ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

A Study on Steganography & Data Hiding

Vivek Kumar

B.SC (HONS.) Student, Dept. of Biotech., PSC, Patna, Patna University, India

ABSTRACT: Steganography is the art of hiding information and an effort to conceal the existence of the embedded information. It serves as a Better way of securing message than cryptography which only conceals the content of the message not the existence of the message. Original message is being hidden within a carrier such that the Changes so occurred in the carrier are not observable. In this paper we will discuss how digital images can be used as a carrier to hide Messages. This paper also analyses the performance of some of the Steganography tools. Steganography is a useful tool that allows covert Transmission of information over and over the communications Channel. Combining secret image with the carrier image gives the hidden image. The hidden image is difficult to detect without Retrieval. This paper introduces steganography by explaining what it is, providing a brief history with illustrations of some methods for implementing steganography, and comparing available software providing steganography services. Though the forms are many, the focus of the software evaluation in this paper is on the use of images in steganography.

KEYWORDS: Steganography, History and Steganography Digital watermarking, and Cryptography, Data Hiding using command prompt.

I. INTRODUCTION

Internet users frequently need to store, send, or receive private information. The most common way to do this is to transform the data into a different form. The resulting data can be understood only by those who know how to return it to its original form. This method of protecting information is known as encryption. A major drawback to encryption is that the existence of data is not hidden. Data that has been encrypted, although unreadable, still exists as data. If given enough time, someone could eventually unencrypt the data. A Solution to this problem is steganography. The ancient art of hiding messages so that they are not detectable. No substitution or permutation was used. The hidden message is plain, but unsuspecting to the reader. Steganography's intent is to hide the existence of the message, while cryptography scrambles a message so that it cannot be understood. Before the invention of digital means, traditional methods were being used for sending or receiving messages. Before phones, before mail messages were sent on foot. For the messages where privacy was of prime concern, the ways of implementing security were following:
1. Choosing the messenger capable of delivering the message
Securely.
2. Write the message using such notations that actual meaning of the
Message was concealed.
3. Hide the message such that even its presence can't be predicted.

In steganography, the possible cover carriers are innocent looking carriers (images, audio, video, text, or some other digitally representative code) which will hold the hidden information. A message is the information hidden and may be plaintext, cipher text, images, or anything that can be embedded into a bit stream. Together the cover carrier and the embedded message create a stego-carrier.Hiding information may require a stego key which is additional secret information, such as a password, required for embedding the information. For example, when a secret message is hidden within a cover image, the resulting product is a stego-image.
Steganography can be viewed as akin to cryptography. Both have been used throughout recorded history as means to protect information. At times these two technologies seem to converge while the objectives of the two differ. Cryptographic techniques "scramble" messages so if intercepted, the messages cannot be understood. Steganography, in an essence, "camouflages" a message to hide its existence and make it seem "invisible" thus concealing the fact that a message is being sent altogether. An encrypted message may draw suspicion while invisible messages will not [JDJ01].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

II. HISTORY AND STEGANOGRAPHY

Throughout history, a multitude of methods and variations have been used to hide information. David Kahn's *The Code breakers* provides an excellent accounting of this history [Kahn67].
Bruce Norman recounts numerous tales of cryptography and steganography during times of war in *Secret Warfare: The Battle of Codes and Ciphers*.
One of the first documents describing steganography is from the Histories of Herodotus. In ancient Greece, text was written on wax covered tablets. In one story Demeratus wanted to notify Sparta that Xerxes intended to invade Greece. To avoid capture, he scraped the wax off of the tablets and wrote a message on the underlying wood. He then covered the tablets with wax again. The tablets appeared to be blank and unused so they passed inspection by sentries without question.
Another ingenious method was to shave the head of a messenger and tattoo a message or image on the messengers head.

