Assured Information Security: Stego Intrusion Detection System

By

Mike Sieffert, Rodney Forbes, Charles Green, Leonard Popyack, Thomas Blake

From the proceedings of

The Digital Forensic Research Conference

DFRWS 2004 USA

Baltimore, MD (Aug 11th - 13th)

Stego Intrusion Detection System

Michael Sieffert[1], Rodney Forbes[1], Charles Green[1], Leonard Popyack[1], Thomas Blake[2]

[1]Assured Information Security, Inc. PO Box 1182, Rome NY 13442, USA
{sieffert, forbesr, greenc, popyack}@ainfosec.com
[2]Air Force Research Laboratory, 525 Brooks Road, Rome NY 13442, USA
Thomas.Blake@rl.af.mil

Abstract. This paper will describe the design, development, and testing of a prototype computer network Steganography Intrusion Detection System (SIDS) architecture. The Air Force Research Laboratory (AFRL) recognized the need to have the ability to detect the presence of hidden data in perception-based objects (e.g., images) passed into and out of computer networks. With widespread use of commercial and freeware steganography tools, it is very likely that these techniques will be used against our networks eventually and we must be prepared for such an attack. The SIDS effort at AFRL developed the first intrusion detection system of its kind that demonstrates how hidden data can be discovered as it enters or leaves an enterprise network. While further research is required in steganalysis techniques, this paper will describe the framework that is a step towards steganography detection in web traffic. This paper will describe our method to reconstruct image data from HTTP web traffic, the plug-in interface for steganalysis algorithms, and the graphical user interface as well as provide test data from realistic network testing. Future work and needs will be presented.

1 Introduction

Steganography, the art of hiding the existence of communication, has been used for thousands of years. It is a method of concealing data in channels that are considered commonplace; specifically those are not typically scrutinized. The hope is that the data passes unnoticed to the receiver through a medium whose main purpose is generally benign in nature. Such communication is in many cases able to render useless possible attempts to prevent such data transfer.

In this paper, we will use the term steganography to refer to hiding information in media files. This "hiding" is done in a manner that does not alter a human's perception of the picture or sound. In SIDS, we concentrate on steganography with the use of image files. Image files present an attractive cover object for hiding information because they are usually observed only in their graphical form by the human eye. Except in very rare circumstances, human eyes are incapable of determining whether or not slight changes have been made to an image's color/hue/brightness to store hidden information (See Figure 1). Image files are rarely held up to careful inspection, can easily be found on the majority of computers, and are transferred constantly over HTTP from web pages. For these reasons, images are very likely to be used as stego cover objects in covert communication.

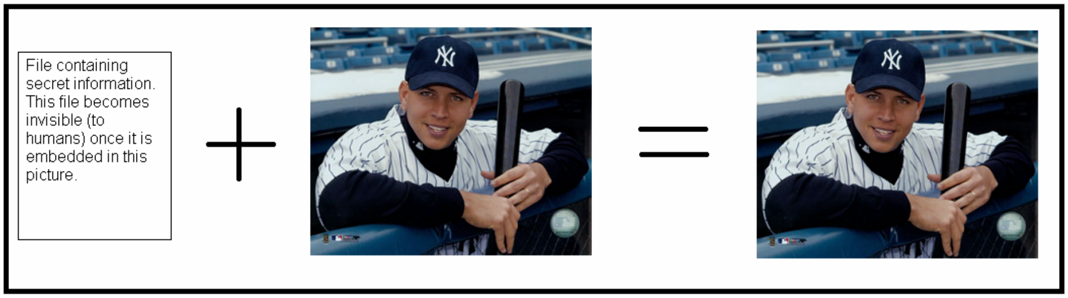

File containing secret information. This file becomes invisible (to humans) once it is embedded in this picture.
+
=

Figure 1. A file containing stego is indistinguishable from the original, from a human's perspective.
A computer, however, can tell the difference.

