A Case Study for Automatic Detection of Steganographic Images in Network Traffic

Ömer Erdem[1], Metin Turan[2]

[1]Department of Computer Engineering, Istanbul Commerce University, 34840 Istanbul, Turkey
omer.erdem@istanbulticaret.edu.tr
[2] Department of Computer Engineering, Istanbul Commerce University, 34840 Istanbul, Turkey
mturan@ticaret.edu.tr

Abstract

Detection and prevention of data breaches in corporate networks is one of the most important security problems of today's world. The techniques and applications proposed for solution are not successful when attackers attempt to steal data using steganography. Steganography is the art of storing data in a file called cover, such as picture, sound and video. The concealed data cannot be directly recognized in the cover. Steganalysis is the process of revealing the presence of embedded messages in these files. There are many statistical and signature based steganalysis algorithms. In this work, the detection of steganographic images with steganalysis techniques is reviewed and a system has been developed which automatically detects steganographic images in network traffic by using open source tools.

1. Introduction

The increasing and widespread use of the Internet brings lots of advantages to each part of life. Besides, it has some security problems. Detection and prevention of data leakage is the important one of these. Especially, in terms of corporate networks, it is observed that the significance of this issue has increased even more in recent years [1]. Many commercial and open source Data Loss Prevention (DLP) solutions are used to prevent data theft. DLP applications are successful when evaluated with various parameters. Though, these solutions are inadequate if steganography or hidden channels are used [2].

Steganography is the process of storing confidential messages in digital media files such as pictures, videos and audios. Although it seems to be similar fields with cryptography, they are different. The most important distinction is that; in cryptography, while data is being transferred, it can be seen by someone that the data is encrypted, but in steganography, the transferred secret data cannot be directly recognized by someone. Steganography was used in the past for various reasons. For instance, confidential message transmission between two points, espionage, terrorism, etc. In addition to this, today, it is also used to leak critical information out of the company or institution such as trade secrets and agreements.

Steganalysis is a science of detecting embedded messages stored in any digital media by steganographic methods. It focuses on breaking the security of steganography. Steganalysis can be classified into several classes. There are many works for each class, but as a matter of course, the researches in the steganalysis are concentrated on the determination of the most commonly used steganography algorithms in practice.

There are significant researches for the automatic detection of steganography. N. Provos and P. Honeyman have developed a system that automatically retrieves JPEG images from the Internet and analyses to ascertain that hidden messages exist. They use statistical methods for detection and a distributed calculation method on loosely coupled servers for the purpose of realizing dictionary attacks [3]. G. Berg, I. Davidson et al. have proposed a machine learning algorithm to distinguish difference between normal and steganographic image. The designed machine learning model can automatically observe the message embedded images by using the trained steganography techniques. They also aimed the detection of more sophisticated and previously unseen steganography algorithms [4]. J. Taylor and M. Dailey have developed a software system that uses various steganalysis techniques and performs automatic analysis on all files or directory given as input to detect if input contains a specific type of steganography. Moreover, if possible, the system tries to automatically recover the hidden content [5].

In this work, a solution has been developed for detection the steganographic contents which DLP applications fail. The developed system provides extracting the image files from network traffic automatically, detecting whether the extracted file contains hidden data or not by using steganalysis techniques and producing an alert on the system according to the obtained result.

The rest of the paper is organized as follows. In Section 2, the literature review for steganography and steganalysis is given to clarify the research impact. Section 3 explains the basic features of the developed system as well as design, development and test phases in detail. Summary, conclusions drawn from the study and suggestions for possible future works have been given in Section 4.