After allowing his hair to grow, the message would be undetected until the head was shaved again.
Another common form of invisible writing is through the use of Invisible inks.
Such inks were used with much success as recently as WWII.
An innocent letter may contain a very different message written between the lines [Zim48].
Early in WWII steganography technology consisted almost exclusively of invisible inks [Kahn67]. Common sources for invisible inks are milk, vinegar, fruit juices and urine. All of these darken when heated.
With the improvement of technology and the ease as to the decoding of these invisible inks, more sophisticated inks were developed which react to various chemicals.
Some messages had to be "developed" much as photographs are developed with a number of chemicals in processing labs.
Null ciphers (unencrypted messages) were also used.
The real message is "camouflaged" in an innocent sounding message. Due to the "sound" of many open coded messages, the suspect communications were detected by mail filters. However "innocent" messages were allowed to flow through.
Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, where the "enemy" is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other "harmless" messages in a way that does not allow any "enemy" to even detect that there is a second secret message present [Markus Kuhn 1995-07-03].eC" @ hee_E_Ae._C&jP-hT,eAT_A eq.Pe_._A@#*,-h6~?]`V(UY3A/X?UokiRO_+Yu?DU)>YOTc*\:Mu',...

III. DIGITAL WATERMAKING

Digital watermarking is the process of embedding information into a digital signal in a way that is difficult to remove. The signal may be audio, pictures or video, for example. If the signal is copied, then the information is also carried in the copy. A signal may carry several different watermarks at the same time. Visible watermarking in this, the information is visible in the picture or video. Typically, the information is text or a logo which identifies the owner of the media. When a television broadcaster adds its logo to the corner of transmitted video, this is also a visible watermark. Invisible Watermarking In this, information is added as digital data to audio, picture or video, but it cannot be perceived as such (although it may be possible to detect that some amount of information is hidden). The watermark may be intended for widespread use and is thus made easy to retrieve or it may be a form of Steganography, where a party communicates a secret message embedded in the digital signal. In either case, as invisible watermarking, the objective is to attach ownership or other descriptive information to the signal in a way that is difficult to remove. It is also possible to use hidden embedded information as a means of covert communication between individuals. Digital Watermarking can be used for a wide range of applications such as: Copyright protection Source Tracking (Different recipients get differently watermarked content). The numbers of possible applications for digital watermarking technologies are increasing rapidly. For example, in the field of data security, watermarks may be used for certification, authentication, and conditional access. Certification is an important issue for official documents, such as identity cards or passports. Digital watermarks are created by converting copyright information into apparently random digital "noise" using an algorithm that is imperceptible to all but special watermark reading software. So while a JPEG file that is read by a Web browser may display a pretty picture, that same file will display the copyright when read by the watermark software.

IV. WHITE NOISE STORM

Arachelian encourages encrypting the message before embedding it into an image. *White Noise Storm* (WNS) also includes an encryption routine to "randomize" the bits with in an image. His use of encryption with steganography is well integrated, but is beyond the scope of this paper. WNS was designed based on the idea of spread spectrum technology and frequency hopping. "Instead of having *X* channels of communication which are changed with a fixed formula and passkey. Eight channels are spread within a number of 8-bits**W* byte channels. *W* represents a random sized window of *W* bytes. Each of these eight channels represents one single bit, so each window holds one byte of information and a lot of unused bits. These channels rotate among themselves, for instance bit 1 might be swapped with bit 7, or all the bits may rotate positions at once. These bits change location within the window on the byte level. The rules for this swapping are dictated not only by the passphrase by also by the previous window's random data (similar to DES block encryption)" [Arachelian, RE: Steganography].

V. WNS

WNS also used the Least Significant Bit (LSB) application of steganography and applies this method to PCX8 files. The software extracts the LSBs from the container image and stores them in a file. The message is encrypted and applied to these bits to create a "new" set of LSBs. These are then "injected" into the container image to create a new image. The documentation that accompanies *White Noise Storm* is well organized and explains some of the theory behind the implementation of encryption and steganography.
The main disadvantage of applying the WNS encryption method to steganography is the loss of many bits that can be used to hold information. Relatively large files must be used to hold the same amount of information other methods provide.