Discussion of steganography necessarily leads to discussion of steganalysis. Steganalysis is the art of discovering the presence of stego content in media, and is crucial in any attempt to thwart stego communication. It is an inexact science, and steganalysis algorithms generally output a number within a given range to indicate the likelihood that stego was actually used on the input file. Steganalysis is a rapidly advancing science, and will continue to develop as long as steganographic algorithms are being created and used.

As stated above, steganographic communication is a great candidate for covertly transmitting data through the Internet, possibly to places where there are mechanisms aiming to prevent all types of overt communication to or from certain parties. In this paper we focus on the transfer of sensitive data into or out of a network that is protected by controls such as firewalls, network intrusion detection systems (NIDS), host intrusion detection systems (HIDS), anti-virus applications, etc.

Programs capable of hiding steganographic content in common file types are freely and readily available on the Internet today. Any one of these tools can be used to smuggle data into or out of a network by utilizing any unstructured carrier such as digital images or sounds. Consider the following scenario: An employee uses a stego tool to embed proprietary information into a JPEG image file. The image is then placed on the company website. Outside of work, he is now able to anonymously visit the web page, retrieve the image, and extract the proprietary information. Another example of stego use is the following: An advanced Trojan horse is installed on a company's network. This malicious program is designed to retrieve instructions by extracting data out of images downloaded from certain web pages controlled by the author. These images are obtained through what appear to be normal web traffic. A malicious program with this capability, once installed, could be controlled by an author outside of even a well-protected network's perimeter defenses. An investigation into both the freely and commercially available network defenses today reveals that there is no tool available to detect or deter this type of attack.

SIDS, the Steganographic Intrusion Detection System, is able to detect advanced attacks such as those described above. Administrators will be alerted, and prompt action can be taken to identify any individual or entity taking part in this type of nefarious communication.

2 Motivation

With our realization of the ease with which images on web pages can be used for communication of information whose intent is questionable or illegal, we designed and developed a system that can unobtrusively monitor for this type of activity on any network and produce alerts when conditions warrant. SIDS is an application that monitors all web traffic going into an out of a given network and checks for possible steganographic content in every single image that is transferred unencrypted. SIDS could generally be considered another perimeter defense system, which concentrates its efforts at a network's gateway to the Internet and observes all traffic passing into and out of the protected network.

3 Implementation

SIDS attempts to thwart any attempts at using steganographic algorithms to communicate through channels such as HTTP traffic. To do so, it must be placed at the border of a network that is to be protected (See Figure 2). Preferably, SIDS should be placed just inside of the firewall, so that only web traffic taking place over a connection with a computer on the inside of the network is monitored and the system is not overwhelmed with data destined for other networks.

