Data Hiding in Identification and Offset IP fields

Enrique Cauich[1], Roberto Gómez [2], Ryouske Watanabe[2]

[1] California University at Irwing, Computer Science and Engineering 204B
University of California, Irvine, CA 92717
USA
ecauichz@ics.uci.edu
[2] ITESM-CEM, Depto. Ciencias Computacionales, Km 3.5 Lago Guadalupe,
51296, Atizapan Zaragoza, Edo México, Mexico
{rogomez, A00445577}@itesm.mx

Abstract. . Steganography is defined as the art and science of hiding information, it takes one piece of information and hides it within another. The piece more used to hide information are the digital images. In this paper we present a way to use unused fields in the IP header of TCP/IP packets in order to send information between to nodes over Internet.

1 Introduction

Steganography literally means “covered languages” [1,2]. In today’s computer world, it has come to mean hiding secret messages in any digital multimedia signals. Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, HTML, or even floppy disks) with bits of different, invisible information. This hidden information can be plain text, cipher text, or even images. Must of the scientific word is focus in hiding information into images. The techniques used have the intention to make impossible to detect that there is anything inside the innocent file, but the recipient must obtain the hidden data without any problem. The most important feature of a steganographic system is the fact that it allows communication between two authorized parties without an observer is aware that the communication is actually taking place.

TCP/IP is the protocol used in Internet. TCP /IP were developed by a Department of Defense (DOD) research project to connect a number different networks designed by different vendors into a network of networks (the "Internet"). IP (Internet Protocol, [3]) is responsible for moving packet of data from node to node, and TCP (Transmission Control Protocol, [4]) is responsible for verifying the correct delivery of data from client to server. The IP protocol defines the basic unit of data transfer through the Internet as a packet. All the data is partitioned into IP packets on the sending computer and reassembled on the receiving computer. Each packet begins with a header containing addressing and system control information. The header packet is divided into The IP packet header consists of 20 bytes of data divided in several fields. Each

field has a special purpose, depending on the type of data contained in the packet payload.

Many scientific work has been made in order to create software and methods to hide information into digital images. Our approach take advantage of the unused fields of the IP header packet. As mentioned earlier we not all the fields of an IP packet are always used. These fields are used to hide the information we want to send without raising any suspicion.

This paper is organized as follows: In section two we present an analysis of steganographic methods. This is followed by an overview of the Internet Protocol in section three. Previous work, that uses a similar approach than us, is analyzed in section four. Our proposal is explained in section five and the implementation and experiments are showed is section six. The last section presents our conclusions, limitations and advantages of our work.

2 Steganography overview

Communication confidentiality can be accomplish using cryptography, which involves key administration, algorithm implementation and other management issues. Nevertheless, if an eavesdropper is listening he will realize that exists a secret communication between two entities. Steganography will hide the presence of a message in such a way that an eavesdropper (who listen to all the communications) cannot tell that a secret message is being sent. As the goal of steganography is to hide the presence of a message, it has been as the complement of cryptography, whose goal is to hide the content of a message.

The first scientific study of steganography was presented by Simmons in 1983 [5] who formulate it as the “Prisoners problem”. The problem is the following one: Two prisoners need to communicate, but all the messages pass through the warden who can detect any encrypted messages. They must find some technique of hiding their message in an innocent looking communication.

The generic embedding and decoding process in steganography is presented in [6,7,8] The first step in embedding and hiding information is to pass both, the secret message and the cover message, into the encoder. Inside the encoder, one or several protocols will be implemented to embed the secret information into the cover message. A key is often needed in the embedding process. This can be in the form of a public or private key. Having passed through the encoder, a stego object will be produced. A stego object is the original cover object with the secret information embedded inside. it will then be sent off via some communications channel, such as email, to the intended recipient for decoding. The recipient must decode the stego object in order for them to view the secret information. The decoding process is simply the reverse of the encoding process. After the decoding process is completed, the secret information embedded in the stego object can then be extracted and viewed.