2. Background and Related Works

2.1. Steganography

Steganography, which is a combination of Greek "steganos" meaning covered or protected and "graphein" meaning writing, is the art of hiding data in known cover medium by methods that make the existence of embedded data unknown. Covers such as picture, audio, video, text are completely harmless files. A stego key can also be used to embed the data. So, the process of creating steganographic content can be formulated as follows [6]:

Cover Medium (Known Data) + Embedded Message + Stego Key (Optional) = Stego Medium (Steganographic Content)

The first recorded use of steganography known in the history belongs to the Greek historian Herodotus (484-425 BC). Herodotus message was tattooed his slave's shaved head, then he waited for growing the slave's hair to conceal the message [7]. The use of invisible ink, which appeared only in light of certain wavelengths, hiding secret info into normal text messages by German spies are examples of steganography usage during World War II. Moreover, in 1999, C.T. Clelland and et al. have achieved to embed messages in DNA by altering the locations of organic bases [8].

The primary objectives for steganography algorithms are, capacity, imperceptibility, robustness and security [9]. Capacity is the maximum amount of data that will be embedded in the cover medium. Imperceptibility means that there is no visible difference between the cover and stego medium. Robustness is the resistance of algorithm to attacks and other operations such as rotations, scaling and filtering. Security is another important parameter that indicates even if hidden message is detected, it cannot be revealed.

There are several approaches to classify the steganography techniques. For example, they can be classified according to the types of cover medium or the cover alterations during the process of storing secret message. Steganography techniques are divided into 6 classes based on the modifications applied to cover medium [10]:

- *Substitution*: The process of replacing non-essential parts of a cover with a confidential message.
- *Transform domain*: Converts the cover into frequency domain, then hides secret messages in significant areas of cover image (e.g.: discrete cosine transformation).
- *Spread spectrum*: Steganography adaptation of communications using the spread spectrum technique.
- *Statistical*: Method of changing several statistical properties of a cover.
- *Distortion*: Technique of concealing a message by causing a signal distortion in cover medium.
- *Cover generation*: Generating a special digital media for using only the purpose of covering the secret message.

One of the most important substitution techniques is LSB (Least Significant Bit) method. The LSB substitution technique is based on the process of embedding the secret information to least significant (rightmost, smallest weight) bit of the pixels in the image. The change in original pixel value will be +1 or -1, so it will not cause any visible distortion in the image [11]. The pixels to be altered can be selected in order or randomly. Random selection is done with the pre-shared secret key between the communication pairs.

The many popular steganography software uses the LSB technique by reason of its ease of implementation. Hence, the researchers focus on LSB technique to determine the existence of hidden data in digital media.

2.2. Steganalysis

Steganalysis is a science aimed to detect the presence of hidden message embedded in a digital media by collecting sufficient evidence. In a few words, it is a countermeasure for steganography [12]. Other objectives include revealing, cleaning or destruction of the embedded message. Steganalysis is used in areas such as computer forensic analysis, cyber warfare, collection of evidence and tracking criminals on the Internet. Initial researches on steganalysis started in the late 90's. Earliest work was published by N.F. Johnson, S. Jajodia and R. Chandramouli et al. [13, 14].

Steganalysis looks like similar fields with cryptanalysis. But, the most important difference is that the data to be examined is known in cryptanalysis, whereas in steganalysis it is an extra challenge to identify those which are suspicious of large data sets. However, attempts for detection of steganographic content are similar to cryptanalysis attacks. These attack types include stego only, known cover, known message, chosen stego, chosen message and known stego [15].

Steganalysis techniques can be classified based on various parameters. A general hierarchy is shown in Fig. 1 that is a result of work on classification [16].

