APPROACH ANALYSIS TO PREVENT STEGANOGRAPHY DATA FLOWS AS A CORPORATE SECURITY BREACHES

Lopit P.[1,2], Assoc. Prof. PhD. Alieksieiev V.[1,3]

[1] Lviv Polytechnic National University, Lviv, Ukraine

[2] polina.velebnik.pm.2015@lpnu.ua

[3] vladyslav.i.alieksieiev@lpnu.ua, https://orcid.org/0000-0003-0712-0120

Abstract: *Once the steganography was invented, it still remains one of the most effective ways to hide information in publicly accessed data sources. Many applications of different steganography techniques were known since WWII, but there are no yet any good approach to detect whether there is a hidden message in some data source (image etc.) while it is not expected to be found. This may also cause a fundamental security breach in corporate security systems. In the era of Facebook, Instagram and many other social networks with a huge amount of images and photos it seems obvious to use steganography applications for the purpose of security breach. A saboteur or a spy can use any image from social networks to disclose corporate sensitive data or to exchange hidden messages. That is why steganography detection should be considered as a necessary tool to prevent security breaches and sensitive data leaks. This research is focused on analysis of current approaches in problem of steganography detection. Mainly the approaches to detect steganography in images were discussed.*

Keywords: STEGANOGRAPY, STEGANOGRAPHY DETECTION, CORPORATE SECURITY, DATA FLOWS, SECURITY LEAKS, WEB BROWSER EXTENIONS

1. Introduction

Steganography is a well-known method of cryptography [1]. There is a number of implementations to hide some information within some carriers (i.e. in an image). Common cryptography techniques usually assume the possibility of secret message interception. This guaranties any necessary security level. Unlike to that the steganography relies on the principle, that the hidden or secrete information should be hidden within a public visible carriers and no one should guess there is something hidden.

As many reviewers mention the steganography approaches are in use since ancient ages [2]. Obviously, people are very inventive in their passion to hide secretes they have. First steganography approaches used texts and paintings to hide secret messages. Due to a technologies development there are many new carriers appeared in steganography [3–7]. Since a widespread use of computers and computer networks, the steganography came into all media carriers, including texts, images, audio, video and IP datagrams.

There are many known useful techniques of copyright and theft protection via steganography watermarking to protect media data in the Internet. On the other hand, some modern printers use steganography techniques to watermark each printed page and make an invisible signature allowing printer identification. Digital content protection is the area where steganography was successfully used. Most Internet users do not even think they are forced to see steganography processed media (images, photos etc.) at many websites trying to protect their authorship. All these steganography applications prove the fact the steganography became an integral part of our modern life.

2. Prerequisites and means for solving the problem

Nevertheless, there is a number of reports revealing facts about illegal use of steganography for the villain's purposes, including intelligent services' spies [8] and terrorists [9]. These facts turn us to thoughts of security breaches, corporate data protection and preventing sensitive data leaks. Each company is interested in analysis of possible channels of data flows to be able to identify the breach.

In the era of mobile devices, social networks and mobile internet, allowing any common person to become a "spy", it is of a great importance for cybersecurity purposes to unveil possible channels of leaks, to identify menacing data flows and not to break civil rights simultaneously. Each corporate worker uses to download and looks through tons of text and image information every day. Evidently, this could be the easiest way for hostile informer to use hidden messages in an invisible manner within a publicly visible carrier.

For example, a hidden message transfer scenario may include taking and sharing selfies with some custom encoding/decoding application for a mobile device. However, from the spy point of view, this scenario is vulnerable to disclosure with physical evidences of malicious activity (the device with software installed in it). Another scenario may involve visiting some website performing all the necessary steganography processing. This scenario can envisage some "independent" steganography processing service with anonymous access with just encoding secrete message into user's image and displaying the hidden message after decoding a steganographic image (the image can be given by URL). Combining such service with some "incognito" browser mode and "anonymizer" web service may allow eliminating any digital evidences at spy's device.

The idea of transferring steganography-processed images over public networks is not a new one. Moreover, there is a number of researches with its implementations [5–7]. Some authors (like [5, 6]) are using their custom approaches of LSB algorithm. There are suggestions on how to hide information in a publicly available billboard display [5] and suggestions on how to detect presence of steganographic information in publicly available content (jpeg-images) from Internet [7].

Of course, using social networks, which are performing their own image processing with compression, makes it difficult to guarantee lossless data transfers over these networks. Nevertheless, these networks can be used to share links to some "funny" images at third-party websites. The best candidates for being used as a channel for steganography data flows could be websites providing original (not changed) media content to its visitors.

Applying any of depicted scenarios leads us to a conclusion, that there should be some way either to check all corporate incoming and outgoing Internet traffic (check for steganographic data on a firewall, proxy etc.) or to check Internet traffic directly at workplaces in a web browsers. The ideal solution could be a corporate web browser to prevent steganographic leaks with embedded content inspection. Unfortunately, this solution is very expensive and requires many resources to support such custom web browser. However, all modern web browsers support extensions or plugins. This is a convenient way to extend a browser functionality without intruding the browser itself.

3. Solution of the examined problem

Our approach to solve the problem of preventing steganography data flows will exploit the opportunity to use web browsers plugins. Such plugin should match the following requirements:

- Plugin should analyze content of a web page, the user (company employee) visits.
- Plugin should extract images from the HTML-code and perform detection of hidden steganographic information. Statistics analysis methods should be used to analyze this information.
- Plugin should have a modular structure to allow further improvement, refactoring and support, including ability to add new methods of hidden information detection.
- Plugin should block suspicious data transfers and send a notification to security department with detailed description of possible malicious activity.