The most used cover messages are digital images. In [9] Nelson and Jajodia gives an introduction to steganography in digital images. According to them must of the

techniques use common approaches that includes least significant bit insertion, masking and filtering and transformations. The LSB method works by using the least significant bits of each pixel in one image to hide the most significant bits of another. Masking and filtering techniques hide information by marking an image, in a manner similar to paper watermarks. Transformation take advantages of algorithms and coefficients form processing the image or its components to hide information. One example of this technique is the discrete cosine transformation. In [10] the authors uses digital imagery as a cover signal to hide information. In [11] the authors propose to use random bit-sequences generated by linear shift registers (LFSRs) within the pixel-byte instead of just the LSB. They established that such changes within any given pixel of the image will result in better hiding of the data and hence secure data transmission.

Other covert messages include audio signals or slack space in disks. In [12] propose a technique that uses autocorrelation modulation, with several variations, to hide information within audio-signals. A MP3 resistant oblivious data hiding technique is presented in [13].

Like many security tools, steganography can be used for a variety of reasons, some good, some not so good. Legitimate purposes can include things like watermarking images for reasons such as copyright protection. Digital watermarks (also known as fingerprinting, significant especially in copyrighting material) are similar to steganography in that they are overlaid in files, which appear to be part of the original file and are thus not easily detectable by the average person.

Attacks on steganographic systems exists and are named steganalysis. Their goal is to determine whether or not they have a payload encoded into them, and, if possible, recover that payload. More information can be found in [14,15,16]. An interesting analysis of limits of steganography is presented in [17]; the authors presents a discussion of the obstacles that lie in the way of a general theory of information hiding systems.

3 The Internet Protocol

Internet use the Internet Protocol (IP) as a standard way to transmit information and actually almost al the network is based in the IP version 4. The header of this protocol usually uses some fields that have some redundancy or normally are not used during the transmissions. We can use this fields that are not used for our purposes, but first we will analyze how the IP header works. For the aim of our investigation we will focus in just the second and third 32-bit worlds of the header; it mean's; the identification, flags, fragment offset, Time to Live, protocol and Checksum fields of the header.

Vers	HLen	TOS	Total Length	
Identification			Flags	Fragment Offset
Time to Live		Proto-col	Header Checksum	
Source Address				
Destination Address				
Options				Padding

Fig. 1. Fields of an IP header

When the transmission over the internet occurs; the information is wrapped by different protocols at different layers of the TCP/IP network model. Two of these layers are the Physical layer and the Transport layer. The communication over the transport layer is standardized by the IP protocol, but over the network layer exists some different technologies and implementations, which implies that each technology has a maximum size of data it can carry per transmission or Maximum Transfer Unit(MTU).

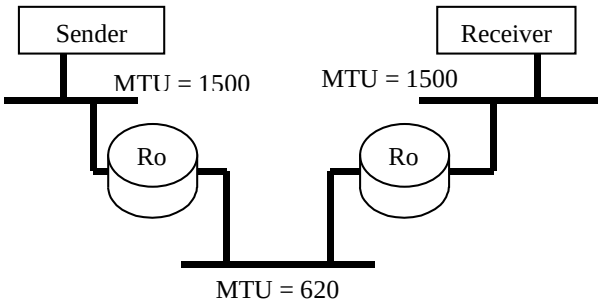

Fig. 2. MTU example

The Transport layer solve this problem with the fields located at the second and third 32-bit words. During the transmission of an IP packet, if the MTU of the source network is smaller there is not problem; but if not, the router needs to fragment the IP packet.

When fragmentation occurs; the router splits the IP datagram with a maximum size of the new MTU. The new headers has the same information, but now the bit of More Fragments is turned on and the Fragment Offset indicates the offset of the data. Otherwise, if fragmentation does not occur; both fields, the Flags Field and the Fragment Offset, are set in zero.

