Camouflage 1.2.1
Cloak 7.0	Data Stash 1.1	Data Stash 1.1b	Data Stealth 1.0
Hiderman	Safe and Quick 2002	Steganography 1.50	Steganography 1.61

2.1.5. Added Software Functionality

While the StegalyzerSS product was used as the processing engine to analyze the Web-based images, that engine needed to be embedded within a larger system capable of crawling the Web in search of those images. The following additional functionality was developed:

- Added capability to run on multiple processing nodes that then aggregated their findings onto a single master node.
- Utilized MD5 hashing to prevent the downloading and analysis of duplicate pages (i.e. the same image file on multiple pages)
- Utilized concept of worksets where a workset is a single URL to be used as a starting point for crawling. All crawling was limited to the Internet domain that the initial URL belongs to.
- Recorded all downloaded files for option of re-analysis later with additional signatures.

2.1.6. Equipment

Web crawling technology was used to find and download images by the research team. The survey was conducted with a variable number of machines as seen in the table 2. One database server was always on line. This database server was the supervisor, and passed each next URL to the crawling nodes (machines which performed the HTTP requests).

Table 2: Crawling Node Force Deployment

Date	Crawling Nodes
3/31/2007	2
4/24/2007	3
6/19/2007	5

The database and file server computer was a Dell Optiplex GX280 Pentium 4 at 2.8GHz with 1 GB of RAM. The crawling nodes were Pentium IIIs running at 667 MHz with 512MB RAM. The connection to the Internet was made through a fiber optic connection. The overall system layout of the survey is illustrated in figure 3.

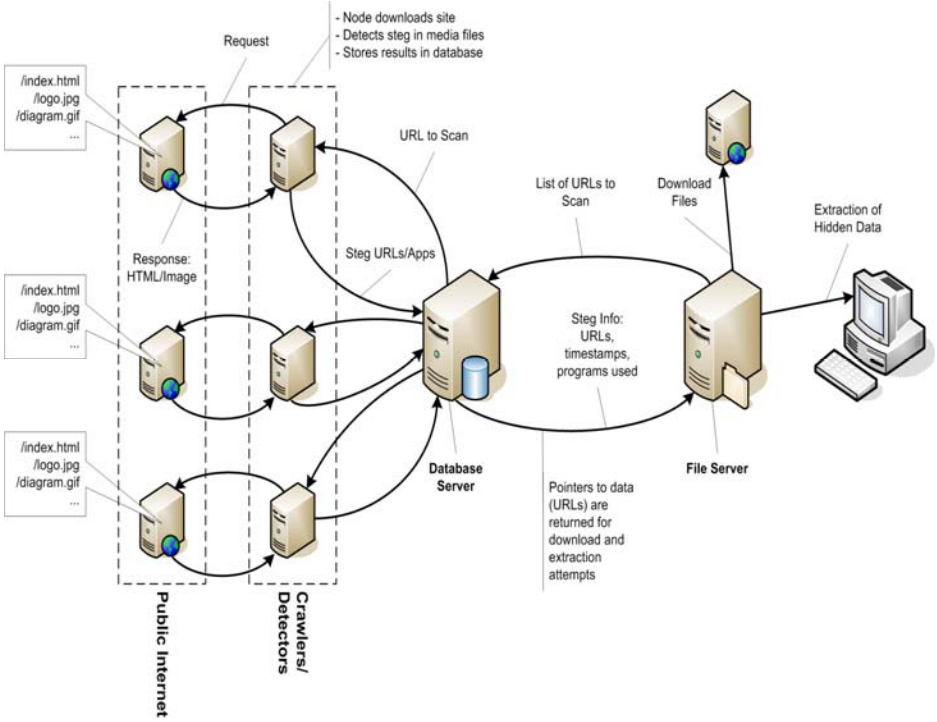

Figure 3: Web Crawler System Diagram

2.1.7. Data Gathering and Analysis

The survey was conducted by crawling selected base URLs recursively until there were no more links in the domain of the base URL to visit. Sites that had URLs different than specified base URLs were not visited. The base URLs of the sites that were surveyed were provided by parties interested in the results of the research as it pertained to their areas of responsibility. These organizations included the National White Collar Crime Center (NW3C) and the Indiana State Police. Heavily trafficked foreign sites figured prominently in the list of sites crawled for steganography. The total also included sites which had less than ten URLs searched. As seen in table 3, the number in parentheses indicates the number of base URLs in the grouping. The total is the total for all of the sites crawled with that grouping of completed URLs.