VI. S-TOOLS

Steganography Tools (*S-Tools*) for Windows 3.00 by Andy Brown is the most versatile steganography tools of any applications tested. It includes several programs that process GIF and BMP images (ST-BMP.EXE), audio WAV files (ST-WAV.EXE) and will even hide information in the "unused" areas on floppy diskettes (ST-FDD.EXE). In addition to supporting 24-bit images, *S-Tools* also includes a barrage of encryption routines (Idea, MPJ2, DES, 3DES and NSEA) with many options.

VII. S-TOOLS

*S-Tool s*applies the LSB methods discussed before to both images and audio files. Due to the lack of resources, only images were tested. Brown developed a very nice interface with prompts and well developed on-line documentation. The only apparent limitations were the resources available. There were times large 24-bit images would bring the Windows to a halt. A very useful feature is a status line that displays the largest message size that can be store in an open container file. This saved the time of attempting to store a message that is too large for a container. After hiding the message, the "new" image will be displayed and let you toggle between the new and original images. At times the new image looked to be grossly distorted, but after saving the new image looked nearly identical to the original. This

may be due to memory limitations. On occasion a saved image was actually corrupted and could not be read. A saved image should always be reviewed before sending it out.
The following is derived from S-Tools BMP - How it is done by Andy Brown: "S-Tools works by 'spreading' the bit-pattern of the message file to be hidden across the least-significant bits of the color levels in the image. S-Tools tries to reduce the number of image colors in a manner that preserves as much of the image detail as possible. It is difficult to tell the difference between a 256 color image and one reduced to 32." "S-Tools adds some extra information on to the front of the message file before hiding. 32 bits of time-dependent random garbage is added first. This step means that two identical hidden files that are encrypted in CBC or PCBC mode will never encipher to the same ciphertext. The 32 bit length of the hidden file is then included. This is required for S- Tools to be able to extract the hidden file. Encryption will conceal this value."

To further conceal the presence of a file, S-Tools picks its bits from the image based on the output of a random number generator. This is designed to defeat an attacker who might apply a statistical randomness test to the lower bits of the image to determine whether encrypted data is hidden there (well-encrypted data shows up as pure white noise). The random number generator used by S-Tools is based on the output of the MD5 message digest algorithm, and is not easily (if at all) defeatable" [*S-Tools* Documentation by Andy Brown].

SOFTWARE NOT TESTED BUT WORTH NOTHING

The following software packages were reviewed but not tested: Jpeg-Jsteg v4 and Stealth v1.1.

VIII. JPEG-JSTEG v4

Cryptography and steganography rely on retrieving a message in its original form without losing any information. Such is the idea behind lossless compression. Since JPG images use lossy encoding to compress its data, it is generally thought that steganography would be infeasible with such images. "This version of the Independent JPEG Group's JPEG Software has been modified for 1-bit steganography in JFIF output files" [Independent JPEG Group]. The Jpeg-Jsteg software comes with source code and instructions for compiling the code on various platforms.

According to the Independent JPEG Group (IJPG), the JFIF format is composed of lossy and non-lossy stages. Information can be inserted between these stages without corrupting the image.
As discussed earlier with Renoir's *Le Moulin de la Galette* compression is a great advantage JPG images have over other formats. JPEG images are becoming more abundant on the Internet because large images with unlimited colors can be stored in relatively small files (a 1073 x 790 pixel image with 16 million colors can be stored in a 170 Kilobyte file. The same image is over 2 Megabytes if converted to a BMP).

IX. STEALTH

Stealth by Henry Hastur in and of itself is not a steganographic program or method. It is usually found with steganographic software on the Internet and is used to complement the steganographic methods. *Stealth* is a filter that strips off the PGP header that is on a PGP encrypted file. This leaves only the encrypted data. Why is this important? Applying steganography to an encrypted message is more secure than a "plain text" message. However, many encryption applications add header information to the encrypted message. This header information identifies the method used to encrypt the data.
Stealth For example, if a cracker has identified hidden data in an image and has successfully extracted the encrypted message, a header for the encryption method would point the cracker in the right direction for additional cryptanalysis. But, if the header is removed, the cracker cannot determine the method for encryption. Some steganography software (*White Noise Storm* and *S-Tools*) provide this step in security, but others do not.