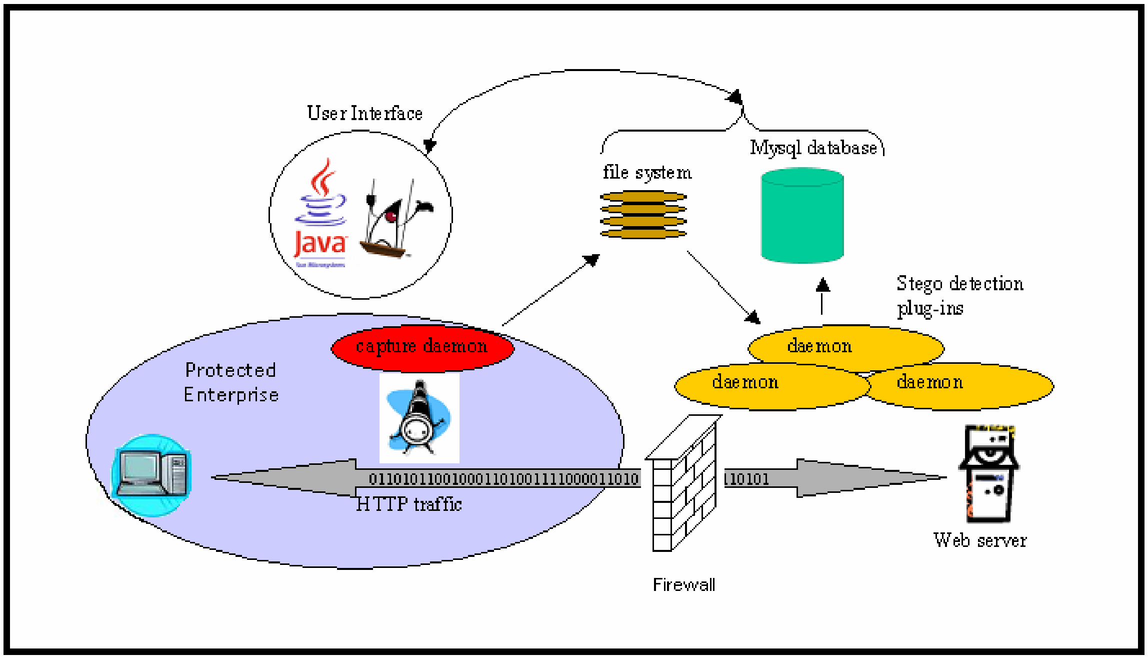


Figure 2. Graphical representation of the SIDS architecture.

SIDS must make use of steganalysis algorithms to determine if content captured from the wire contains stego. A plug-in capability is included in the application that allows for updating and adding to the steganalysis algorithms in place. A published API is specified that allows anyone to add his or her own steganalysis algorithm and place it in the plug-in directory to be used in SIDS operation. This allows a company that does not wish to release its stego detection algorithms to utilize them in SIDS without making them public, thereby protecting the intellectual property of the detection algorithm.

In addition to a plug-in architecture for steganalysis algorithms, SIDS allows the user to customize alert thresholds for each algorithm present in the system. Therefore, over time an administrator can tune the thresholds as he or she sees fit in order to maximize system effectiveness and minimize false positive and false negative errors.

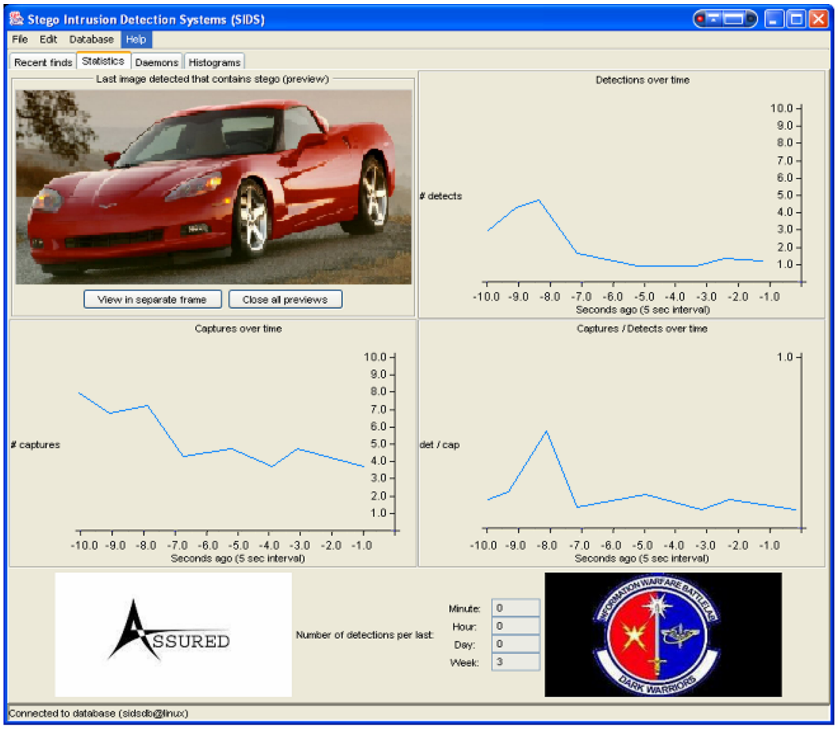
SIDS was developed for the Linux platform and written in C++. It makes use of a MySQL database to store statistics regarding the data captured as well as steganalysis results for each piece of data.