As an initial approach, we will use a good known LSB algorithm. Building the described above plugin should allow first a simple detection of LSB encoded steganography for bitmap images (BMP image format), and the most popular in the Internet image formats: JPEG-images and PNG-images. Next, we will develop and add methods for steganographic data detection in other popular graphic formats. Our current target is to develop the plugin as a platform for its further improvement via embedding new methods.

4. Results and discussion

Our plugin parses the HTML code, and extracts the list of images from of a web page. Then it performs a test of each image for possible hidden content in the image. Fig. 1 presents the example of parsing and image extraction from HTML.

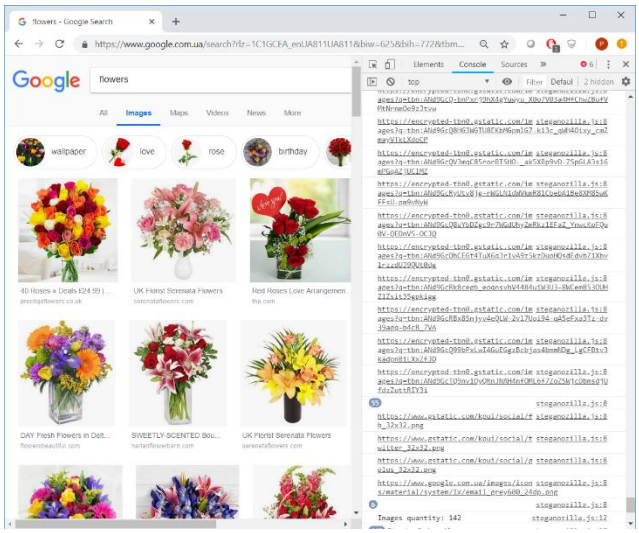

Fig. 1 *Parsing and extracting list of images from HTML*

We prepared an image to perform a test of our plugin working on detection of possibly hidden data within the image. Then we were adding a hidden data (text file named "cubaan.txt" with text string in it: "alpacas rule the world") to the image using LSB algorithm.

Fig. 2 shows the original JPEG-image. The steganographic JPEG-image with a hidden data shown at Fig. 3.

Fig. 4 presents the original PNG-image and the steganographic PNG-image with hidden data shown at Fig. 5.

As one can see we have successfully processed with steganographic data insertion most popular image types – JPEG and PNG. Results of steganographic processing for BMP images are good known, so we do not present the examples here in this paper. All images of these three types are detected if a hidden steganographic data appears there. It is a very limited solution yet, because we only used popular LSB steganography algorithm in our plugin as method of encoding. Nevertheless, we had built a good initial platform solution as an approach for the required web browser plugin. This plugin allows easy extension later with new modules to implement detection of more steganography encoding methods.

5. Conclusion

Here in the paper we offer an initial problem analysis and a test implementation of steganography detection plugin for a web browser. The set of requirements for such kind of an online security solution was offered. Following these requirements, we are going to improve the developed plugin and allow a wider set of image types (not only BMP, JPEG and PNG) to be analyzed on-the-fly. Based on the presented ideas a security web-service can be developed. The service may gather and accumulate security alerts from different users and help to identify most typical channels of steganography data flows.

6. Literature

1. Steganography – Wikipedia // Wikipedia.org – 01.10.2018. – https://en.wikipedia.org/wiki/Steganography
2. Steganography: Past, Present, Future / James C. Judge // SANS Institute, 2001. – 29 p. – Retrieved from: https://www.sans.org/reading-room/whitepapers/stenganography/paper/552
3. Review on steganography for hiding data / Sagar S.Pawar, Vinit Kakde // International Journal of Computer Science and Mobile Computing – Vol. 3 Issue 4, Apr. 2014, P. 225-229. – Retrieved from: https://www.ijcsmc.com/docs/papers/April2014/V3I4201468.pdf
4. Steganography and Steganalysis: Different Approaches / Soumyendu Das, Subhendu Das, Bijoy Bandyopadhyay, Sugata Sanyal // ArXiv.org, 2011 – Retrieved from: https://arxiv.org/ftp/arxiv/papers/1111/1111.3758.pdf
5. Steganography An Art of Hiding Data / Shashikala Channalli, Ajay Jadhav // International Journal on Computer Science and Engineering – Vol. 1(3), 2009. – P.137-141 – Retrieved from: https://arxiv.org/ftp/arxiv/papers/0912/0912.2319.pdf
6. Unseen: An Overview of Steganography and Presentation of Associated Java Application C-Hide / Jessica Codr // Washington University in St. Louis, Apr. 2009. – 21 p. – Retrieved from: https://www.cse.wustl.edu/~jain/cse571-09/ftp/stegano.pdf
7. Detecting Steganographic Content on the Internet / Niels Provos, Peter Honeyman // CITI Technical Report 01-11, Center for Information Technology Integration, University of Michigan, Aug. 2001. – 14 p. – Retrieved from: http://www.citi.umich.edu/techreports/reports/citi-tr-01-11.pdf
8. Criminal complaint by Special Agent Ricci against alleged Russian agents (PDF). United States Department of Justice, 28.06.2010 – http://www.justice.gov/opa/documents/062810complaint2.pdf
9. Jack Kelley. Terror groups hide behind Web encryption. USA Today, Feburary 2001. – http://www.usatoday.com/life/cyber/tech/2001-02-05-binladen.htm

Fig. 2 *Original JPEG-image (before steganographic processing)*

Fig. 3 *JPEG-Image with hidden data (after steganographic processing)*

Fig. 4 *Original PNG-image (before steganographic processing)*

Fig. 5 *PNG-Image with hidden data (after steganographic processing)*