X. CONCLUSION AND COMMENTS

Steganography has its place in security. It is not intended to replace cryptography but supplement it. Hiding a message with steganography methods reduces the chance of a message being detected. However, if that message is also

encrypted, if discovered, it must also be cracked (yet another layer of protection). There are an infinite number of steganography applications. This paper explores a tiny fraction of the art of steganography. It goes well beyond simply embedding text in an image. Steganography does not only pertain to digital images but also to other media (files such as voice, other text and binaries; other media such as communication channels, the list can go on and on). Consider the following example:
A person has a cassette tape of Pink Floyd's "The Wall." The plans of a Top Secret project (e.g., device, aircraft, covert operation) are embedded, using some steganographic method, on that tape. Since the alterations of the "expected contents" cannot be detected, (especially by human ears and probably not easily so by digital means) these plans can cross borders and trade hands undetected. How do you detect which recording has the message?

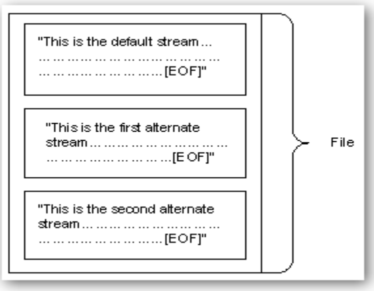
This is a trivial (and incomplete) example, but it goes far beyond simple image encoding in an image with homogeneous regions. Part of secrecy is selecting the proper mechanisms. Consider encoding using an Mandelbrot image [Hastur].

In and of itself, steganography is not a good solution to secrecy, but neither is simple substitution and short block permutation for encryption. But if these methods are combined, you have much stronger encryption routines (methods).

For example (again over simplified): If a message is encrypted using substitution (substituting one alphabet with another), permute the message (shuffle the text) and apply a substitution again, then the encrypted ciphertext is more secure than using only substitution or only permutation. NOW, if the ciphertext is embedded in an [image, video, voice, etc.] it is even more secure. If an encrypted message is intercepted, the interceptor knows the text is an encrypted message. With steganography, the interceptor may not know the object contains a message.

XI. DATA HIDING USING COMMAND PROMPT

Ever since Windows 2000, the NTFS file system in Windows has supported Alternate Data Streams, which allow you to store data "behind" a filename with the use of a stream name. It's not detectable while browsing the file system, or


"This is the default stream ... .[EOF]"
"This is the first alternate stream ....[E OF]"
"This is the second alternate stream ....[EOF]"
File

anywhere within Windows… you can only access it with the "secret key" which is really just the name of the stream.You can think of these extra streams as secret compartments within the file that can only be accessed if you know the "secret code," which in this case is just the name of the stream.This isn't a completely secure way to hide data as we'll illustrate below, but it's a fun trick to know about in a pinch.

Note: This only works on a drive formatted with NTFS.

XII. HIDING DATA IN A SECERET COMPARTMENT

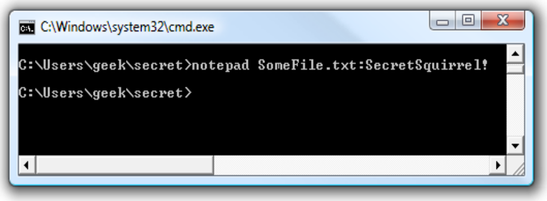
In order to use this feature, you'll have to open a command prompt and use the following syntax:notepad SomeFile.txt:SecretWordHere.txtYou can use anything after the colon as a secret word; the key is that there can't be any spaces between the first filename and the colon.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