A network traffic daemon in SIDS places the network card into promiscuous mode and observes (sniffs) all traffic passing through the wire. Specifically, all HTTP traffic is monitored. Each image that is transferred over an HTTP connection is reconstructed and logged to the filesystem. Each time an image is logged, an entry is made to the database specifying information associated with the connection from which the image was captured. Both the source and destination IP addresses are logged along with source and destination port numbers, time of capture, and the location of the image on the filesystem after logging.

From the network's point of view, the SIDS machine poses no overhead on the network utilization or available bandwidth. The machine hosting SIDS does not even need an IP address. This allows networks with even the heaviest traffic load to remain unaffected by a SIDS installation, while adding another layer of protection around a network. SIDS acts exactly like an additional IDS system and possesses the same properties as perimeter IDS systems currently in use today.

Another daemon, the stego detection daemon, is constantly running in the background. This daemon runs each applicable steganalysis algorithm in the plug-in directory against every image captured off the line, and logs the results in the database. This will give SIDS the greatest chance of identifying data containing stego, and hence give the operator the best possible view of what is actually taking place on his/her network.

The remaining portion of SIDS is the graphical user interface (GUI), which is written in Java using Swing (See Figures 3 through 5 for screen shots). Tabs are used to separate the levels at which data can be viewed through SIDS. The GUI is designed to give an operator the opportunity to view network wide trends and statistics in easily understandable graphs, as well as very precise tables that contain entries for each piece of data in the database. The graphs are updated in real time, and make it easy to tell when there are high levels of web traffic being generated and how much of that traffic is testing positive for stego content. The tables are used to allow careful inspection of specific images in the database, and allow the user to drill down and view detailed information regarding the image's capture.

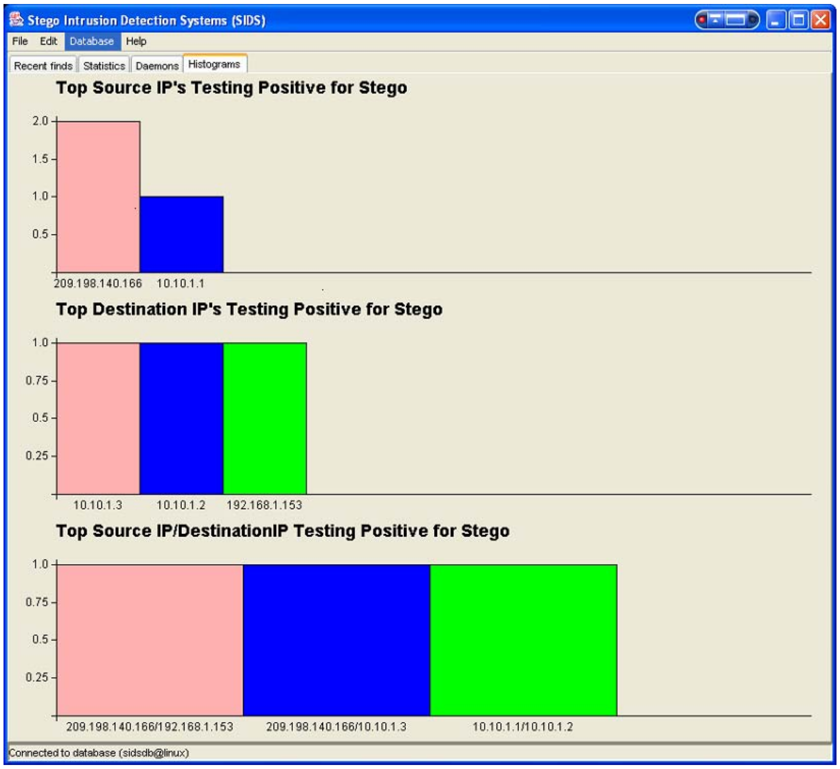


Number of detections per last:

Minute:	0
Hour:	0
Day:	0
Week:	3

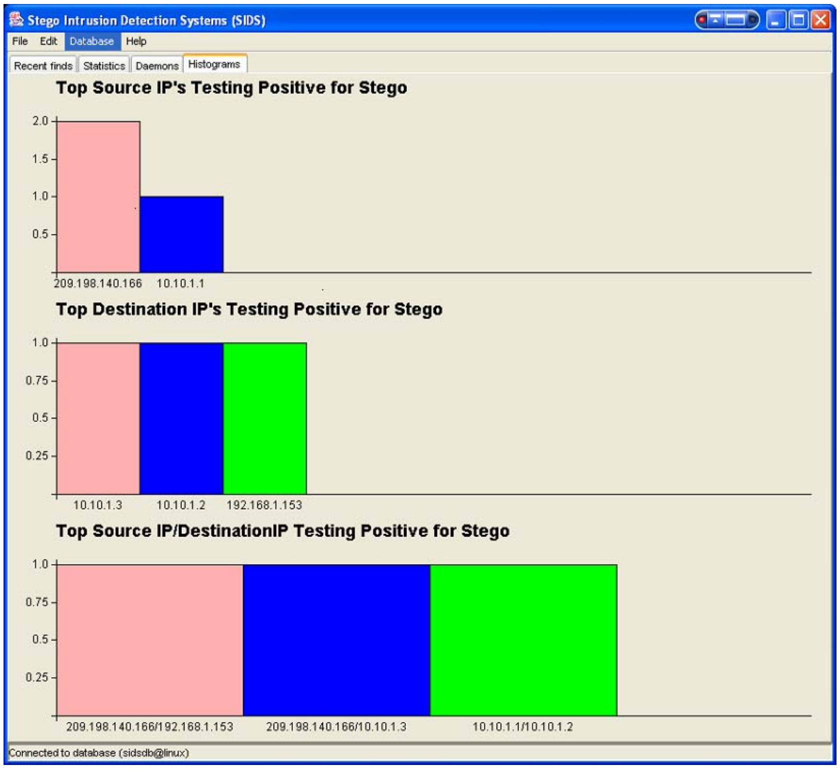
Connected to database (sidsdb@linux)

Figure 3. The 'Statistics' pane of SIDS gives the user a quick view of recent activity. A preview of the last image testing positive for stego is also shown.



Connected to database (sidsdb@linux)

Figure 4. The 'Histograms' tab indicates the top offenders' IP addresses.



Connected to database (sidsdb@linux)

Figure 5. The 'Recent Finds' tab lists the images most recently captured off the line, and those that have tested positive for stego.

4 Test Results

Early demonstrations of SIDS have shown that its effectiveness is based very strongly on the strength of the steganalysis algorithms being used. The two types of errors, false positives and false negatives, will occur in just about every steganalysis algorithm developed. A high false positive rate will likely lead to ineffectiveness as a network administrator is desensitized to alerts and fails to investigate those incidents that were true positives. On the other hand, a high false negative rate could undermine the program's purpose by allowing all but the most blatant attempts at stego communication to occur. A useful steganalysis algorithm will seek to minimize these types of errors to keep effectiveness high and deter potential attacks.

The plug-in support provided by SIDS is of great importance. Although we distribute a handful of steganalysis algorithms with the prototype application, our main objective in developing SIDS was to create a framework capable of utilizing any algorithm to detect stego communication. This means that keeping the algorithms used by SIDS up to date is crucial as new steganographic algorithms will be developed. The threat of stego will remain a moving one.

If the steganalysis algorithms used in SIDS are believed to be sufficiently accurate, it has been shown that the graphs readily communicate to the user exactly where on his or her network stego communication is taking place. It points out the top destination and source IP addresses of machines taking part in this type of activity, as well as the top source and destination IP pairs doing so.

