12-4-2013

Steganographic Checks In Digital Forensic Investigation: A Social Networking Case

Brian Cusack
Edith Cowan University, brian.cusack@aut.ac.nz

Aimie Chee
Auckland University of Technology, aimie.chee@aut.ac.nz

DOI: 10.4225/75/57b3c2edfb86b
11th Australian Digital Forensics Conference. Held on the 2nd-4th December, 2013 at Edith Cowan University, Perth, Western Australia

STEGANOGRAPHIC CHECKS IN DIGITAL FORENSIC INVESTIGATION: A SOCIAL NETWORKING CASE

Brian Cusack[1,2], Aimie Chee[1]
[1]Auckland University of Technology, Auckland, New Zealand
[2]Security Research Institute, Edith Cowan University, Perth, Australia
brian.cusack@aut.ac.nz, aimie.chee@aut.ac.nz

Abstract
Steganography is an ancient art that has received a mega boost in the digital age. Electronic communications are easily accessible by most people and have a wide range of opportunities to embed secret messages in a diverse range of cover objects. Our research questions were: What can an investigator do to check for hidden messages in social media? And, how much searching is enough? The testing was conducted in replicated social networking sites and digital images were selected as the cover objects. The research findings showed that steganography is as easy as sending an email and not much more difficult than downloading and using one of the many steganographic tools available online. Our advice is that investigators do check for hidden messaging in digital media and that the best practice guide developed be used as a minimal baseline.

Keywords
Steganography, Social Networking, Cover Objects, Digital Investigation, Best practice

INTRODUCTION

Hidden messaging is an ancient art that has received a mega boost in the digital age (Dunbar, 2002; Ashok, Raju, Munishankaraiah & Srinivas, 2010; Fridrich, 2010). Most digital investigators are alert to cryptographic and watermarking methods used to protect information but often hidden messaging has complexities that even the best tools can only give a percentage strike rate (Kipper, 2004). The barriers for anyone to use electronic communication channels has greatly reduced, the rich media capability increased and the availability of software to perform complex tasks is easy to access. As a consequence messaging for public and private good are valuable assets in the social and economic networks of relationships that drive business and community relations. A positive side is always balanced by a negative exploitation (Castigilione, D'Alessio & De Santis, 2011). All communication media can be used for criminal purposes and the undermining of legitimate activities. The digital mediums permit open communication and, consequently, the potential of hidden message propagation (steganography). The rich opportunities in social networking sites present a vast scope for messaging in images, text, sound files and so on. The purpose of our research was to answer: *What can an investigator do to check for hidden messages in social media?* And given the extensive scope for hiding messages the investigator requires guidance on the sufficiency of any given search (Berg, Davidson, Duan & Paul, 2003). Our research focused on the social networking sites' management of images as a way of eliminating potential cover objects and observing others. Tool testing is out of scope in this paper.

The research testing was carried out in a laboratory environment using scenarios that contained multiple test runs. In the pre-test, five steganographic techniques with different image formats were uploaded on Facebook and Google+ social network websites and then downloaded to identify the techniques that may or may not be used (Curran & Duvitt, 2008; Cheddad, Condell, Curran & McKevitt, 2010). A full cycle for covert communication up to the extraction of the hidden messages was executed (Hosmer, 2006). Two suitable techniques, *JP Hide and Seek* and *StegHide* with common JPEG images were chosen for the experimental case scenarios, based on the pre-test results. The experimental case scenarios were simulated on laboratory computers and digital forensic examinations were undertaken to identify both the uploaded hidden messages in different images and to extract the hidden messages in the uploaded and downloaded image files (Potdar, Khan, Chang, Ulieru & Worthington, 2005; Zax & Adelstein, 2009). Based on the digital forensic examinations performed, a guideline for the steganographic examination process was also established (Hayati, Potdar & Chang, 2007; Hamid, Yahya, Ahmad & Al-Qershi, 2012).

