Steganography Analysis: Efficacy and Response-Time of Current Steganalysis Software

Jordan Green, B.S.[1], Ian Levstein, M.S. [1], Cpl. Robert Boggs[2], Terry Fenger, Ph.D.[1]

[1]Marshall University Forensic Science Center, 1401 Forensic Science Drive, Huntington, WV 25701

[2]West Virginia State Police Digital Forensics Unit, 1401 Forensic Science Drive, Huntington, WV 25701

Abstract

Steganography is writing hidden in plain sight. For law enforcement, this form of hiding data can be a problem in the discovery of traded, illicit information. Steganalysis software such as StegAlyzer™ aids law enforcement by discovering hidden data.

This study found that message and carrier size differences do not affect StegAlyzer™'s analysis time. Additionally, StegAlyzer™ identified five out of nine downloaded applications, and two steganography signatures from six of those applications.

Introduction

Steganography grows more complex with an increase in open source applications designed to hide data. StegAlyzer™ is software designed to find steganography and its applications.

This study examined StegAlyzer™'s abilities against open-source steganography applications and investigated three questions:

Question 1: Does size and format of carrier images or message images affect steganalysis-time?

Question 2: How well does StegAlyzerAS™ detect multiple applications?

Question 3: How well does StegAlyzerSS™ detect steganography from various applications?

Materials and Methods

Question 1: A steganography appending application was downloaded and used to create steganography.

Images were used to test the analysis-time of StegAlyzer™ for different message formats and sizes.

Steganography files were analyzed using Backbone Security's StegAlyzerSS™ v3.91 (x86) and the analysis times were recorded.

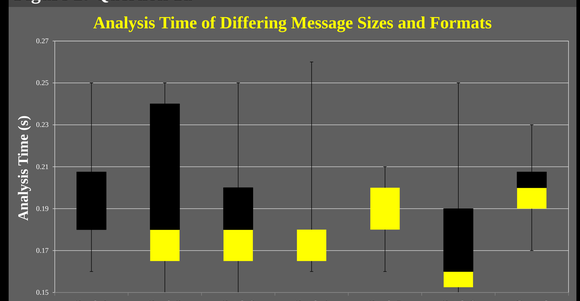
Question 2: Nine applications were downloaded and analyzed using StegAlyzerAS™.

Question 3: Steganography was created from six applications and analyzed using StegAlyzerSS™.

Results

Question 1: **Figure 2** represents changes in analysis time with differing message file sizes. There was no statistical difference between groups ($F_{2.25} = 0.87$).

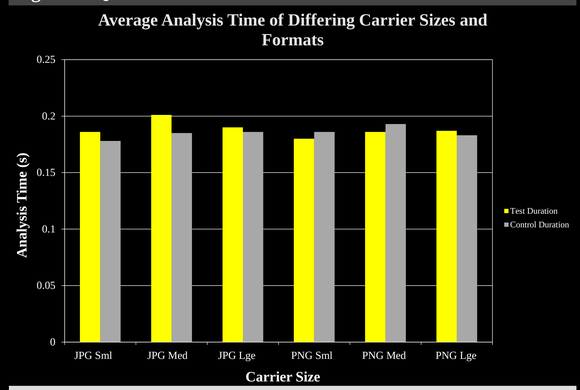
Figure 2: Question 1a



Run-times of analyzed images. Carrier images were 5 MB JPEGs (n = 10). Batch 1 was embedded with a 34 KB .doc file; Batch 2, a 103 KB .doc file; Batch 3, a 1 MB JPG file; Batch 4, a 10 MB JPG image; Batch 5, a 1 MB PNG image; Batch 6, a 10 MB PNG image. The control had no embedded media

Figure 3 shows analyses of the same 5 MB image embedded within six differently sized and formatted image carriers.

Figure 3: Question 1b



The average run-times for each group of images (n = 10). Experimental images were embedded within the same JPG image, 5 MB in size. JPG Sml and PNG Sml represent an image size of 1 MB of corresponding image formats; JPG Med and PNG Med were 5 MB in size; JPG Lge and PNG Lge were images 10 MB in size. Controls had no embedded message images.

Statistical results confirmed no significant difference between groups ($F_{1.87} = 0.55$).

Question 2: Of the nine applications analyzed, StegAlyzerAS™ discovered signatures from five (**Figure 4**).

Figure 4: Results of StegAlyzer's Analysis of Steganography Applications

Application	Embed Method	Embed within	StegAlyzerAS Detection
SecretLayer	LSB	PNG	No
SilentEye	LSB	BMP, WAV	Yes
GhostHost	Append	All images, audio, text, and video	Yes
ImageSpyerG2	LSB - robust soliton distribution	BMP, TIF	No
OpenPuff	LSB - non-linear coding	BMP, JPG, PCX, PNG, TGA, AIFF, MP3, NEXT/SUN, WAV, 3GP, MP4, MPG, VOB, FLV, SWF, PDF	Yes
OpenStego	LSB, Watermarking	JPG, TXT, PNG, BMP	No
Steg	LSB	JPG, TIF, PNG, BMP, PPM	No
SteganographyStudio	LSB	BMP, PNG, GIF	Yes
Steghide	LSB – non-linear coding	JPG, BMP, WAV, AU	Yes

Of the applications StegAlyzerAS™ scanned, five out of the nine applications were discovered: Steghide, SilentEye, OpenPuff, Virtual, GhostHost, and Steganography Studio. These applications embed least significant bit (LSB), watermarking, and appended steganography into various file formats.

