IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology
ISO 3297:2007 Certified
Vol. 5, Issue 6, June 2018

Steganographic System for Hiding Information in TCP/IP Packets

Erika R. Llanes Castro[1], Lizzie E. Narváez Díaz[2], Victor M. Chi Pech[3]

Faculty of Mathematics, Autonomous University, Merida, Yucatán, Mexico [1-3]

Abstract: In this article we present a steganographic system that permits the sending, reception and recuperation of secret data hidden in the TCP/IP packet header fields that are often unused or that exhibit some intrinsic redundancy to said protocols. The Mukul (meaning secret or hidden in the Mayan language) steganographic system is developed using freeware: the Libpcap and Libnet function libraries for programming the routines in C, and the Q library for the graphic interface. We also present the protocol implemented for the testing of the system and the obtained results.

Keywords: Steganography, Hidden information, TCP/IP network packets, Libpcap, Libnet.

I. INTRODUCTION

In an ideal information system, each of the components would only receive the pertinent information; and this information would remain unknown to the rest of the elements that do not require it [1].

There are two ways of stopping the information from being known by those for whom it is not pertinent; coding it or hiding it. [2]. In the first case, the sent message is transformed in such a way that, although others can see that there is a message, they cannot interpret the meaning of the message. The branch of science that studies the ways in which messages can be transformed into something inaccessible for all bar the intended recipient is called cryptography. In the second case, the sent message is hidden in such a way that only the recipient knows that the message exists. The branch of science that deals with the hiding of information is called steganography.

II. STEGANOGRAPHY

Steganography allows extra information to be hidden along with a message, information that only the recipient knows is present. An example of steganography are microdots, photographs reduced to the size of an orthographic character that can be adhered to a letter without the reader knowing that the punctuation marks contain information not included in the main text of the letter [3]. In this case the information would effectively be hidden.

From a more formal perspective, steganography takes a host message and modifies it to find another different message with the same meaning. This process of modification is done by means of a secret message that remains hidden, whereby only those who know the process followed during concealment can recover it satisfactorily. Depending on the nature of the host message (ASCII text, a JPEG image, an MP3 audio clip, etc.), the concept of meaning would change radically, and likewise the modification processes that allow for the hosting of the secret message without arousing suspicion [3].

The digital hosting media include text, HTML, e-mail, images, audio, video, disk space, disc partitions, network packets and they work by replacing the unused or underused bits in these digital files for secret information. The hidden data can be plain text, encrypted text, even images [4] [5].

Given that the purpose of steganography is to conceal the true message, it can be used to hide the sending of information, within normal network traffic. Equally its can be used with other techniques either to attack a network making detection more difficult or to register Internet access and use.

III.TCP/IP NETWORK PACKETS AS HOST MEDIA

To hide information in the headers of each TCP/IP protocol packet, it is necessary to consider that this protocol is in fact made up of a group of protocols and they are all present both in the transmitter and in the receiver. This protocol pile is present in all computers that use the Internet and given that it was created more than thirty years ago, some of the fields are not used or can be modified without affecting the correct transmission of data.

The purpose of the headers is to provide essential data to the information receiver. The IP protocol heaters contain the necessary information for the packets to be appropriately directed. The transmission control protocol (TCP) is used to make the transmission of data more trustworthy. All Internet-connected computers use the TCP to transmit certain types of information.

DOI 10.17148/IARJSET.2018.5611

A. Header of an IP Packet

An IP packet is divided into two parts, a header and a data block (Fig. 1). Both parts have a variable size, the maximum length of the entire packet being 64KB. The header can be between 20 and 60 bytes and it contains essential information for the routing and submission and is divided into several fields, each with a specific purpose [10], however some of these fields store redundant information or are not used during transmission.