BACKGROUND PRE-TESTS

The focus of our research was images in the social networking environment and for exploratory pre-testing we uploaded and downloaded a number of images that had hidden messages that were prepared by a number of different steganographic tools. The results served to limit the scope of our main tests and to disclose OSN image management methods. The findings from the pre-test showed that steganography is difficult to perform in the Facebook photo upload feature. Here the hidden message cannot be extracted after the image is downloaded from Facebook, but it can be successfully performed through the message file attachment and group file sharing features with a variety of image formats such as JPEG, PNG, BMP, and GIF. With Google+ photo sharing, the complete cycle of steganographic communication from embedding up to the extraction of hidden messages was successfully undertaken with JPEG, PNG, BMP or GIF image formats. The results demonstrate that steganography can be propagated in social media; therefore it is necessary to include steganographic evaluation in the standard digital investigation procedures. The research found that there is a lack of effective forensic tools in the area of steganographic image analysis or signature detection. The current steganalysis tools are designed for specific signatures but signatures continue to evolve and even the current set may not be complete. This is a challenge for the professionalism of the digital forensic investigator who must comply with acceptable methodologies but may be using inadequate tools.

Free steganographic tools available on the market are capable of performing information hiding in formats including BMP, GIF, JPEG or PNG. The success or otherwise of hidden messaging in social networking sites is dependent on first understanding how the tools function and then on how images are processed on the social networking sites along with any restrictions the site has for photo sharing. Usually online social networking sites (OSNs) have policies that constrain the size and format of an image and non-conformant images are either rejected or auto compressed, cropped, resized, or reformatted. This modification is serious for images embedded with a secret message, as any of the modifications may destroy or damage the hidden message. Steganographic tools generally rely on the structural stability of an image and exploit the standard properties of an image format. When these parameters are altered by the OSN site management system then the hidden messages may become corrupted to the point where they are non recoverable. Many OSNs pre-process the uploaded images before publishing them on the user’s content pages. The result is that the images’ characteristics are modified. A survey of three OSNs (Facebook, Badoo, and Google+) identified that all three OSNs change the pixel resolution and metadata of uploaded pictures to fixed values. Facebook and Badoo use pre-defined JPEG quantization tables to compress the images and only accept JPEG image files with any other image format being automatically converted to JPEG. Google+ is more flexible with JPEG, BMP, PNG and GIF image formats accepted for uploading without format conversion. Usually, if uploaded images satisfy the OSN’s defined size and format, they will be published without resizing or reformatting. If the images are not within the defined constraint, they will be adjusted to a size and format that complies with the OSN’s policies. Since compression, resizing, and format changes will destroy steganographic messages, it is necessary to assess each OSN before making covert communications (see Table 1).

Table 1. Pre-processing Activities

	Facebook	Badoo	Google+
Compressed image	Yes	Yes	No
Resize	Yes	Yes	Only when it's over the size constraint
Format converted	Not on JPEG	Not on JPEG	No
	Others will be converted to JPEG	Others will be converted to JPEG	
Format accepted	JPEG	JPEG	JPEG, BMP, PNG, GIF

If a steganographic image was to be posted on Facebook or Badoo, the only possible carrier would be the JPEG format. However, the newly released Facebook service, called 'file sharing' has given other options to users within a group to share a file of up to 25MB. The terms of service does not permit music or executable files. Consequently, any image file type can be shared via the file sharing feature and the images do not have to go through regular Facebook photo upload pre-processing. The choice of cover-object is not limited and may include BMP, JPEG, GIF, PNG, and even TIFF can be used. Even though music files and .exe files are not permitted, those files can still be transmitted through other channels.