Table 3: Number of Completed URLs

Base URL Groupings	Completed URLs
A (4) - 10-100 URLs	205
B (6) - 101-1,000 URLs	2,912
C (2) - 1,001-10,000URLs	5,599
D (4) - 10,001 - 100,000 URLs	189,014
E (3) - 100,001 +	1,221,380
Total (sites with < 10 completed URLs)	1,419,114

The crawler application started at the base URL and crawled through all the links from each page to the maximum possible depth from the base URL. All of the items which were contained in a HREF= or IMG SRC= tag in the page source were downloaded, cataloged and then analyzed. The base URL was never traveled away from in the search, i.e., when a link specified a site with a different base URL, that link was not followed. The first recorded result was time-stamped 03/31/2006 at 1552hrs (local, Eastern time) and the final result was recorded on 06/30/2006 at 0444hrs.The image file types discovered in the survey are summarized in table 4.

Table 4: Image File Types Discovered in Survey File

Extension	Number Found
BMP	9
GIF	8,544
JPG	67,414
PNG	64
TIFF	7
X-3DS	7

These images were gathered and analyzed with the help of StegAlyzerSS application. No carrier images were found to contain steganographic signatures. Thus, the project failed to answer the research questions presented in figure 1 and figure 2.

2.1.8. Intermediate Conclusion

The Web crawling survey and signature detection did not find any conclusive evidence that steganography is being used on the Web. There are several possible explanations for this:

- There is no steganography in images on the Web
- The sample size was too small
- There was steganography, however it was not hidden in the file formats that the detection software was able to detect against, or
- There was steganography in the surveyed images; however it was not hidden using the algorithms the detection software was aware of.

2.1.9. Research Redirection

The detection problem in steganography is multivariate. There are over 825 embedding applications which have been available for download from the Internet at various times (J. Goldman, personal communication, May 16, 2007). This number is to be compared to the number of signatures which image analysis software is currently capable of detecting, which is in the neighborhood of 25-35. Not all of these signatures can be detected by a single steganography detection application.

The most likely explanation for the steganography not being detected in the Research Project I is that the steganography used and posted on the public Internet uses hiding techniques and hiding applications which were not being tested by the StegAlyzerSS detection application.

There are many steganography-producing applications that could possibly be used and it is impractical to write signature detection algorithms for all of them. For this reason, some prioritization of effort in developing signature based detection methods must be achieved. To do so there must be a sense of which applications the users of steganography are employing. To study this concept, the following "Stego-Usage Timeline" (table 5) was formed.