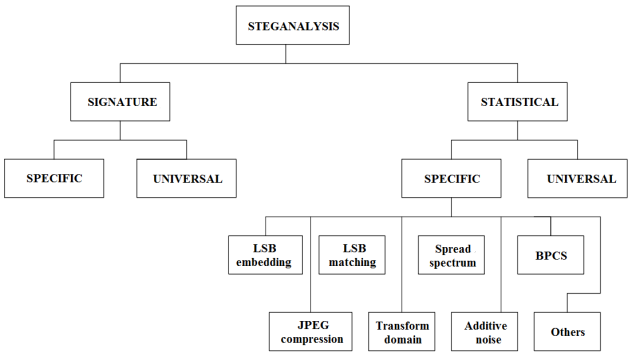

Fig. 1. Classification of steganalysis techniques

Steganalysis techniques can be classified into two main classes as signature based and statistical based. In signature-based techniques, altered, corrupted, or generated media properties during steganography act as signatures to sense the existence of hidden message. For instance, in GIF images, these techniques basically look palette tables and anomalies. Because, many steganography applications that use the LSB method leave these signatures for GIF images [17]. Statistics-based techniques use a variety of statistical calculations on the image to detect the embedded message. They are more efficient techniques than signature based since they use mathematical methods.

A number of works have been reported about statistical steganalysis and LSB embedding analysis targeting the detection of LSB steganography. The first statistical technique was proposed by A. Westfeld and A. Pfitzmann. A statistical Chi-Square test is designed for the detection of hidden messages based on the Pairs of Values (POV) changes. The result of work shows that this method is successful in detecting sequentially embedded messages [18]. Raw Quick Pair (RQP) is proposed by J. Fridrich et al.. In RQP, the increase in the ratio of the number of close colors to the total number of unique colors is analyzed. Although the method gives reliable results, it does not work in grayscale images [19]. Another method is presented by L. Zhi et al. and named as Gradient Energy-Flipping Rate Detection (GEFR). The technique is based on the relation between the embedded message length and the gradient energy form of image. As an output of this work, the presence of the secret message with an embedding rate greater than 0.05 bpp (bits per pixel) could be reliably detected [20]. RS steganalysis method is proposed by J. Fridrich et al. and it is clear that this technique has more advantages when compared with other works on LSB steganalysis. The advantages can be summarized as follows; detects messages in randomly scattered pixels with a minimum

size of 0.03 (bpp), works like a charm in both color and grayscale images, shows better performance than other techniques [21]. Therefore, in this paper, RS steganalysis method is chosen and implemented for the steganography detection.

RS steganalysis basically use sensitive dual statistics derived from the spatial correlation in the image. According to the technique, in the first step, the image is divided into pixel groups with fixed size. The distribution of the pixels is calculated by summing the differences of the pixels in the group. The goal is to quantify the smoothness or regularity among pixels. Then, some of the pixels in each group are selected with a mask and an invertible operation called "flipping" applied to them. Flipping, simply, changes pixel values according to the mask by considering that they are odd or even. Even values are incremented 1 and odd values are decremented 1 or vice versa. The distributions in the resulting image are calculated and compared with the distributions in the original image to form groups of pixels called "Regular", "Singular" and "Unusable". The same operations are repeated for different masks. The values obtained from relations between groups by using different masks are placed to a quadratic equation. Then, estimated message length is calculated by a mathematical method using the root of this equation [21, 22]. In the end, the message length is compared with a determined threshold value to give a decision whether the image has hidden content.

3. Proposed Detection System

In this section, the developed system for the automatic detection of steganographic images in network traffic has been explained deeply in two sections: design & development and testing & evaluation.

3.1. Design & Development

The main goal of developed system is to extract images from network traffic, automatically detect those with steganographic content by using steganalysis methods and generate alarms. The system consists of two phases. In the first phase, the images are selected from network traffic and in the second phase, steganographic content is searched in these images by using developed application. Fig. 2 shows the general structure of the system and its positioning in a sample network topology.