LIVE TESTING

The focus of our research is images and hence the main concern is to investigate the photo sharing capabilities of OSNs. From the pre-testing exploratory phase we chose the two most popular OSNs Facebook and Google+ (as Badoo had the same performances as Facebook) to test steganographic tools and OSN mediation performance. The most commonly used feature is uploading photos via upload Photo/Video or the create Album feature in Facebook. Both features can be found in either a person's home wall or a group wall. Once the photos have been selected (in our case, selecting the photos that have been embedded with secret information) by clicking the post button, the photos will be uploaded to the user's or group's wall. Our tests showed that secret messages were unable to be extracted from the downloaded steganographic images and in particular those images that were created by JP Hide and Seek, StegHide, F5, and SteganPEG. This is due to Facebook's pre-processing compression algorithm that is applied to all uploaded photos regardless of image file size. This is not the case in Google+. One tool, SilentEye, however, had the ability to extract the embedded secret message in images that have gone through the Facebook compression algorithm. SilentEye had the capability to survive the Facebook compression process with minimal distortion and sufficient communication that the human eye could view the embedded image and text.

The other way to share photos in Facebook is through the upload file feature in a group wall. In order to share files, the user has to first create a group with members with whom the user wishes to communicate. The upload file feature is similar to virtual storage where User A is able to upload files onto the group's wall and User B can download it later from the group's wall. For example if Alice created a group named 'Dream' in Facebook and added Bob as a member of this group then Alice and Bob are able to communicate in the 'Dream' group. If Alice has a steganographic image to share with Bob, she can use the upload file feature in the 'Dream' group and upload the image file. To extract the secret message, Bob can download the image file from the 'Dream' group's wall and extract the secret message using the appropriate steganographic tool both Alice and Bob have agreed upon. This way of file sharing successfully exfiltrates the steganographic image and successfully transmits the secret message without having to worry about Facebook's photo compression. With the upload file feature, steganographic images generated by any available image steganography tool can be successfully transmitted in a Facebook social network group either in an open group, closed group or secret group, which is determined by the group's privacy settings. If it is an open group, anyone can see the group, who is in the group, and all the posts or activities of the group. When it is a closed group, anyone can see the group and the members of the group but only members can see the posts or activities. A secret group is only open to its members and only members can see the group, who is in the group, and the content of the group's page.

Sending messages is also a common activity on the Facebook social network and a steganographic image can be sent as an attachment to a message to friends in the network or to the intended recipients using the recipients' email addresses. Likewise, Facebook users can receive messages with steganographic image attachments from friends in their network or receive messages sent to their Facebook email account (e.g. user@Facebook.com) from someone using a traditional email system such as Hotmail, Yahoo Mail or Gmail. For example, Alice sent a message with a steganographic image attachment to Bob, who is a 'friend' in Alice's Facebook. Alice can also send a steganographic image as an attachment to Bob's email address even though Alice and Bob are not 'friends' in Facebook. Furthermore, Bob does not need to have a Facebook account to receive a Facebook message from Alice. Similarly, Bob is able to send steganographic image attachments to Alice's Facebook's email address without having to be Alice's Facebook friend or having a Facebook account. Hence, the file attachment feature is also capable of facilitating steganographic distribution.

The photo sharing feature in Google+ is not as complex as Facebook. Google+ has a basic photo sharing feature which is the 'add photo' function found on the user home page, profile page, or the '+ Share' icon at the top right hand corner of the screen. Users can either instantly upload the photos into a selected circle's page or into a selected album. Unlike Facebook, Google+ does not pre-process the uploaded images with photo compression. If the uploaded images are within the constraints of the uploading policy, the image will be published as it is. Google+ users can either share their photo publicly, which allows everyone who has Google+ to see and download the photos or limit sharing to people who are in the user's 'Circles'. 'Circles' in Google+ are similar to friend lists in Facebook where each category or circle may have different information streams that the users want to share. The 'Circles' can be configured as friends, acquaintances, family and so on. For example, if Alice wanted to share a steganographic image with Bob, Alice can upload the image publicly and Bob will be able to see and download the image from Alice's public profile. On the other hand, Alice can also add Bob to her circles and choose the circle allocated to Bob when uploading the image.