Finally when the packet arrive to its destination, this device must be able to join the packet again; therefore, IP needs to assure when the device joins the pieces again; that

each one corresponds to the original packet. To assure it, IP uses the Identification field.

There are other modification that occurs over the IP header every time that a packet pass trough a router. When a packet reach to a router, the field of Time to Live in the header decreases its value, originally set in 30, by one. If a packet reaches the router with a value of zero in the Time to Live field, the packet is dropped; this is because IP need to assure that a packet will no be forever traveling over the network without reaching its destiny. At least, because of all the modifications that occur in the IP header while traveling, the Checksum field is modified every time the packet reaches a router.

4 Previous work of steganography in IP

Our proposal is to use some of the fields described in the previous section. Similar approaches have been published in [15] and [16].

In [15] the authors idea resides in the manipulation of the IP Identification Field. The Identification Field of the IP Packet is assigned by the original sender. This number consist basically in a random number generated while the packet was being constructed. The Identification Field is only used when fragmentation occurs. Therefore; if we assure that no fragmentation will occur because of the size of the packet; it is possible to hide data in this field without any consequence in the transmission.

The advantages in this work is that it is used to send information from point to point, but the limitations are the quantity of information that you send. Furthermore if by any circumstances the datagram is fragmented, the receiver will listen noise in the transmission because it will receive the same information more than one time with every new fragment of the datagram.

In [16] the work is focused in the manipulation of the Do Not Fragment Bit. There is possible to indicate if we do not want that our packet be fragment by the routers in the way. In consequence; again, if we assure that our packet will be not fragmented because of the size of it; we can hide information in the Do not fragment Bit at the flags field.

In this work the problem of the size of data is worst than the Identification Field, because here we can only transmit one bit for each datagram. Imagining that the datagram does not carry any data but the header, then the ratio useful information to total data is 1:160, it means that if you want to transmit the phrase “hello world” you will need to transmit 88 datagrams producing and overhead of almost 2 Kb for just 11 bytes.

5 Our proposal

Our idea is not really to hide information, but to use the non-used bits to send messages and information, node to node, related with the router performance, best routes or even to update the new routes between the gateways without generating more traf-

fic. For this purpose we will analyze two fields that are not quite often used in a IP datagram transmission.

As it was mentioned, the Fragmentation Offset Field always is set to zero if fragmentation does not occur at all and the Identification Field also becomes useless if there it occurs. Unfortunately, we cannot be sure of it because we are not sure that the source MTU is the smaller in the travel that the packet will take; furthermore, we are not sure of which path the packet will take during the travel. In consequence we can not use the Fragmentation Offset Field or the Identification Field if we want to transmit point to point information, but in node to node.

5.1 Packet fragmentation

There are two scenarios when a IP packet cross from one network to other. The first one is that it is fragmented because the MTU of the second one is smaller the former, in which case the More Fragments bit of the Flags Field is set to 1 and the Fragment Offset became used. The second scenario is when fragmentation is not necessary; therefore, the Fragment Offset will be zero, and the only two modifications that the datagram will receive is the decrement of the TTL field and a recalculation of the Header Checksum.

It is in the second scenario when it is possible to substitute the fragment offset by some data without any consequence.

5.2 Datagram selection to carry information

Now the problem is how we can identify when a datagram is loading information of fragmentation in the Offset field or when the datagram is loading our information. IT. This not possible to know only with the More Fragment bit, because it is set to one in every fragment except the last one. Therefore, in the last fragment we will have a More Fragment bit in zero and nonzero Fragment Offset. Furthermore we cannot track if the datagram is part of a fragmented one or not because maybe every fragment can take a different path. Moreover, ff this could be possible we will require to store the ID of the fragment with the destination IP address.