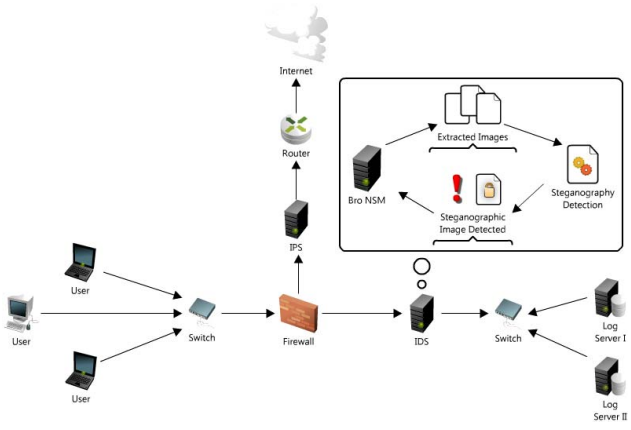

Fig. 2. General structure of the developed system and positioning in the network topology

The main tasks of the first phase is to extract images from network traffic, send extracted file information as input to the steganography detection phase and generate an alarm on the system according to the result returned from the detection phase. Bro NSM is used to realize these steps. Bro is an open source, UNIX based, BSD licensed intrusion detection system (IDS), network analysis and monitoring tool. It is a complete network traffic analysis framework unlike other classic signature-based IDSs. Traffic analysis is not only related security, but also includes performance analysis and network troubleshooting. Some important features of Bro NSM are as follows; keeping records of many network activities into tab-separated parseable log files, port independent protocol analysis, capturing info about files in HTTP, FTP, SMTP, IRC traffic and saving it with MD5/SHA1/SHA256 values and also extracting these files to save a chosen directory in the system [23]. Moreover, Bro NSM has its own scripting language and it is the most significant feature that differentiates Bro from other intrusion detection systems. So, it has a very flexible and developable structure. Each user can customize and increase the functionality of the system with custom scripts [24]. In this work, a Bro NSM script has been developed that provides to perform the first phase tasks. More specifically, it extracts images in PNG, JPEG, BMP formats, save them to a specified directory, send the necessary file information to the steganography detection phase for analysis and generate an alarm in the system according to the received result. Bro events that describe the activity of any protocol in network are used in this script. The flow diagram of the script is shown in Fig. 3.

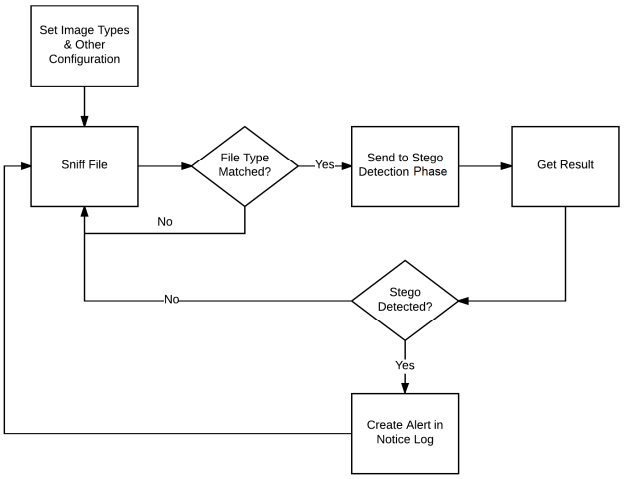

Fig. 3. Flowchart of the developed Bro NSM script

The second phase decides whether there are embedded messages in the images extracted from the network traffic. It takes the file path on system as input, analyses the file and send result to the first phase. RS Steganalysis is used for detection because of the advantages it provides compared to other statistical steganalysis techniques. The algorithm is implemented using the Python programming language. The development environment consists of Linux, Python 2.7, OpenCV (an open source computer vision library), NumPy (a fundamental package for scientific computing with Python) and Gedit [25]. The ZigZag algorithm is applied for JPEG images unlike the basic functions of the RS Steganalysis algorithm. The flowchart of the developed application is shown in Fig. 4.