The advantage of disseminating steganographic images in Google+ is that images generated by JP Hide and Seek, S-Tools, StegHide, HIP, GIF-It-Up, F5, SteganPEG, SilentEye and so on, can be directly uploaded with the add photo function in Google+ without any destruction as long as the generated image is in JPEG, BMP, PNG or GIF format and has a resolution of less than 2048 pixel either in height or width. The images will be successfully transported to the intended receiver and the receiver will be able to successfully extract the secret message. SilentEye generates significant artefacts on its stego-object that disclose the use, whereas using other steganographic tools such as JP Hide and Seek, StegHide, F5 and SteganPEG such disclosure is avoided. These tools are able to generate a steganographic image without perceivable artefacts. Additionally, the use of JPEG images is less conspicuous as it is a common format for digital photography. These findings are helpful for an investigator who wishes to be alert to which tool signatures may be present.

			Face book		Google+	
Features	**Tools used**	**Format Used**	**Successful Extraction Secret Message**		**Successful Extraction Secret Message**	
			Yes	**No**	**Yes**	**No**
Photo Upload	JP Hide and Seek	JPEG		√	√	
	Silent Eye*	JPEG		√	√	
	EOF	JPEG		√	√	
	StegHide	BMP		√	√	
	S-Tools	GIF		√	√	
	Invisible Secrets 4	PNG		√	√	
File Sharing	JP Hide and Seek	JPEG	√		Not Applicable	
	Silent Eye	JPEG	√			
	EOF	JPEG	√			
	StegHide	BMP	√			
	S-Tools	GIF	√			
	Invisible Secrets 4	PNG	√			
Message Attachment	JP Hide and Seek	JPEG	√			
	Silent Eye	JPEG	√			
	EOF	JPEG	√			
	StegHide	BMP	√			
	S-Tools	GIF	√			
	Invisible Secrets 4	PNG	√			

Note: * Luminance Interval was set at 5 and JPG quality was configured to 30%

Figure 1. OSNs and Tool Steganographic Capabilities

GUIDELINES FOR INVESTIGATION

The live testing in the Lab showed that conditions apply to secret messaging in images in the two most popular OSNs. A detailed understanding of the site capabilities, policies and rules allows an investigator to narrow searches and to look for audit trails in log files. It can be assumed that if someone wanted to communicate secret messages then evidence can be found in the OSN media. It cannot be assumed that the size of a file is an indicator extra payload is being carried. The number of bits used in each pixel in an image can vary depending on the image format and the number of bits allocated per pixel. In the raster format, the digital true colour image is normally stored in a 24-bit file that derives from the RGB colour scheme. Each primary colour is represented by 8 bits, which means that there are 3 bytes or 24-bits to represent a colour in a pixel and in each pixel there can be 256 quantities of red, green, and blue that can add up to more than 16 million combinations, and therefore can create more than 16 million colours. In addition, the Raster format usually uses lossless compression to decrease the amount of image data that needs be stored. With so many variations and possibilities to 'mix' colours a hidden message payload can have a zero impact on the file size.

Our research therefore eliminates some potential locations for finding hidden messages but did not restrict the possible number of cover objects (that number in the tens of millions in any OSN). Consequently investigators have guidance as to where to look but not on how to look. The standard forensic investigation requires a sweep for steganography and this is usually performed by the use of a standard tool or tools the investigator has customized in repeat use. We found that most tools used in the experiment lacked all the functionality for detection and in several instances we had to use multiple tools and to write our own code. As such most investigators will only have ready access to a cursory glance at any digital evidence containing hidden messages and many important communications can be missed. We tested six steganographic tools namely, JP Hide and Seek, SilentEye, EOF injection, StegHide, S-Tools, and Invisible Secrets 4. These tools have both detection and preparation functions. Hence we have shown that OSN will mediate steganographic content (in the ways described) eliminating some search requirements and artifacts to be examined (for example based on OSN, on file type, conversion policies, luminance and so on). When a suspect's computer becomes available for examination the various audit trails identified in our research can be mapped onto the event in an OSN as confirmatory evidence.