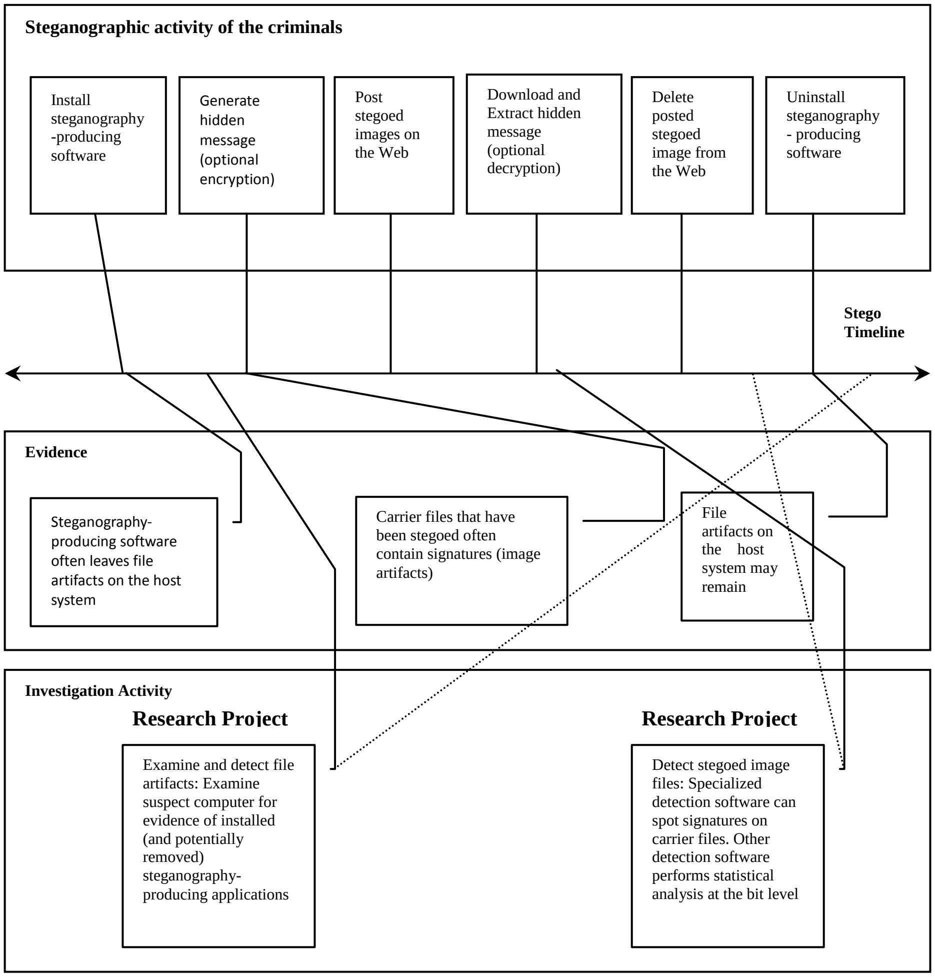
Steganographic activity of the criminals
Install steganography -producing software
Generate hidden message (optional encryption)
Post stegoed images on the Web
Download and Extract hidden message (optional decryption)
Delete posted stegoed image from the Web
Uninstall steganography - producing software
Stego Timeline
Evidence
Steganography-producing software often leaves file artifacts on the host system
Carrier files that have been stegoed often contain signatures (image artifacts)
File artifacts on the host system may remain
Investigation Activity
Research Project
Examine and detect file artifacts: Examine suspect computer for evidence of installed (and potentially removed) steganography-producing applications
Research Project
Detect stegoed image files: Specialized detection software can spot signatures on carrier files. Other detection software performs statistical analysis at the bit level

Figure 4: The Stego-Usage Timeline

In order to produce a hidden message, steganographic software must be acquired and installed. This creates file artifacts on the host system. The file artifacts can take the form of changed registry entries, temporary files, directories created, and shortcuts on the desktop. Stegoed files are then generated. This involves selecting cover data, selecting the message and then embedding the message. These files would presumably be one time use files, and would be deleted at the conclusion of sending the files. The deletion of the data would, if not done securely, also leave evidence on the host machine. Specifically, standard forensic analysis would reveal these files. If the user did not delete the files in any way at the conclusion of the use of the steganography application, these files would provide images for comparison of cover images and provide an easier attack to extract the steganography. At the time the stegoed file is transmitted or posted, it could be observed over the communications channel. It would also be theoretically possible to run detection techniques against all of the data observed during transmission.

Once the suspect files have been detected, further analysis is performed. Knowing the specific steganography- producing application which hid the information would allow for possible extraction. If the message is extracted, then there may be a decryption step if the information was encrypted before it was embedded and sent. The user of the steganography software would then potentially uninstall the software perhaps thinking naively that no trace of his or her activity existed. In fact, because the vast majority of these applications are written with less than professional production values, the artifacts of the existence of that software would still be present on the host system and would be detectable if proper detection methods were used. Artifact detection would be the final step. This step requires the forensic examiner to have access to the host machine and is able to analyze it with the proper software.

Thus, a research redirection from Project I to Project II was introduced in order to address the research question as shown in figure 5. The research was carried out in two subsequent phases where Project II followed Project I and in which the approach taken for addressing the research question was modified based on the conclusions presented in 2.1.8 and the fact that the Project I failed to establish that steganography was used by criminals. Thus, Research Project I provides context for Research Project II. The redirection was based on the concept of ‘Stego-Usage Timeline’ presented in figure 4. As explained above, steganography-generation process involves many phases and while the time when the signature-based method (as demonstrated in Research Project I) can be used for detection is later in the timeline, the host-file detection method (demonstrated in Research Project II) allows us to traverse back to the earlier stages of steganography-generation and its detection is applicable over a larger span (dotted line in figure 4 marks the end of this span) of the Stego-Usage timeline as compared to that of the former.

Thus, the redirection consists of a focus shift from the analysis of the ‘product’ of the steganography-producing software; viz. the images, to the ‘evidence’ left by the steganography-producing software while they are being used to generate steganographic images.