5 Limitations

Many different file formats have steganographic algorithms available for their utilization as cover objects, and the number of them is steadily increasing. In addition, new file formats are constantly being produced which further open the door for more stego algorithms. This leads to steganalysis lagging consistently behind the state of the art in steganography. Also, because steganalysis algorithms almost always have a false positive rate that is non-negligible, the advantage currently falls to the attackers or those wishing to hide their communications in channels that are either not being closely examined or examined for the wrong thing entirely.

SIDS is designed to change with the state of the art of steganalysis technologies, and is able to be updated and have its functionality increased by allowing for pluggable steganalysis algorithms. This is very important in the dynamic environment of steganography and hidden data sources. New hiding methods are routinely created (as demonstrated by the papers at each of the past five International Information Hiding conferences, http://msrcmt.research.microsoft.com/IH2004.)

SIDS, as mentioned, is currently considered a border defense mechanism. As such, it is only able to view data on its network segment. Therefore, if a switched network is used (rather than a non-switched network), a port must be allocated to SIDS that allows for the viewing of all traffic on the network. Only traffic that passes through the network neither encrypted nor obfuscated can be captured and tested for stego content. A solution to this has been proposed for future releases.

The SIDS capture daemon is currently single threaded, and will not be able to keep up with heavy traffic on a network with multiple users. This limitation will be removed in future releases of SIDS by multi-threading the capture engine.

6 Future

Images are certainly not the only type of files that are able to be steganographic carriers. Any unstructured carrier can be structured to hide the existence of embedded data. Many files such as video files, music files, text files, and even executable files can be used to hide data. Only cosmetic changes, and new steganalysis algorithms would be required to add the capability of SIDS to monitor for these different file types. The more steganalysis algorithms developed and added to your defense system, the less likely it becomes that stego communication is taking place on your network without your knowledge.

SIDS is an effective defense against those pursuing covert communication through images on web pages. An attacker, however, will not be limited to that choice and can make use of other stego channels that SIDS does not monitor. This includes email traffic, file transfer protocol (FTP) traffic, peer-to-peer traffic, etc. Future versions of SIDS will check for different types of channels through which media files containing stego can be transferred.

As mentioned earlier, only data that is transferred unencrypted, or in the clear, is observable by SIDS and checked for stego content. Therefore, any images transferred over a HTTPS connection would pass by SIDS undetected. To aid in solving this problem, we will implement a host version of SIDS (HSIDS). HSIDS would sit on protected hosts that an administrator would like to monitor on his/her network. It will integrate itself into the host's web browser, email client, peer-to-peer applications, and any other application that at some point caches transferred data. Any data that is transferred through the network encrypted must be unencrypted at some point to be useful. As soon as it is unencrypted on the host and saved to disk, all available steganalysis algorithms will check it. Any potentially suspect data would be quarantined, and the administrator notified to its presence on the system. This version of SIDS would function much as anti-virus software does today, constantly checking incoming data saved to the hard drive and alerting the user to the presence of malicious data.

An extension to HSIDS will be developed that is tailored for use by an enterprise. In this case, HSIDS will run on multiple network hosts and be coordinated by a central SIDS server. The server would continually collect information from each of the HSIDS hosts on the network, presenting the data to an administrator in one easy to user interface. This will provide an easy way to recognize those machines on a network that seem to be involved in the transfer of data containing stego.

In all future version of SIDS, we will build the infrastructure that will allow for automatic updates of steganalysis algorithms on any machine running SIDS. Since updated steganalysis algorithms are of paramount importance to a system that wishes to recognize stego communications, it would be very useful to have a program that goes out to an Internet site supporting the download of new algorithms for SIDS.

7 Appendix

For information about updates to SIDS and future development contact Thomas Blake or Charles Green, both listed as authors at the beginning of this document.