Our work has narrowed down the number of places to look for hidden messages and elaborated a framework for reducing searching based on tool performances, tool signatures and OSN image management policies. The investigator must have extensive knowledge of the OSN preprocessing policies and practice to know what to expect. For example Facebook's photo publishing preprocesses changes the integrity of the uploaded images by allocating its own file name to the uploaded photo. In our case the image uploaded was named as SFB_P2.jpg but renamed as 149889_168496316622410_ 84868167_n.jpg when published. What we cannot tie down is the cost of doing searches for hidden messages. The potentially unlimited number of cover objects (we used image attributes as an example) prevents a 100% positive hit rate and the best research scenario relies on increasing the percentage from its current level. Consequently to answer the question "How much searching is enough?" requires a management judgment that is not only based on a financial cost-benefit analysis but also best practice guidance. Our proposal is to isolate pivotal questions an investigator is to answer as the investigator proceeds through a case. The investigator must assume all digital media can be used for covert communications and proceed by selecting and testing a relevant tool set.

CONCLUSION

The research has answered the question; *What can an investigator do to check for hidden messages in social media?* by eliminating the possibility of some forms of hidden messaging appearing in social media sites. The understanding of how OSNs manage images was thoroughly developed to scope the possible elimination effects. We further developed a flow chart to assure the correct questions are being asked and decisions made in a logical sequence by investigators. However, as always the success of the professional practice will depend upon the effectiveness of the detection tools and the investigator analysis and reporting capability. There is much work yet to be done in the area of tool testing and the development of tools. Tool development is a continuous process where the program for improvement does not stop and the targets for testing continue to evolve. A benchmark for proficiency can be success rate against current signatures but there is no guarantee that success today is to be celebrated tomorrow or an effective tool today is useful tomorrow.

REFERENCES

Ashok, J., Raju, Y., Munishankaraiah, S., & Srinivas, K. (2010). Steganography: An overview. *International Journal of Engineering Science and Technology*, *2*(10), 5985–5992. Retrieved from http://www.ijest.info/docs/IJEST10-02-10-100.pdf

Berg, G., Davidson, I., Duan, M., & Paul, G. (2003). Searching for hidden messages: Automatic detection of steganography. *Proceedings of IAAI 2003*, 51–56. Retrieved from http://www.aaai.org/Papers/IAAI /2003/IAAI03-007.pdf

Castiglione, A., D'Alessio, B., & De Santis, A. (2011). Steganography and secure communication on online social networks and online photo sharing. *2011 International Conference on Broadband and Wireless Computing, Communication and Applications*, 363–368. doi:10.1109/BWCCA.2011.60

Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. *Signal Processing*, *90*(3), 727–752. oi:10.1016/j.sigpro.2009.08.010

Curran, K., & Devitt, J. M. (2008). Image analysis for online dynamic steganography detection. *Computer and Information Science*, *1*(3), 32–41. Retrieved from http://ccsenet.org/journal/index.php /cis/article/viewFile /1825/1735.

Dunbar, B. (2002). A detailed look at steganographic techniques and their use in an open-systems environment. *SANS Information Security Reading Room*. Retrieved March 12, 2013 from http://www.sans.org /reading_room/whitepapers /covert/detailed-steganographic-techniques-open-systems-environment_677

Fridrich, J. (2010). *Steganography in digital media*. Cambridge, UK: Cambridge University Press.

Hamid, N., Yahya, A., Ahmad, R. B., & Al-Qershi, O. M. (2012). Image steganography techniques: An overview. *International Journal of Computer Science and Security (IJCSS)*, *6*(3), 168–187. Retrieved from http://www.cscjournals.org/csc/manuscript/Journals /IJCSS/volume6/Issue3/IJCSS-670.pdf

Hayati, P., Potdar, V., & Chang, E. (2007). *A survey of steganographic and steganalytic tools for the digital forensic investigator*. Retrieved from http://www.pedramhayati.com/images/docs /survey_of_steganography_and _steganalytic_tools.pdf

Hosmer, C. (2006). Discovering hidden evidence. *Journal of Digital Forensic Practice*, *1*(1), 47–56. doi:10.1080/15567280500541447

Kipper, G. (2004). *Investigator's guide to steganography*. Boca Raton, Florida: CRC Press LLC.