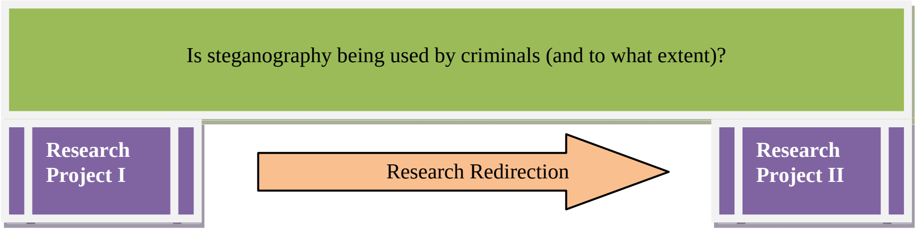
Is steganography being used by criminals (and to what extent)?
Research Project I
Research Redirection
Research Project II

Figure 5: Research Redirection

2.2. Research Project II: File Artifact Detection on Criminal Computers

The second phase of the research consisted of scanning computers seized by law enforcement agencies for steganographic file artifacts.

2.2.1. Motivation

A tool which can detect steganographic file artifacts can be used on suspects' machines to determine if certain steganography-producing software was ever present or used and what it may have been. With positive results regarding the use of steganography software on machines seized by law enforcement or the national security apparatus, a sense of which steganography-producing software is popular with which criminal elements can be gained, thereby allowing research in signature detection to be focused in that direction.

It would be useful to require that all imaged hard drives booked into evidence and examined should be examined for host system artifacts with a file artifact detection application. It is critical for efficient use of law enforcement and national security resources to know which applications are being used by criminal and terrorist suspects. With the knowledge of which steganography-producing software is in use, based on evidence provided from seized computers which exhibit artifacts of that particular steganography software, there will be a more structured focus possible in terms of writing signature based detection tools. These tools would detect against the signatures of steganography-producing applications found to be favored by the groups which are of interest. Thus, a research focus shift to host system artifact detection was indicated by the Stego-Usage Timeline.

2.2.2. Tools and Validation

The product used for performing detection of the steganographic file artifacts was Backbone Security's StegAlyzerAS version 2.1. This software tool enables the investigators to detect the presence of steganography-producing programs as well as the remnants of such programs within a computer's file and registry. The program enables the user to search files, directories, entire drives, and forensic images to locate evidence of data hiding activity on a disk. The program is capable of mounting forensic images in the following formats: EnCase RAW (dd), ISO, and SMART. StegAlyzerAS includes case management features to log all actions performed during an investigation, record detailed information about each case, and manage collected evidence. The application generates reports in HTML format (CyberScience Laboratory, 2008). Version 2.1 of StegAlyzerAS can detect all the file and Windows Registry artifacts associated with 625 digital steganography and information hiding tools (Backbone Security). StegAlyzerAS allows the search of the files by using hash values such as MD5 and SHA-256 which are stored in the Steganography Application Fingerprint Database (SAFDB) and registry entries stored in the Registry Artifact Key Database (RAKDB) distributed with

StegAlyzerAS (Backbone Security).

This host artifact detection tool was tested for performance by conducting a study of a test machine on which a variety of steganographic software was installed and deleted or uninstalled. StegAlyzerAS was able to detect that all of those steganography-producing programs were once installed on the test machine. Also, StegAlyzerAS was found to be effective for identifying file and registry artifacts by the Defense Cyber Crime Institute (DCCI) and the CyberScience Laboratory (CSL) (Backbone Security).

2.2.3. Data Collection Process

The product along with its usage instructions was sent to Chicago Police Department (CPD) which scanned the criminal computer disks with the help of StegAlyzerAS. A member of the research team was allowed to carry out the scans of the criminal computers for a day at the CPD. The research team also scanned images provided by the Indiana State Police. The tool was run against the disk images of total 96 computer drives from seized computers and each one was known as a case. This formed the sample set which corresponded to the population of users of steganography with criminal background (n=96).

2.2.4. Data Processing and Analysis

The information from the generated HTML files was extracted into a MS Access database (2002- 2003 format) which consisted the following for each case:

- Description of the case: This consisted of details regarding the seized computer, crime associated with it, agency handling the case and criminal details
- Results obtained after scanning the respective drive: This consisted of information about the artifacts found in the report of each case, the corresponding steganography-producing application which generated it, the location of the artifact and its MD5 hash value.