Question 3: **Figure 5** shows example analyses of least significant bit steganography and appended steganography.

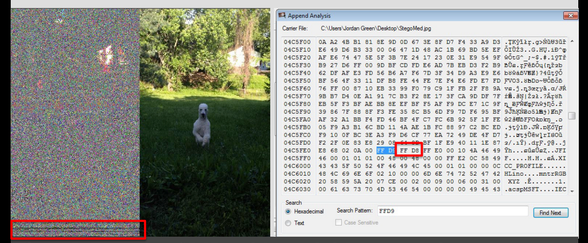


Figure 5: Examples of LSB and Appended Steganography

In least significant bit steganography, a lattice may be observable when running steganalysis software. In this image, the steganography is contained within the bottom portion of the carrier. Appended steganography begins its message code after the completion of the carrier's code, in this case at hexadecimal FF D9. A hex-editor can easily detect appended steganography.

Two signatures were detected by StegAlyzerSS™. **Figure 6** provides details of this analysis.

Figure 6: Results of StegAlyzerSS Analysis of Steganography

Steganography Application	Cover Image	Message	Steganography File	StegAlyzerSS Detection
SecretLayer	StegoLge.jpg, 9.764 MB	RouxRun.Lg, 103 KB	SecretLayer.jpg, 9.764 MB	No
GhostHost	StegoLge.jpg, 9.764 MB	RouxRun.Lg, 103 KB	GhostHost.jpg, 9.867 MB	SS, Append
ImageSpyer G2	StegoLge.jpg, 9.764 MB	Roux.txt, 1 KB	ImageSpyerG2.bmp, 62.53 MB	LSB
OpenPuff	StegoLge.jpg, 9.764 MB	Roux.txt, 1 KB	OpenPuff.jpg, 9.764 MB	No
OpenStego	StegoLge.jpg, 9.764 MB	RouxRun.Lg, 103 KB	OpenStego.png, 9.740 MB	No
Steg	StegoLge.jpg, 9.764 MB	RouxRun.Lg, 103 KB	Steg.jpg, 9.762 MB	No
Steganography Studio	NA	NA	NA	NA

Conclusions

Statistical analyses revealed that StegAlyzer™ analysis time is not affected by message or carrier size or Format. StegalyzerAS™ detected five out of nine applications. StegAlyzerSS™ detected two signatures from six different applications.

References

Aggarwal S, Jaiswal U. Kryptos+Graphein= Cryptography. Int J Eng Sci Technol 2011;3(9):7080-4.

Ashok J, Raju Y, Munishankaraiah S, Srinivas K. Steganography: An Overview. Int J Eng Sci Technol 2010;2(10):5985-92.

Atoum MS, Ibrahim S, Sulong G, M-Ahmad A. MP3 Steganography: Review. Int J Comput Sci 2012;9(6):236-44.

Boora M, Ghambir F. Binary Image Steganography. Int J Recent Technol Eng 2013;2(5):126-31.

Cheddad A, Condell J, Curran K, McKevitt P. Digital Image Steganography: Survey and analysis of current methods. Sign Proc 2010;90(3):728-50.

Fogie S. Steganography. Informit.com. Pearson Education 2014. Accessed: June 25, 2014.

Gadichal AB. Audio Wave Steganography. Int J Soft Comput Eng 2011;1(5):174-6.

Image SpyerG2. ITNTSRL. http://imagespyer-g2.soft32.com/. Accessed June 03, 2014.

Open Puff. Embedded SW. http://embeddedsw.net/OpenPuff_Steganography_Home.html. Accessed June 03, 2014.

Open Stego. GNU. http://www.openstego.info/. Accessed June 03, 2014.

Raphael J, Sundaram V. Cryptography and Steganography – A Survey. Int J Comput Tech Appl 2011;2(3):626-30.

Secret Layer Steganography. Easy Sector. http://www.steganographypro.com/. Accessed June 03, 2014.

Silent Eye. http://www.silenteye.org/. Accessed June 03, 2014.

Steg. Drupal Gardens. http://steg.drupalgardens.com/. Accessed June 03, 2014.

Steganography Analysis and Research Center. www.sarc-wv.com. BackBone Security 2014. Accessed: June 03, 2014.

Steganography Studio. Source Forge. http://stegstudio.sourceforge.net/. Accessed June 03, 2014.

Steghide. http://steghide.sourceforge.net/. Accessed June 03, 2014.

Yugala K. Steganography. Int J Eng Trends Technol 2013;4(5):1629-35.

Acknowledgements

We thank Backbone Security and Chad Davis for technical support; we also thank the MUFSC for providing resources used during this project.