Potdar, V. M., Khan, M. A., Chang, E., Ulieru, M., & Worthington, P. R. (2005). e-Forensics steganography system for secret information retrieval. *Advanced Engineering Informatics*, *19*(3), 235–241. doi:10.1016/j.aei.2005.04.003

Zax, R., & Adelstein, F. (2009). FAUST: Forensic artefacts of uninstalled steganography tools. *Digital Investigation*, *6*(1-2), 25–38. doi:10.1016/j.diin.2009.02.002

IDENTIFYING BUGS IN DIGITAL FORENSIC TOOLS

Brian Cusack[1,2], Alain Homewood[1]
[1]Auckland University of Technology, Auckland, New Zealand
[2]Security Research Institute, Edith Cowan University, Perth, Australia
brian.cusack@aut.ac.nz, alain.homewood@aut.a.nz

Abstract
Bugs can be found in all code and the consequences are usually managed through up-grade releases, patches, and restarting operating systems and applications. However, in mission critical systems complete fall over systems are built to assure service continuity. In our research we asked the question, what are the professional risks of bugs in digital forensic tools? Our investigation reviewed three high use professional proprietary digital forensic tools, one in which we identified six bugs and evaluated these bug in terms of potential impacts on an investigator's work. The findings show that yes major brand name digital forensic tools have software bugs and there is room for improvement. These bugs had potential to frustrate an investigator, to cost time, to lose evidence and to require compensatory strategies. Such software bugs also have the potential for malicious exploitation and anti-forensic use.

Keywords
Bugs, Risk, Digital, Forensic, Tools, Work

INTRODUCTION

A software bug is weakness in a computer program either by code or design that produces an incorrect or unexpected result, or causes it to behave in unintended ways (Garfinkel, 2007). The research question regards the value of these vulnerabilities for anti-forensic hacks or the implications for the preservation and presentation of evidence (Hilley, 2007). Exploiting software bugs can occur in many ways. The focus of our interest was fuzzing exploitations. Fuzzing is the process of providing intentionally invalid data to an application in an attempt to trigger an error or fault condition of some kind. This type of activity can be classified as anti-forensic as the consequences can block evidence, counterfeit evidence, confound investigation, frustrate processes, and confuse analysis. Code execution is an integral part of software tool functionality and the associated vulnerabilities require securing. We used fuzzing to create malformed data structures through methods such as randomly replacing single bytes. In its simplest form fuzzing can consist of simply randomly replacing bytes in a data structure; at its most advanced it requires manipulating specific byte locations with knowledge of the properties of a data structure. We used a set of mutations that are designed to exploit typical programming mistakes commonly found in software. An example of one of these mutations is replacing a sequence of NUL bytes with random values of the same length. Fuzzing was performed on a number of file formats such as JPEG images and PDF documents with the goal of detecting problems with the built in file viewers in the forensic tools. Fuzzing was also performed on file system structures in an attempt to reveal issues with the methods used by forensic tools to interpret file systems (Sutton, Green, & Amini, 2001; Harris, 2006).

A second technique used was manual targeted manipulation of data formats. Targeted manipulation is the process of modifying specific portions of a data structure guided with detailed knowledge about the data structure. Two data structures were targeted for testing; individual files and file system structures. Individual files were targeted in an attempt to again locate issues with a tool built in file viewer. File systems and entire disk images were also targeted in an attempt to locate issues with the techniques used to analyse file systems. Function based software testing uses standardised and benchmarked input data but fuzzing addresses the residual risk inherent in such testing. Importantly we identified a number of bugs in several different types of tool and this report focuses