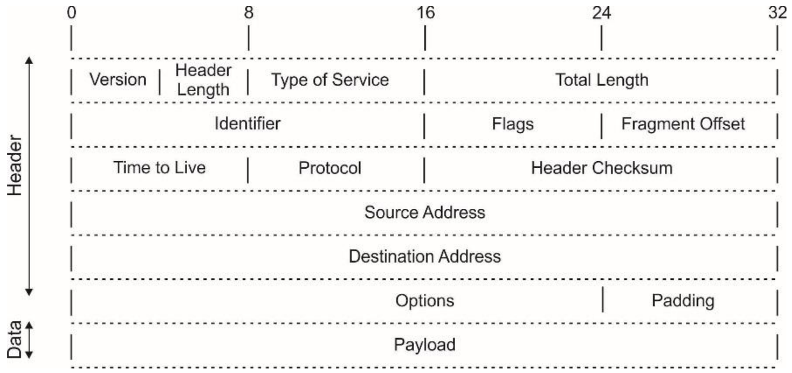


Fig. 1. IP Packet

B. Header of a TCP Packet

One of the protocols that can travel while encapsulated by IP is TCP and it is necessary for TCP to be connection-oriented since IP (with which it works closely) is not a trustworthy protocol. TCP divides data into segments and add a header (Fig. 2) [10]. It is important to remember that some methods used for the concealment of information in network packets exploit the weaknesses presented within some header fields due to the redundant information that they contain or to the fact that they remain unused during data transmission.

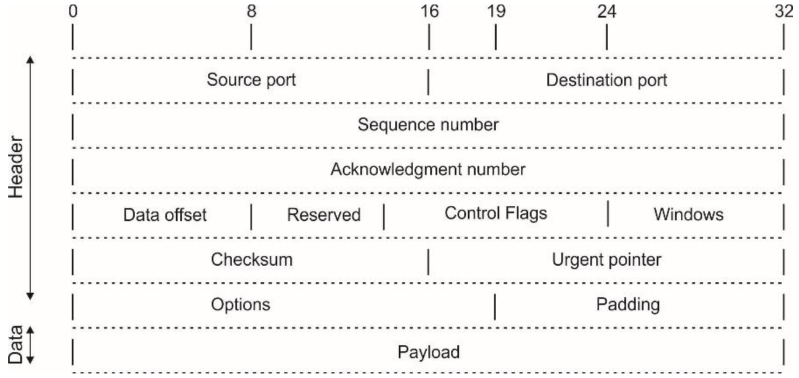


Fig. 2. TCP Packet

IV. RELATED WORK

Next we present some research related to our work. In [8] the writers have a specific purpose: to send information relative to the network's traffic and routing, in a point to point communication, without generating additional traffic. This work was not intended to send large quantities of information. The authors mention that they didn't consider coding nor special concealment methods.

In [11] the authors attempt to demonstrate that the sending of hidden information is feasible because of the intrinsic attributes of the Internet's protocols. Their efforts, therefore, do not work towards conducting concrete and specific tests, but instead on demonstrating the possibility of concealed sending, although they did estimate the possible bandwidth. One significant contribution is the proof that it is possible to send information while manipulating the order of the transmitted packets. The only important restriction is that it must operate within IPv4.

On the other hand, in [6] Rowland is focused more on network security and Internet transmission, in that he deals with aspects relating to data theft. Nonetheless he does propose the sending of totally concealed information of a single bit hidden inside each of the three packet fields.

Finally, Allix in [12] offers a broad and comprehensive analysis of methods for sending hidden information, as well as analyzing the attributes of IPv4 and IPv6. This author, like Rowland, is focused more on security aspects than the actual sending of concealed information. Nevertheless he makes several proposals for the sending of concealed.

V. THE MUKUL STEGANOGRAPHY SYSTEM

The Mukul steganography system is composed mainly of two modules: ANA which is responsible for the concealment and sending of the message and BOB which is responsible for the reception and recovery of the message.
The ANA module is essentially comprised of a sniffer and a packet injector. The main tasks of ANA are encoding the message that is to be sent secretly, catching the TCP/IP packets that circulate around the network and selecting ideal packets in which the secret bits can be saved. In turn the injector is responsible for sending stego-packets via the network in order for the secret message to arrive to the destined recipient.
The BOB module essentially consists of a sniffer which 'sniffs' the network looking in search of the packets containing secret bits in order for them to be extracted, decoded and finally for the original message to be recovered.