2.2.5. False Positive Analysis

A large number of artifacts were flagged as likely steganography-producing software artifacts. But these artifacts may or may not have been the indicators that steganographic tools were installed on that machine because certain artifacts that are left behind by the steganography-producing applications can also be left by common and harmless applications; e.g. unwise.exe is a common artifact found bundled with the Wise installer package. Although, StegalyzerAS flags it to be associated with steganography-producing software, a particular instance of the artifact may be left behind by software like MS Word since unwise.exe is used by non-steganographic software too. Therefore, the entire artifact list obtained in the scan results had to be subjected to a false positive analysis.

In order to ensure that a particular artifact was specific to a steganographic application the following false positive treatment was devised. The NIST National Software Reference Library database (NSRL) consists of MD5 hash values of file artifacts left by known, traceable software applications and it also lists the type of application it is. The NIST database was broken down into 3 different classes as follows:

Table 5: NIST Database Artifact Classification

Class 1	Class 2	Class 3
Consisted of MD5 hash values of artifacts which belonged only to the non-steganographic applications	Consisted of MD5 hash values of artifacts which belonged only to the steganographic applications	Consisted of MD5 hash values of artifacts which were common to both the steganographic as well as the non-steganographic applications.

The false positive analysis was carried out by assigning a value to each (artifact, corresponding steganography-producing application) pair based on the comparison of its hash value with those in the NSRL database and depending on the class under which it fell. The values were assigned from the following set:

Table 6: Results Artifact Classification

NS	PS	ID	NF	MS
Not Steganography, if the artifact was found in the Class 1	Positive Steganography, if it was found only in Class 2	Indeterminate, if it was found in Class 3	Not Found, if it was not found in either of the classes of table 5, so it is not known whether the artifact is steganographic or not.	Manual Scan: The pairs flagged with the value NF were checked manually for their location where the artifacts were found and depending on intelligent human judgment, they were flagged as MS if the path or pathnames showed any indication of the artifact being in steganography related folders.

2.2.6. Results and Analysis

The artifacts and applications were then subjected to analysis as per the value taken by the false positive flag. The obtained results and their implications are enlisted as below:

- The term 'interesting items' refers to the entries flagged with a 'PS' or a 'NF' value because it indicates that either the (application, artifact) pair is confirmed as steganography or it cannot be eliminated as a non-steganographic instance respectively.
- The total of the 96 cases had 4708 (application, artifact) pairs and not all of them were unique. In terms of artifacts, only 62 unique artifact names were found where 11 of them appeared only once. The remaining 51 artifact names accounted for 928 of the 939 total interesting items (98.8%). Thus, a large number of duplicates was found in the (application, artifact) pairs.
- Manual scan results depended on the intelligent judgment of the human who was conducting it. As seen in table 7, in spite of going through the paths of each and every (application, artifact) pair, it was not possible to conclude in any case that an artifact was definitively steganographic in nature from its path name and the location of the folder on the seized criminal computer. Thus, the manual scan failed to produce any positive results for this dataset (MS=0).

The number of occurrences for each of the flag values in the order of its decreasing importance is specified in table 7.

Table 7: Analysis of flag occurrence counts

Serial Number	Flag	Occurrence (Total 4708)	Percentage of Occurrence (rounded) (%)
1	PS (Interesting Items)	12	0.2549
2	MS	0	0.0000
3	NF(Interesting Items)	927	19.6899
4	ID	1785	37.9142
5	NS	1984	42.1410

- It can be observed that there is 42.1410 % of confirmed non-steganographic data which means that in the best possible case there can be 57.8590% of steganography (if all the NF and ID flags were indeed positive in terms of steganography) out of which this research was able to prove and confirm 0.2549%. The other tools that StegAlyzerSS was capable of detecting were either not popular among the criminals or their corresponding artifacts were categorized as NF or ID. And thus the possibility of the usage of those applications by criminals cannot be eliminated.

- Twelve Positive Steganography flags were found. They were distributed over four cases respectively. Table 8 provides the details regarding these findings. It can be seen that the results generated by StegAlyzerAS and the false positive treatment confirmed the use of four different applications for generating steganography; viz. Gif-It-Up, OutGuess, MP3StegoEncoder and wbStego. If these applications are compared with those mentioned in table 1 it is observed that the signature detection tool StegAlyzerSS was only capable of scanning the criminal computers for 16 programs and out of which only one; namely wbStego was used by the criminals and also the version used was different. Thus table 8 demonstrates that steganography had been used by criminals of varied backgrounds and that certain steganographic tools are likely to be popular among criminals that the signature detection tools were not capable of recognizing.