The solution to this problem is to use a non used bit, than can be every reserved bit of the header that actually is not used. It can be the two less significant bits of the TOS field or the most significant bit of the Flags field. For convenience we will use the bit of the Flags field, because its only necessary to do an “AND” in a 32 bit length word to extract if the datagram carries our information. Another advantage of this approach if that we already have extracted the information. We have also to discern when a datagram can use a datagram; it can be used only under two circumstances. The first one is when the datagram has the chosen reversed bit on, meaning the datagram carries information of the previous gateway. The second case occurs when the More Fragments Bit is off and also the Fragment Offset is set to zero that indicates the datagram has not been fragmented.

After the gateway extracts the information we embedded in the datagram, these fields are replaced with a random value in the Identification Field and set to zero in the Offset Field if there is no information we need to transmit to the next router, or with the new information in the other case.

6 Implementation and tests

The code that implements our proposal uses the LibNet library for the construction of the packets two computer Pentium running Open BSD 2.x.

Our environment test was constituted by two computers Pentium running OpenBSD 2.x that work as the gateways. One of them (R1) was running the program that injects the information within the datagram. The principal roll of this program was to read from the internal interface the datagram, check if it has fragmentation. If not if there was some information to send to R2, it sets the reserved bit from the flags field to one and write down the information in the ID and Offset Field. After the decrement of the TTL and the recalculation of the checksum the datagram is sent through the external interface to the next network.

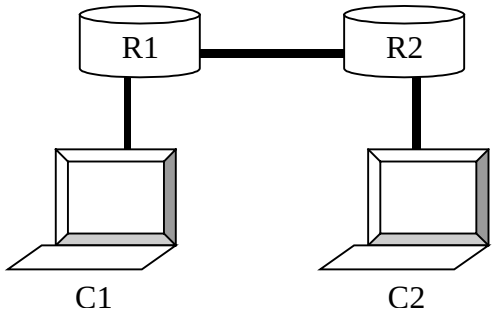

Fig. 3. Implementation architecture

The information that R1 sent was write down in a text file in R1. And it was sent while there exists traffic in the network from C1 to C2 and the EOF of the text file was no reached.

The second gateway (R2) was running the program that take out the information and reestablish the packet. This program read the flags field of the datagram, if it has the reserved bit on and the fragmentation bit of then it takes out the information of the ID and fragment Offset field and reestablishes the value of the Offset field to zero, copies the Checksum to the ID field (this because we need any number in this field), decreases the TTL and recalculates the Checksum. The information that R2 received was displayed to the screen of R2.

C1 and C2 was to laptops running Windows 2000 that was doing ping and telnet. Also both of them was running ethereal to check the structure and the data that the datagrams were carrying. Additionally we connected and sniffer between R1 and R2

in order to maintain a tracking of the packets and the information that they were carrying.

7 Conclusions

We have presented a new another technique to hide information over a valid communication channel. The covert messages were the Identification and Offset IP fields of the TCP/IP packets used in a communication between two valid entities. The experiments shown some limitations but they also presented some advantages over similar steganographic techniques.

As we mention above; it is not possible to send information point to point because we cannot assure that the IP datagram will be not fragmented. Furthermore we do not know exactly which way the packet will take, so is not possible to be sure in our information will arrive to destiny or the datagram will take another way that never pass thought our destination. That is caused because the owner of the datagram is not ours. Actually there are two ways to assure the packet will cross thought a gateway.

The first way when we route the datagram thought a known interface with a known MAC address of a known gateway at the same network segment.

The second way is to put a static rout in the Options field of the datagram, but with this we are doing and extra work that causes overloading at the gateway and also and overheading at the network that is what we try to avoid.

Another limitation is that in presence of an Intrusion Detection System (IDS), and depending the configuration of, it is possible that the datagram can be identified as a malicious one.

By the other hand, our work presents several advantages. The first advantage is that we have an effective 12-bit word to be used in every datagram that is not fragmented, and the only extra work that the gateway need to do is to replace the data carried at the Offset Field with zeros. There is no more work for the gateway because finally every time a datagram cross thought a gateway the TTL field is decreased and the checksum must be recalculated.