A. Description of functionality the Mukul system

The Mukul system works in the following way:

1. It starts by reading the secret message in plain text which is formatted, encoding for transmission.
2. From ANA, and through a sniffer, the network packets are read in promiscuous mode, with the intention of capturing all packets circulating around the network.
3. Captured packets are analyzed with a view to selecting those that are ideal for the concealment of data, the attributes that establish this idealness are:

- They are directed to the recipient to whom the hidden information is being sent.
- They have unused bits.
- They do not have segmented information: they are unique.

4. The secret message is received and all possible bits are concealed for sending in fields from the IP and TCP header that were identified as ideal
5. From ANA, through an injector, stego-packets are injected into the network, with the data hidden in the headers and the rest of the data left exactly as captured from the network.
6. Meanwhile, from BOB, the sniffer function is waiting to capture the TCP/IP packets that are headed to this node (normal mode) and whose source node is specifically the transmitter node via which ANA is running.
7. When all packets have been captured at the receptor node, BOB unpacks the hidden mesage and shows it in plain text.

It is important to mention that Mukul Works to the following rules:
a) Messages to be sent must contain text only.
b) Data will only be hidden in the headers of messages addressed to the sertain pre-established recipient.
c) There is no acknowledgement of receipt by the receiver.Software architecture of the Mukul system.

The ANA module is essentially comprised of a sniffer and a packet injector. The main tasks of ANA are encoding the message that is to be sent secretly, catching the TCP/IP packets that circulate around the network and selecting ideal.

B. Software architecture of the Mukul system

The Mukul system is developed in the GNU/Linux operating system environment, distributed by Kanotix™ with Kernel version: 2.6.24-3-kanotix. The system was coded in C ANSI language, using the GNU/Linux compiler G++ 4.1.2. The C libraries used were: Libnet 1.1.2.1 and Libpcap 0.9.5. For the graphics interface QT 3.3.7 was used and finally Wireshark version 0.99.4 was also used as a protocol analyzer.

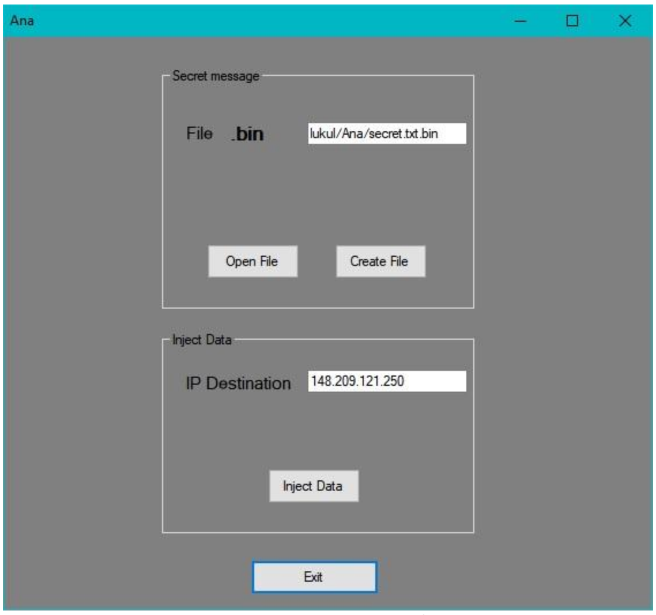
The fields of the TCP/IP packets used by the Mukul system to send the hidden information are described in Table 1.