Table 8: Positive Steganography Flag details

Case Number	Number of PS Flags	Crime Type	Application Name	Artifact Name
1	8	Attempted Homicide	wbStego99 v3.51	_ISREG32.DLL
			wbStego99 v3.5	_ISREG32.DLL
			wbStego99 v3.1	_ISREG32.DLL
			wbStego v4.2	_ISREG32.DLL
			wbStego v4.0	_ISREG32.DLL *
			MP3StegoEncoder v1.1.15 (Linux)	Huffdec
			MP3StegoEncoder v1.1.15 (Linux)	Dewindow
			wbStego v4.1	_ISREG32.DLL
2	2	Fraud on a Financial Institution	MP3StegoEncoder v1.1.15 (Linux)	Dewindow
			MP3StegoEncoder v1.1.15 (Linux)	Huffdec
3	1	Unknown	OutGuess v0.2 (Linux)	install-sh
4	1	Child Porn	Gif-It-Up v1.0	_ISREG32.DLL *

*Although the commonality of the artifact is observed over different steganography-producing applications, the aim of the flag classification is indicating the presence of positive steganography and not establishing a unique correlation between an artifact and an application

- These results can also contribute in formulating efficient signature detection approach by providing input in the form of a list of popular steganography-producing programs. This is done by introducing prioritization in the process of signature detection as shown in figure 6. As explained in section 2.1.9, the detection software faces limitations in terms of the number of signatures it is capable of detecting. This is because the number of existing steganography-producing applications is too large to be incorporated into the detection engine and it keeps growing over time. For instance, from the feedback of the analysis provided by table 8, the detection tool should incorporate the capability for detecting the steganography-producing tools found in file artifact scanning; namely Gif-It-Up, OutGuess, MP3StegoEncoder and the specific versions of wbStego mentioned in the table 8, i.e. it should be aware of the algorithms used by these applications. The applications which are associated with NF and ID flags and which occur over and over again can also be incorporated in the signature detection tool. Since a large number of (artifact, application) duplicate pairs were found in this case, it becomes easier to choose the applications for prioritization.

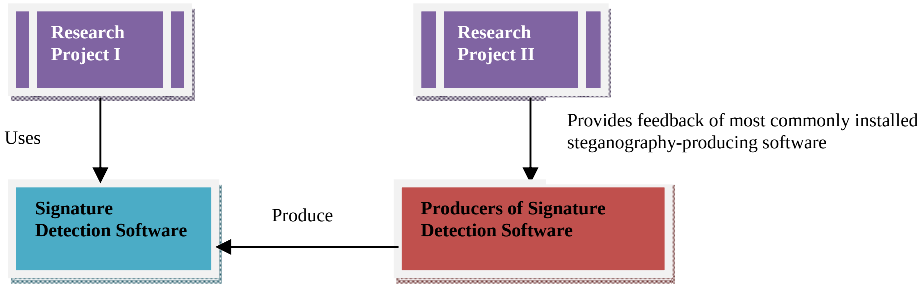
Research Project I
Research Project II
Uses
Provides feedback of most commonly installed steganography-producing software
Signature Detection Software
Produce
Producers of Signature Detection Software

Figure 6: The Concept of Prioritized Feedback

2.3. Conclusion

The results of the Research Project II demonstrate that evidence pertaining to the usage of steganography by criminals was found and thus a proof of concept of the effectiveness of the new, enhanced procedure proposed for finding steganography was presented. The procedure proposed in Research Project I failed to provide evidence in favor of the argument that ‘steganography is being used by criminals’. Thus, the new methodology based on Stego-Usage Timeline is an improvement over the one proposed in Research Project I and has been able to answer the research questions presented in figure 1 and figure 2 positively. The research also demonstrates the concept of prioritization of the steganography-producing software by providing feedback based on the scanning results for the purpose of designing efficient signature detection algorithms. The following figure summarizes the contribution of the entire research effort.