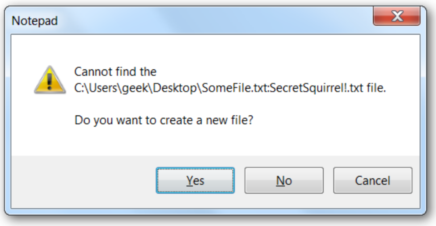
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015



```
C:\Users\geek\secret>notepad SomeFile.txt:SecretSquirrel!

C:\Users\geek\secret>
```

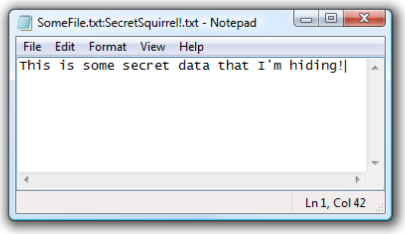
If you didn't specify .txt on the end, Notepad will automatically add it, and ask if you want to create a new file, even if SomeFile.txt already existed, because SecretSquirrel!.txt doesn't already exist.



Cannot find the C:\Users\geek\Desktop\SomeFile.txt:SecretSquirrel!.txt file.

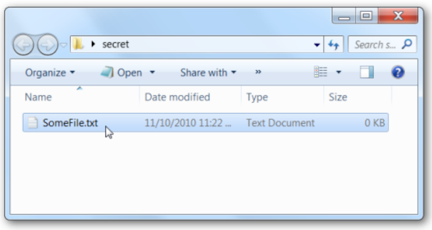
Do you want to create a new file?

Now you can enter in whatever data you want here and save the file:


SomeFile.txt:SecretSquirrel!.txt - Notepad
File Edit Format View Help
Ln 1, Col 42

```
This is some secret data that I'm hiding!
```

When you look at the file, it will still be the exact same size as before:



Name	Date modified	Type	Size
SomeFile.txt	11/10/2010 11:22 ...	Text Document	0 KB

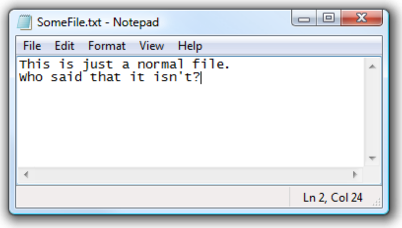
You can even open up the file by double-clicking on it, and add whatever data you want to make the file look normal:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

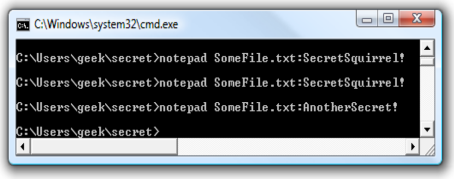
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015



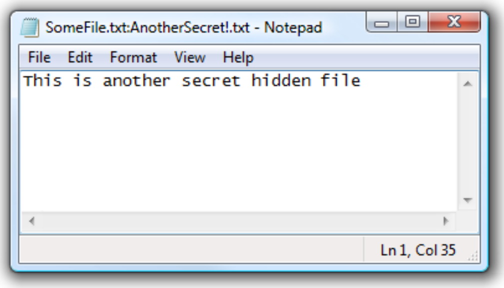
```
This is just a normal file.
Who said that it isn't?
```

You can use the command line again to add a second hidden "compartment" with a different name:



```
C:\Users\geek\secret>notepad SomeFile.txt:SecretSquirrel!
C:\Users\geek\secret>notepad SomeFile.txt:SecretSquirrel!
C:\Users\geek\secret>notepad SomeFile.txt:AnotherSecret!
C:\Users\geek\secret>
```

You can add whatever other information to this file that you'd like:


SomeFile.txt:AnotherSecret!.txt - Notepad
File Edit Format View Help
Ln 1, Col 35

```
This is another secret hidden file
```

None of these hidden files will affect the other, or change the main file. Just remember, you have to use the command line to access the hidden data.

Note: Once you create a hidden stream, that stream isn't exactly part of the file… you can't copy your file to another location and access the streams over there.