TABLE I
FIELDS USED

Field	Header	Available bits
ToS	IP	4
TTL	IP	1
Id	IP	16
Bandera DF	IP	1
Urgent Point	TCP	16

VI. OPERATION OF THE MUKUL SYSTEM

Next we show a test run as well as the usability of our system's graphic interface. After the welcome screen, ANA shows the following screen (Fig. 3) and it is necessary for the user to enter the requested information:



File .bin: lukul/Ana/secret.txt.bin

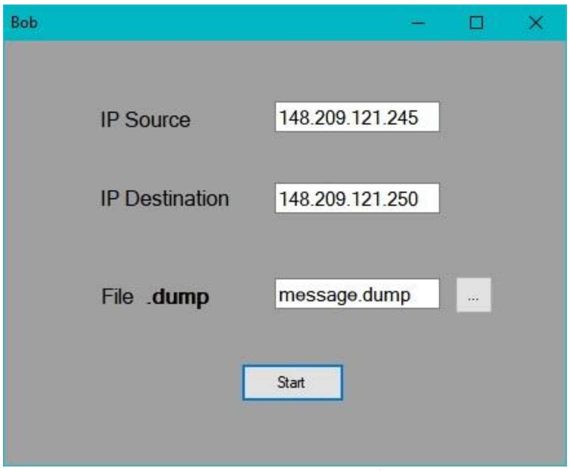
IP Destination: 148.209.121.250

Fig. 3. ANA system interface

The .bin file is the encoded secret message, if the user does not already have one, it can be created with the 'Create file' button.

The 'Inject data' button is pressed to trigger the injection of packets with the bits already hidden inside, the application sends a message to start the packet injection and another when the injection of the secret message has finished.

BOB meanwhile is running in the receiver in readiness for the capture of the stego packets. The BOB interface appears as shown below (Fig. 4). The information required for this interface is the source IP and destination IP as well as the name of the file in which the captured packets are to be saved.



IP Source: 148.209.121.245

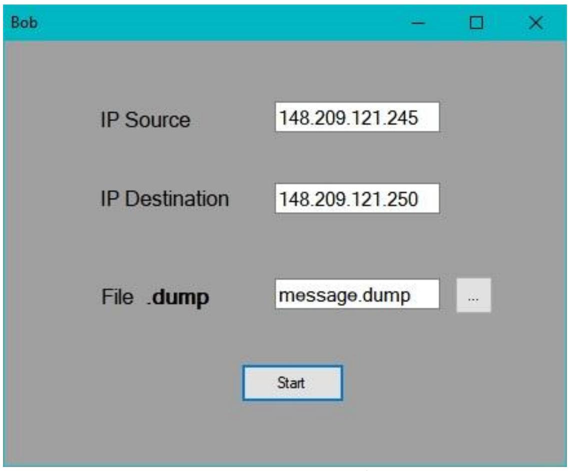
IP Destination: 148.209.121.250

File .dump: message.dump

Fig. 4. BOB system interface

Once reception is finished, the application presents the user with the following interface to decode the message (Fig. 5). The received secret message is written in a text file that can be viewed in any editor.

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588



IP Source 148.209.121.245

IP Destination 148.209.121.250

File .dump message.dump

Fig. 5. Decoding interface

Once reception is finished, the application presents the user with the following interface to decode the message (Fig. 5).

VII. SYSTEM TESTING

Testing was conducted according to ANSI/IEEE 1008-1987 norms [13], over the LAN network of the Tizimín campus of the Autonomous Yucatan University (UADY). This is subnetwork of the UADY network, geographically separated by at least 200 Km. The WAN connection is made via a dedicated E1 line with 1 GB of bandwidth. The LAN is a switched Ethernet network with two assigned subnetworks each with up to 255 nodes; one used for students and the other for staff. Tests were conducted on subnetwork 148.209.121.0/24.

Prior to the carrying out of the tests, it was necessary to obtain a pool of data, in this case TCP/IP packets. For with, with authorization from the network administrator, a program called DUMPER.CPP was installed to catch all packets transmitted over a 30 minute period. With this it was possible to collect a pool of 216 packets, a file of just over 15MB with an average length of 236 Bytes per packet, which we took as a normal distribution in terms of attributes sought in each packet by the Mukul software. All tests were done using the same data pool and they were all initiated from the first packet, in order to continue reading them sequentially. It is worth reiterating that the purpose was to demonstrate the concrete possibility of sending hidden information in the header of TCP/IP packets.

