The Danger of Data Exfiltration over Social Media Sites

Dan Gunter, University of Louisville; Solomon Sonya, Western International University

Abstract — *The pervasive utilization of social media sites within both the personal and commercial settings introduce new challenges to the information security sector. Modern network security platforms and procedures fall short in the detection and prevention of this emerging threat. Our research demonstrates new considerations network security professionals must consider when assessing and protecting intellectual information against digital theft.*

Index Terms — Social Media Sites, Steganography, Cryptography, Data Exfiltration

I. Introduction

From the early days of ARPANet to the modern day Internet, societies across the world continue to embrace the wide sweeping benefits provided by rapid communication through the Internet. The Internet has grown since its days of infancy to a medium that many in society feel they can't live or operate without. The technological advances of the Internet has developed the concept of establishing a "presence" in cyberspace. By presence, people are able to stay connected to each other in ways that was only romanticized in science fiction books as early as 10 years ago. Social Networking sites has brought hundreds of thousands of people together, connected through very familiar mediums to share thoughts, ideas, and experiences in one place. People can be connected much closer than before all through the exchange of information through these networking media websites. The rise of social media sites where people can share events from their daily lives now holds a prominent place of importance. Companies, not wanting to be accused of censorship, also get the pressure to participate in the social media forums.

The inclusion of social media sites on company networks introduces risks not previously faced. The rapid growth of social media now provides a camouflage medium for Steganographic and Cryptographic attacks. Facebook alone reports 300 million photos uploaded per day. The sheer volume of data leaving a network presents a challenge to network security professionals auditing traffic due to the fact that data exfiltration can take advantage of this legitimate traffic. Modern network security devices fail to catch this traffic due to the ubiquitous characteristics of this traffic. Our project examines one such implementation that takes advantage of this reality.

II. Cryptography and Steganography

Societies all over the world tell stores of Cryptography and Steganography techniques employed over hundreds of years. Cryptography involves the manipulation of some source message or data using a defined algorithm protected by one or more encryption keys. Steganography involves the covert inclusion of a given message inside some other medium. Both Cryptography and Steganography algorithms should protect the message even when routed through adversarial routes. Julius Caesar employed a simple cryptographic shift cipher to secure messages. He would shift each of the letters in his message forward by three characters prior to sending a courier to deliver s secret message to his generals. This technique protected the initial message should the messenger defect or be captured by enemy forces. Many historians credit Johannes Trithemius with the earliest known application of stegranographic techniques in his book *Steganographia* in 1499.

Various techniques exist, and even more steganographic algorithms have been developed which enables one to hide a secret message into a set of carrier files. We employ the Least Significant Bit (LSB)

Steganography algorithm in our proof of concept project to hide data. While impossible to know the initial employment date of this technique, the hacker magazine Phrack published an article by Hacklab titled *Steganography Thumbprinting* on January 26, 1998 that contains a very clear description of the algorithm. LSB involves the embedding of a message within an image by manipulating the least significant, or last bit, of a given byte. A placement algorithm is used to determine the appropriate indices within a carrier file to extract a byte and prep for data inclusion. The source message (to embed within the carrier file) is completely converted into bits and parsed into fragmented arrays ready to be scattered into the image carrier. To embed a message, predetermined bytes are extracted from the carrier, its least significant bit is cleared (i.e. set to 0) and then one bit from the source message is placed at this location. The process continues until the entire source message has been embedded into the carrier image. The reader will note that each byte from the source message requires eight times as many bytes in the carrier message to contain the data we wish to hide. We will discuss techniques to counteract this weakness later. The receiver must take the image file and extract each least significant bit of every byte to reconstruct the original message.

For cryptography, we employed a simple exclusive-or (XOr) cipher. This cipher takes advantage of the mathematical property that you can decrypt an encrypted message through a XOr operation with an agreed upon key through XOr operations between the key and encrypted message. We define the XOr algorithm as a symmetric algorithm due to the fact that the encrypt and decrypt functions utilize the same key. More advanced encryption algorithms can be called asymmetric if the encryption key is different than the decryption key. Further focus on cryptography sits outside the scope of our project but must be considered by information security professionals to adequately counteract sophisticated attackers. We will study weaknesses of the XOr cipher utilized in this proof of concept tool later in this paper.

III. Introducing Our Proof of Concept Project

We named our project SNScat or Social Network Site CAT in reference to the original Netcat tool that provides administrative level socket read/write access and computer system control. A developer named Hobbit developed Netcat around or prior to 1996 as a TCP/UDP read/write utility on UNIX. The oldest documentation found suggests the tool as “simple and versatile, it's like trying to describe everything you can do with your Swiss Army knife.” Our project seeks to show how one can read/write data to profiles on social media sites on one node and pull the data down from another node. We will explore both human driven and automated techniques to accomplish this goal.