DETECTING FILES WITH STREAMS

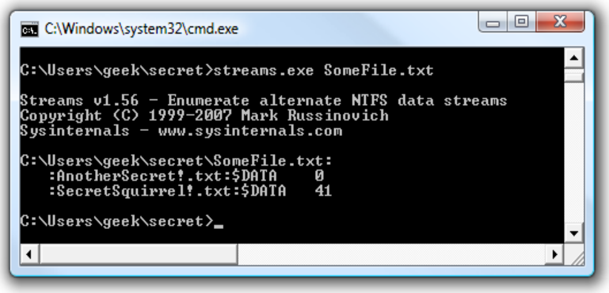
Of course these files aren't completely hidden from everybody, because you can use a small command line application called Streams.exe to detect files that have streams, including the names of the streams. For instance, in my scenario we'd use the following syntax:streams.exe SomeFile.txt

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

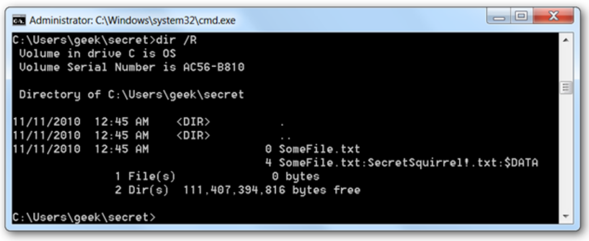
(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015



```
C:\Users\geek\secret>streams.exe SomeFile.txt

Streams v1.56 - Enumerate alternate NTFS data streams
Copyright (C) 1999-2007 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Users\geek\secret\SomeFile.txt:
   :AnotherSecret!.txt:$DATA     0
   :SecretSquirrel!.txt:$DATA    41

C:\Users\geek\secret>_
```

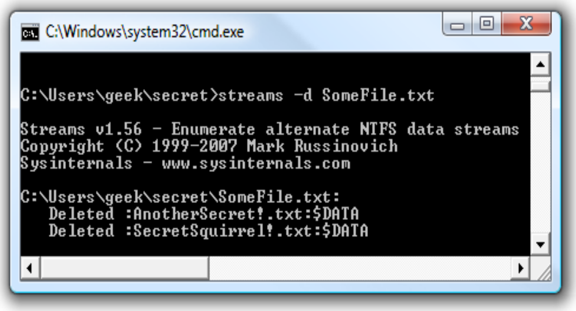
As you can see, the names of the streams are shown, which would allow you to easily access them. If you're using Windows 7, you can simply use the /R argument to the DIR command to see the streams:



```
C:\Users\geek\secret>dir /R
 Volume in drive C is OS
 Volume Serial Number is AC56-B810

 Directory of C:\Users\geek\secret

11/11/2010  12:45 AM    <DIR>          .
11/11/2010  12:45 AM    <DIR>          ..
11/11/2010  12:45 AM                 0 SomeFile.txt
                                     4 SomeFile.txt:SecretSquirrel!.txt:$DATA
               1 File(s)              0 bytes
               2 Dir(s)  111,407,394,816 bytes free

C:\Users\geek\secret>
```

DELETING STREAMS

You can use the same Streams.exe command to delete all streams from a file, although I don't think you can delete just a single stream. Use the following syntax: streams.exe -d SomeFile.txt



```
C:\Users\geek\secret>streams -d SomeFile.txt

Streams v1.56 - Enumerate alternate NTFS data streams
Copyright (C) 1999-2007 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Users\geek\secret\SomeFile.txt:
   Deleted :AnotherSecret!.txt:$DATA
   Deleted :SecretSquirrel!.txt:$DATA
```

As you can see in the screenshot, the streams are now removed from the file.