1. The testing plan included the following steps in chronological order:
2. The ANA module catches TCP/IP packets, correctly reads the addreses and chooses messages destined for the desired address.
3. The ANA module analyzes the header contents of the chosen messages and determines the possibility of including a hidden message within the packet.
4. The ANA Module determines the number of bits that can be included in each of the selected packets, identifies the same number of bits in the message to be send and registers the las bit position to be included in the current packet.
5. The ANA Module includes in hidden form the bits that can be embedded in that specific packet.
6. The ANA Module inserts the packets with hidden information into the network.
7. The BOB Module registers the packets and correctly reads the recipient identifier.
8. The BOB module chooses the packets that go from a specific origin to the intended final recipient.
9. The BOB module determines if there is hidden information in the selected packet.
10. The BOB module extracts the hidden information from the packets.
11. The BOB module constructs, with the bits that it receives, the original message.

The data pool that was used has attributes described in Table 2.

TABLE II
CHARACTERISTICS OF TEST DATA

Concept	Quantity
Total TCP / IP packets	65,535
Total source addresses	238
Total destination addresses	289
Total Source-Destination links	196
Average of packages from a specific origin to a destination	226

DOI 10.17148/IARJSET.2018.5611

VIII. RESULTS

The evaluation of the sending of hidden data, via a stego-message with limited length of 38 bits showed that a message can indeed be sent in a concealed state. In the tests carried out it was proven that up to 9.14 Kb of secret message can be sent from the source computer and this can be correctly reconstructed by the recipient, this is due to the use of a field of length of 16 bits in order to send to the recipient the total length of the hidden message that is to be recovered. If a larger number of bits are sent, the bits arrive at their destination inside the stego-packets but the recovery of the message is no longer possible. In none of the tests were we able to find inconsistencies or non-conformity. As such it is assumed that the Mukul software worked effectively.

Thus it has been established that:
1. The sent message M arrived at its destination and could be interpreted.
2.The header of packet n did not suffer changes to its information on arrival at its destination.
3. The information inside packet n did not suffer changes to its information.

IX. CONCLUSIONS

With these test results it is possible to establish that the necessary software was developed to enable the masking of information in the header of a TCP/IP message.
1. Compaction techniques: in this case only standard character text was sent, which in ASCII code range from 0 to 128, that is to say are encoded with 7 bits, if both upper and lower case letters are used; in total 50 characters, they could be encoded with only 6 bits. In fact, if only uppercase letters are used, only 4 bits would be necessary, and this way the transfer rate could be doubled.
2. Substitution cryptography techniques: in the case of text messages, it was established in [1] that languages use less than 2000 words to transmit 96.9% of messages. Thus, with a substitution table, for each word only 11 bits would be required regardless of the length of the word. This way it would be possible to send up to 2.5 words per packet. Although in this case it would be necessary for the sender and receiver to share the substitution table before sending each message.
When development of the software began, it was decided that, for simplicity, it should be via point to point communication. However these conditions are not always present. Similarly, Mukul was conceived based on conditions without communication issues and assuming communication would not be interrupted.
Due to the design of the web, all packets were IPv4, this way it was not necessary to write specific code for each protocol.
Another aspect that was assumed was the length and type of message. It was established in the testing protocol that only text messages would be used. In general in the entire design and programming of Mukul, measures were not taken to ensure optimum resource consumption; we were looking exclusively at establishing the efficacy of the process, not its efficiency.
Given the steganographic nature of communication, it is not possible for ANA (sender) to know if BOB (destination) received the message; this is a natural attribute of steganography: the sender will not know if the recipient received the complete and correct message.
With the above work we have demonstrated the real and concrete feasibility of sending hidden information in the header of TCP/IP packets.