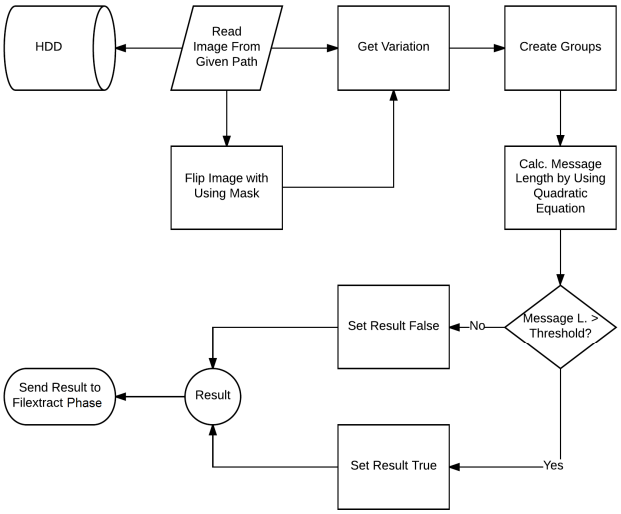

Fig. 4. Flowchart of the steganography detection phase

3.2. Test & Evaluation

The following parameters are used for testing the developed system:

- Different protocols such as HTTP, SMTP, FTP
- Various image file formats such as JPG, BMP, PNG
- Colorful and grayscale images
- A variety of resolutions
- Embedded messages in different sizes
- Various steganography tools

Steganographic images are created using tools running on Linux or Windows. Tools used on Linux are Steghide, OpenStego, Stego Suite, StegoShare, Steganography Studio, VSL, LSBSteg and on Windows are SteganPEG, SSuite Picsel, OpenPuff. These tools are chosen since they all implement the LSB technique in message hiding.

A data leak scenario has been realized in the system. According to this, the attacker hides confidential information about his company to an image file and uploads it to a web site. The image is uploaded to the website via HTTP protocol. This attack was detected at runtime and generated an alert on the system. The log records of the related scenario on the system are shown in Fig. 5 and Fig. 6.

Fig. 5. Extracting image files from network traffic

Fig. 6. Alarms generated in the system according to result of the steganography detection phase

The scenarios that data theft with e-mail using SMTP, uploading files to remote server via FTP are tested using different file types. It is observed that the steganographic images with embedding rate greater than 0.03 bpp (as a feature of the RS Steganalysis algorithm) automatically detected similar to HTTP scenario. Furthermore, in case of suspicion, the scenario of analyzing the captured traffic in PCAP format is tested and it is seen that the same results are obtained like real time network traffic.

4. Conclusion and Future Work

DLP solutions are used to detect and prevent data breaches in corporate networks that is one of the major security problems. However, these applications cannot detect the steganographic content. So, steganography still can be used to leak data. In this work, a system has been developed that automatically detects steganographic images in network traffic using open source tools and implemented algorithms. The developed system is a solution for the mentioned problem that DLP applications fail.

There are some possible improvements and future works for developed system. Other statistical steganalysis algorithms can be added to the system in order to minimize errors like false positive or false negative in determining the steganographic content. For this purpose, calculating the average of the different analysis results or calculating an average with different coefficients according to the algorithm can be implemented. In addition, various updates can be added to the application to improve performance since in statistical steganalysis techniques, pixel based processing is performed.

7. References

[1] Internet Crime Complaint Center (IC3) - Federal Bureau of Investigation (2017, June 10). *Internet Crime Report* [Internet]. Available: https://pdf.ic3.gov/2016_IC3Report.pdf.

[2] K. Fiscus, “DLP Fail! Using Encoding, Steganography and Covert Channels to Evade DLP and Other Critical Controls”, in *Hackfest*, Quebec, Canada, 2015.

[3] N. Provos, P. Honeyman, “Detecting Steganographic Content on the Internet”, *CITI Technical Report*, Aug., 2001.

[4] G. Berg, I. Davidson, M.Y. Duan, G.Paul, “Searching For Hidden Messages: Automatic Detection of Steganography”, in *Innovative Applications of Artificial Intelligence*, Acapulco, Mexico, 2003, pp. *51-56*.

[5] J. Taylor, M. Dailey, (2017, June 12). *Automatic Detection of Steganographic Content*, [Internet], Available:

https://www.cs.cf.ac.uk/PATS2/@archive_file?c=&p=file&p=578&n=final&f=1-C1129788_Final_Report.pdf.