ADDING TO HIDDEN STREAMS FROM THE COMMAND LINE

You can add data to a hidden stream by using a number of commands, or really anything that can pipe input or output and accept the standard FileName:StreamName syntax. For instance, we could use the echo command:

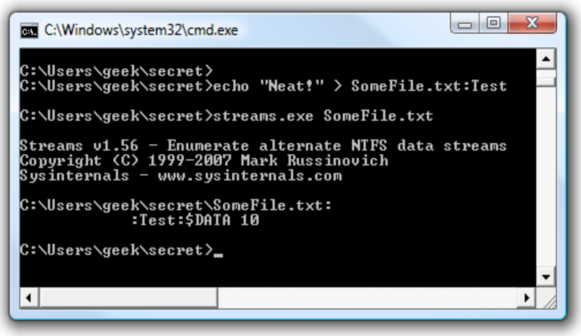
```
echo "Neat!" >SomeFile.txt:Test
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

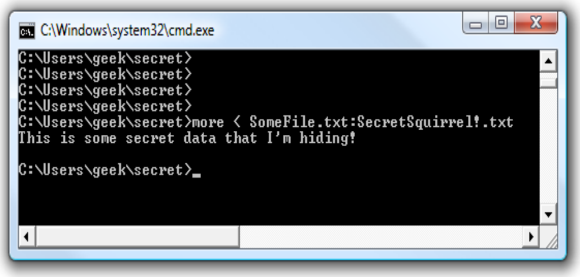


```
C:\Users\geek\secret>
C:\Users\geek\secret>echo "Neat!" > SomeFile.txt:Test

C:\Users\geek\secret>streams.exe SomeFile.txt

Streams v1.56 - Enumerate alternate NTFS data streams
Copyright (C) 1999-2007 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Users\geek\secret\SomeFile.txt:
              :Test:$DATA 10

C:\Users\geek\secret>_
```

You can see with the streams command in the example above that we now have a hidden stream on the file.

READING A STREAM FROM THE COMMAND LINE

You can read data from the stream by piping data into the more command, using this syntax:more <FileName:StreamNameIn my example the actual command was this:more < SomeFile.txt:SecretSquirrel!.txt



```
C:\Users\geek\secret>
C:\Users\geek\secret>
C:\Users\geek\secret>
C:\Users\geek\secret>
C:\Users\geek\secret>more < SomeFile.txt:SecretSquirrel!.txt
This is some secret data that I'm hiding!

C:\Users\geek\secret>_
```

XIII. CONCLUSION

It is crucial to understand how data could be hidden inside file system structure, this knowledge will allow us to understand the potential risks and how hackers could use such techniques to hide malicious files and exploit our system. I have illustrated here using command prompt but various methods are also available.

REFERENCES

1. http://deadlock.hut.fi/ ste/ ste_html.html, ftp://saturn.hut.fi/ pub/ aaura/ ste1195.ps
2. [Brassil-Infocom95] J. Brassil, S. Low, N. Maxemchuk, L. O'Goram,
3. "Document Marking and Identification using Both Line and Word Shifting," *Infocom95*, ftp://ftp.research.att.com/ dist/ brassil/ 1995/ infocom95.ps.Z
4. [Brassil-Infocom94] J. Brassil, S. Low, N. Maxemchuk, L. O'Goram,
5. "Electronic Marking and Identification Techniques to Discourage Document Copying," *Infocom94*, ftp://ftp.research.att.com/ dist/ brassil/ 1994/ infocom94a.ps.Z.
6. [Brassil-CISS95] J. Brassil, S. Low, N. Maxemchuk, L. O'Goram,
7. "Hiding Information in Document Images," *CISS95*, ftp://ftp.research.att.com/ dist/ brassil/ 1995/ ciss95.ps.Z.
8. [JDJ01] Neil F. Johnson, ZoranDuric, SushilJajodia, *Information Hiding: Steganography and Watermarking - Attacks and Countermeasures*
9. Kluwer Academic Press, Norwrll, MA, New York, The Hague, London, 2000.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2015

BIOGRAPHY

Vivekkumar is a B.sc (Hons.) student of Patna Science college department of Biotechnology, Patna University, India. He is undergraduate science student but his current research interests include Genomics and cyber security. I have done cyber security courses.