Furthermore, with the use of this methodology to send information between to gateways that are back to back, they can share routes, the load of each route or the quality of service, throughput of each route without generating overloading in the network.

References

1. D Kanh, The History of Steganography, Proceedings: Information Hiding. First International Workshop, Cambridge UK pp1-5 1996
2. W Bender et al. Techniques for Data Hiding IBM Systems Journal Vol 35 Nos. 3&4, pp 313-336, 1996 15. Craig H. Rowland, Covert Channels in the TCP/IP protocol suite. First Monday, 2003.

3. RFC 793, Transmission Control Protocol, Darpa Internet Program, Protocol Specification, September 1981
4. RFC 791 Internet Protocol, Darpa Internet Program, Protocol Specification, September 1981
5. Simmons, G. J. Prisoners' problem and the subliminal channel., Advances in Cryptology: Proceedings of CRYPTO 83. D. Chaum, ed. Plenum, New York, 1983, pp. 51-67.
6. Neil F. Johnson and Sushil Jajodia, Steganalysis of Images Created Using Current Steganography Software, Proceedings Second Information Hiding Workshop held in Portland, Oregon, USA, April 15-17, 1998, Lecture Notes in Computer Science, Vol. 1525, pp 273-289
7. L.M. Marvel, C.T. Retter, C.G. Boncelet, Jr, Hiding information in images, 1998 International Conference on Image Processing (ICIP '98) 3-Volume Set-Volume 2 October 04 - 07, 1998 Chicago, Illinois
8. Neil F. Johnson and Sushil Jajodia, Steganalysis: The Investigation of Hidden Information,. IEEE Information Technology Conference , Syracuse, New York , USA, September 1st - 3rd, 1998: 113-116.
9. Neil F. Johnson, Sushil Jajodia, Exploring Steganography: Seeing the Unseen, IEEE Computer, February 1998 (Vol. 31, No. 2) pp 26-34
10. L. M. Marvel, C. G. Jr. Boncelet and C. T. Retter, "Spread spectrum image steganography", IEEE Transactions on Image Processing, Volume: 8, August 1999
11. Jamil, T.; Ahmad, A.; An Investigation into the application of Linear Feedback Shift Registers for Steganography SoutheastCon, 2002. Proceedings IEEE , 5-7 April 2002 pp 239 - 244
12. Petrovic, R.; Winograd, J.M.; Jemili, K.; Metois, E.; Data hiding within audio signals Telecommunications in Modern Satellite, Cable and Broadcasting Services, 1999. 4th International Conference on ,Volume: 1 , 13-15 Oct. 1999
13. Litao Gang; Akansu, A.N.; Ramkumar, M.; MP3 resistant oblivious steganography Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on , Volume: 3 , 7-11 May 2001, pp:1365 - 1368 vol.3
14. Huaiqing Wang, Shuozhong Wang; Cyber warfare: steganography vs. steganalysis, Communications of the ACM archive, Volume 47 , Issue 10 (October 2004) pp. 76 - 82
15. Jessica Fridrich and Miroslav Goljan, Practical steganalysis of digital images: state of the art, Proceedings of SPIE -- Volume 4675 Security and Watermarking of Multimedia Contents IV, April 2002, pp. 1-13
16. Chandramouli,R.; Subbalakshmi, K.P., Active steganalysis of spread spectrum image steganography Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on , Volume: 3, May 25-28, 2003
17. Anderson, R.J. Petitcolas, F.A.P., On the limits of steganography Comput. Lab., IEEE Journal on Communications, May 1998, Volume: 16, Issue: 4 pp. 474-481
18. Rowland, Craig H, Covert channels in the TCP/IP protocol suite, DoIS Documents in Information Science, May 1997
19. K. Ahsan and D. Kundur, Practical data hiding in TCP/IP, Proc. ACM Workshop on Multimedia Security, 2002.