[6] N.F. Johnson, Z. Duric, S. Jajodia, "Information Hiding: Steganography and Watermarking-Attacks and Countermeasures", Springer Science & Business Media, New York, USA, 2001.

[7] H. Wang, S. Wang, "Cyber Warfare: Steganography vs. Steganalysis", *Communications of the ACM*, vol. *47*, no. *10*, pp. *76-82*, Oct., 2004.

[8] CT. Clelland, V. Risca, C. Bancroft, "Hiding Messages in DNA Microdots", *Nature*, vol. *399*, no. *6736*, pp. *533-534*, Jun., 1999.

[9] P. Sharma, P. Kumar, "Review of Various Image Steganography and Steganalysis Techniques", *IJARCSSE*, vol. *6*, no. *7*, July, 2016

[10] S. Katzenbeisser, F. Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermarking", Artech House, London, England, 2000.

[11] P. Rai, S. Gurung, M.K. Ghose, "Analysis of Image Steganography Techniques: A Survey", *IJCA*, vol. *114*, no. *1*, Mar., 2015.

[12] S.M. Badr, G. I. Salama, G. M. I. Selim, A. H. Khalil, "A Review on Steganalysis Techniques: From Image Format Point of View", *IJCA*, vol. *102*, no. *4*, Sept., 2014.

[13] N. F. Johnson, S. Jajodia, "Steganalysis of Images Created Using Current Steganography Software", in *Information Hiding*, Springer, Heidelberg, Berlin, 1998, pp. *273-289*.

[14] R. Chandramouli, Li Grace, N. D. Memon, "Adaptive Steganography", Proc. of the SPIE, in *Security and Watermarking of Multimedia Contents IV*, San Jose, CA, 2002, pp. *69-78*.

[15] S. K. P. Reddy, "Steganalysis Techniques: A Comparative Study", M.S. thesis, CS, University of New Orleans, New Orleans, USA, 2007.

[16] A. Nissar, A.H. Mir, "Classification of Steganalysis Techniques: A Study", *Digital Signal Processing*, vol. *20*, no. *6*, pp. *1758-1770*, Dec., 2010.

[17] M. Kaur, G. Kaur, "Review of Various Steganalysis Techniques", *IJCSIT*, vol. *5*, no. *2*, pp. *1744-1747*, 2014.

[18] A. Westfeld, A. Pfitzmann, "Attacks on Steganographic Systems", in *International Workshop on Information Hiding*, Heidelberg, Berlin, 1999, pp. *61-76*.

[19] J Fridrich, M Long, "Steganalysis of LSB Encoding in Color Images", *IEEE ICME*, vol. *3*, pp. *1279-1282*, Aug., 2000.

[20] L. Zhi, S. A. Fen, Y. Y. Xian, "A LSB Steganography Detection Algorithm", *IEEE PIMRC*, vol. *3*, pp. *2780-2783*, Sept., 2003.

[21] J. Fridrich, M. Goljan, R. Du, "Reliable Detection of LSB Steganography in Color and Grayscale Images", *IEEE Multimedia*, vol. *8*, no. *4*, pp. *22-28*, Oct., 2001.

[22] R. Chhikara, L. Singh, "A Review on Digital Image Steganalysis Techniques Categorised by Features Extracted ", *IJEIT*, vol. *3*, no. *4*, pp. *203-213*, Oct., 2013.

[23] L. Randall, "Bro and Bro IDS", in *ShmooCon*, Washington, USA, 2013.

[24] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-Time", *Computer Networks*, vol. *31*, no. *23-24*, pp. *2435-2463*, Dec., 1999.

[25] A. Mordvintsev, K. Abid, (2017, June 15). *OpenCV-Python Tutorials Documentation* (Release 1) [Internet]. Available: https://media.readthedocs.org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals.pdf.