SNScat contains four modules that implement key core functions. The graphical user interface (GUI) or command line interface (CLI) module handles input from the user or system to identify all required files and information to complete the entire process. The cryptography module handles encryption and decryption of the data should the steganography module be compromised. The steganography module embeds and exracts data into one or more carrier files. The social network application programming interface (API) module handles the upload and download function from the social media site. With the exception of the cryptography module, all modules must be implemented for our project to work. All modules on the sender node need to be able to perform operations that the modules on the receiver end can reverse. Both sender and receiver must agree on the algorithm and keys used throughout all operations.

The main goal of the GUI/CLI involves collection of data and information from the user in the interactive version of our program or to keep track of tasks under the automated implementation. The GUI/CLI module handles all SNScat tasks regardless of program state from beginning to end of execution of all modules. All execution options pass from the GUI/CLI to each of the other modules. After gathering the data to hide from the system or user, the GUI/CLI executes the cryptography module to encrypt the data

and either store the encrypted message in memory or write it back to the hard drive. The GUI/CLI then calls the steganography module and provides one or more carrier files to hide the encrypted data. After steganography module hides the encrypted data, the GUI/CLI module invokes the social media API module to transmit the carrier files up to the social media site. The GUI/CLI module on the receiver end scans the profile page waiting for updates that contains photos with embedded data. This requires the sender and receiver to agree on when, where and the format of all data posted. When the receiver identifies embedded data, the receiver end carries out the embedding process in reverse order.

IV. Defending Against the Threat

The naive way to defend against the applied cryptography and steganography threat involves blanket white-listing of websites to keep users from visiting all but a small list of websites. We believe this method to be impractical and counterproductive in the business setting as this can severely limit the legitimate sites where users go to conduct business. On the opposite end of the spectrum, one could simply blacklist popular websites and deny users from visiting a small list of the most popular social network sites. We refer to this method as the whack-a-mole approach and also consider this approach to be naive as this approach suffers from high time and money resource demand. In addition, we have designed our algorithms to work on any digital media. Meaning the blocking of social network sites to prevent this threat is ineffective because we can embed data into normal PowerPoint, Word documents and many other files types users are still able to exchange even if you block their access to social networking sites. As with limiting websites, creating simple antivirus signatures based off any of our project modules would not adequately counter this threat as we could write our modules to be polymorphic.

We believe attacking the technical aspect of this concept to be the best method to counteract this threat Stronger versions of cryptography than the XOr algorithm provide a challenge due to the amount of computational time required to break stronger ciphers. The greatest weakness in steganography exists within the carrier image itself. In steganography, the location of each bit is crucial for the algorithm to be able to extract the appropriate bits to recompose the original message. We discussed the need for the embedded bits in the carrier file to remain at the exact same position between the sender and receiver node. Through the application of a watermark to images or documents in and out of your network that touches even one bit within the embedded message, you can disrupt the entire message. This approach can be implemented on a network infrastructure device or on each endpoint as a method to defeat this type of attack.

IV. Conclusion

Technological advances in the Internet have brought the establishment of social networking sites to keep people connected to each other. This connection allows people to establish a presence of one another enabling the rapid exchange of thoughts, ideas, and experiences. Although social networking sites bring the expanse of unity amongst family, friends, and acquaintances, the use of social media sites on corporate networks introduces new risks not previously faced to the organization. A vast amount of data is exchanged between these sites that network forensics is unable to validate all traffic reaching social networking sites. Due to the shear volume of traffic, many organizations do not monitor traffic destined to various networking sites. New vulnerabilities can be exploited using cryptography and steganography to exfiltrate data and establish command and control between nodes on a network and social networking sites.

SNScat has been developed as a new framework that defeats network forensics to create covert channels, exfiltrate data from protected networks, and establish full command and control (and botnets) using cryptographic and steganographic algorithms to securely embed data into carrier images and exchange via

social networking sites. The most important thing we wish our audience to understand is not the fact that a new tool has been developed but rather that this type of vulnerability exists on enterprise networks.

Although it is not a trivial concept to create, more tools like this one will continue to be developed to enable an attacker exclusive privilege to sensitive data and an avenue to extract confidential information via social networking sites. Network Attackers are constantly improving their tactics creating even better Advanced Persistent Threats (APT). Simply blocking access to sites like this is not a winning strategy. Attack sophistication and misdirection will continue to be amplified. Steganography and cryptography can be applied to nearly all forms of data, not only images. Therefore to best defeat these types of attacks, we recommend focusing on the technique.

References

1. "Key Facts" Facebook. http://newsroom.fb.com/content/default.aspx?NewsAreaId=22 Accessed: 13 Jun 12
2. "Netcat 1.10 Documentation" http://nc110.sourceforge.net/ Accessed: 13 Jun 12
3. "Steganography Thumbprinting" *Phrack Magazine.* vol. 8, no. 52, Jan 26 1998.
4. A. Norman, "Cryptography Defined/Brief History," University of Texas, http://www.laits.utexas.edu/~norman/BUS.FOR/course.mat/SSim/history.html, Accessed 13 Jun 12
5. J. Mathai, "History of Computer Cryptography and Secrecy Systems," Fordham University, http://www.dsm.fordham.edu/~mathai/crypto.html, Accessed: 13 Jun 12