Key Outcomes of Research Project I

- Over 1 million Web sites crawled for images
- Images scanned for steganographic signatures and no steganography found
- Concepts of Stego-Usage Timeline and Prioritization proposed
- Research methodology redefined based on Stego-Usage Timeline with a focus on file

Key Outcomes of Research Project II

- Over 96 crime-related seized drives scanned for steganographic artifacts
- False positive analysis carried out with help of the NSRL database
- Proof of Concept demonstrated for the host file artifact-based detection methodology
- Concept of Prioritization demonstrated

Figure 7: Key Outcomes of Research Projects

2.4. Further Research

Although the research was able to present a proof of concept of the proposed Stego-Usage Timeline methodology, the evidence observed was limited and is not statistically significant to analyze the trends the observed in steganography-producing applications and criminal backgrounds associated with it. This can be due to several possible reasons:

- The sample size (number of cases scanned=96) was too small
- Criminals used steganography; however the file artifacts could not be traced down as being steganographic in nature when they were common to many applications (i.e. they fell under NF and ID categories)
- Criminals used steganography; however the file artifact detection software was unaware of the applications used by the criminals to generate steganography and hence was unable to associate the corresponding artifacts to steganography

Thus, further research is being focused towards obtaining more positives by increasing the sample set in order to overcome the first limitation mentioned above. The proof of concept has demonstrated that steganography is being used but further research will help in answering the question ‘to what extent steganography is being used’.

Also, a model of a multi-phased approach towards steganography detection is being formulated where the usage of both techniques; viz. artifact detection and signature detection is suggested for the detection of steganography. A more prioritized and focused direction which deals with the artifact detection first and then proceeds to the signature detection with algorithms incorporating the prioritization feedback is proposed. This approach has the potential to be an efficient mechanism in terms of time and effort by overcoming the limitations faced by either of the research projects when carried out independently.

REFERENCES

Acharya, T. and Tsai, P. (2005), JPEG2000 standard for image compression: Concepts, Algorithms and VLSI Architectures, John Wiley & Sons, Inc., Hoboken, N.J.

Backbone Security (2008), ‘SARC Releases Enhanced Digital Steganography Detection Tool’, http://www.sarc-wv.com/news/stegalyzeras21.aspx, October 7, 2008.

Backbone Security (2008), ‘StegAlyzerAS’, http://www.sarc-wv.com/docs/stegalyzeras.pdf, October 7, 2008.

CyberScience Laboratory, CyberScience Laboratory Functional Analysis of StegAlyzerSS Version 1.1. (2005). CyberScience Laboratory, Rome, New York.

CyberScience Laboratory, CyberScience Laboratory Functional Analysis of StegAlyzerAS Version 3.0. (2008). CyberScience Laboratory, Rome, New York.

Davidson, I. and Goutam, P. (2004), ‘Locating secret messages in images’. International Conference on Knowledge Discovery and Data Mining. 2004. Seattle, WA, USA.

Goudy, S. (2004), ‘Embedding the evil within’. The Corrections Connection Network News. Jan 21, 2004. http://www.corrections.com/news/article?articleid=14974. July 31, 2007.

Homer-Dixon, T. (2002), The Rise of Complex Terrorism - Foreign Policy.

Hirsh, M. and Kong, E. (2006), Test report for StegAlyzerSS v2.0. Defense Cyber Crime Institute.

Jackson, J. T., Gunsch, G. H., Claypoole, R. L., Jr.and Lamont, G. B. (2003), “Blind Steganography Detection Using a Computational Immune System: A Work in Progress”. International Journal of Digital Evidence, 4(1), 19.

Katzenbeisser, S. and Petitcolas, F. A. P. (2000), Information hiding techniques for steganography and digital watermarking, Artech House, Boston.

Kolata, G. (2001), ‘Veiled Messages of Terror May Lurk in Cyberspace’, The New York Times, October, 30, 2001.

NSRL, http://www.nsrl.nist.gov/, October, 7, 2008.

Provos, N. and Honeyman, P. (2002), 'Detecting Steganographic Content on the Internet', www.citi.umich.edu/techreports/reports/citi-tr-01-11.pdf, July 31, 2007.

Singh, S. (1999), The code book, Anchor Books, New York.

StegAlyzerSS (2006), Backbone Security, Inc.

Stego Suite (2006), Wetstone Technologies, Inc.

Wayner, P. (2002), Disappearing cryptography: information hiding: Steganography & watermarking (2nd ed.), MK/Morgan Kaufmann Publishers, Amsterdam, Boston.