X. FUTURE WORK

In the field of steganography there are several areas which still need investigating, from theoretical aspects to concrete programming elements. Among these we would highlight the following:
1. Establish whether this scheme works equally well with IPv6 protocol.
2. Determine how to send information when the connection is not point to point.
3. Determine the relevance of the latency period required for ANA to insert the hidden message into each packet; it may be necessary to develop optimized code to reduce the run time of ANA.
Finally, the study of steganography has a purpose beyond the mere sending of hidden information. Firstly it allows us to optimize network traffic, sending additional information without increasing traffic. Also it allows us to send information that should effectively be concealed in the context of packet content, such as intellectual property or data regarding the operation of the network and its elements. Finally, if we want to curtail the improper use of the web, it is necessary to understand how said improper use could operate, and establish the necessary safeguards.

ACKNOWLEDGMENT

To conclude, I thank **Dr. Roberto Gómez** for his help in the preparation of this work.

REFERENCES

[1] DOA. Basic Cryptanalysis, Department of the Army, Field Manual NO 34-40-2, <ftp://ftp.ox.ac.uk/cryptanalysis/basic_cryptanalysis.ps.tar.gz>. 2002 - 9

[2] STALLINGS, William. Crytography and Network Security: Principles and Practice. New Jersey: Prentice Hall, Cuarta Edición, 2005. 592p.

[3] KATZENBEISSER, Stefan and PETITCOLAS, Fabien A.P. Information Hiding techniques for steganography and digital watermarking. Estados Unidos: Artech House Publishers, 2000. 220p.

[4] CARRILLO, J. F., OSPINA, C., RANGEL, M., ROJAS, J. A., VERGARA, C., 2003. Covert Channels over HTTP. Tech. rep., Universidad de los Andes, [online]. [Accessed 27th Feb 2007]. Available from the World Wide Web: <http://www.criptored.upm.es/guiateoria/gt_m142m.htm>. - 21

[5] JOHNSON, Neil F.; DURIC, Zorac; JAJODIA, Suchil. Information Hiding: Steganography and Watermarking - Attacks and Countermeasures. Estados Unidos: Kluwer Academic Publishers, 2003 - 7

[6] ROWLAND, Craig H. Covert Channels in the TCP/IP Protocol Suite. Peer Reviewed Journal on the internet. 1997. - 13

[7] COHEN, Frederick. Curso Abreviado de virus en computación. México: Limusa, 1998. 27

[8] CAUICH, Enrique; GÓMEZ Roberto; RYOUSKE Watanabe. Data Hiding in Identification and Offset IP fields. Lecture notes in computer science. Springer, New York. Enero 2005. - 22

[9] ABAD, Carlos. IP Checksum Covert Channels and Selected Hash Collision. University of California, USA, http://downloads.securityfocus.com/library/ipccc.pdf 2001. -31

[10] FOROUZAN, Behrouz A. TCP/IP Protocol Suite. 2a. ed. Nueva York: McGraw Hill, 2003. -35

[11] KUNDUR, Deepa and AHSAN, Kamran. Practical Internet Steganography: Data Hiding in IP. ACM Workshop on Multimedia Security. 2002. - 23

[12] ALLIX, Pierre. Covert channels analysis in TCP/IP networks, Orsay, France. IFIPS School of Engineering, University of Paris Sud XI. 2007

[13] ANSI/IEEE. Standard 1008-1987, IEEE Standard for Software Unit Testing. American National Standards Institute/ Institute of Electrical and Electronics Engineers. New York. 1993. - 41

BIOGRAPHIES

Erika Rossana Llanes Castro
Master in Computer Sciences for the Institute Technology of Monterrey in México, is professor ofthe Autonomous University of Yucatán. His researcher lines: network security,mobile programming, programming fundamentals.

Lizzie Edmea Narváez Díaz
Master of Computer Science for the Institute Technology of Monterrey (ITESM), Campus Cuernavaca. She has been a full time teacher at the Autonomous University of Yucatán since 2000 in the Networking department in Tizimin Mexico. She has participated in software engineering development projects

Victor Manuel Chi Pech
Master of Computer Science for the Institute Technology of Monterrey (ITESM), Campus Cuernavaca. He has been a full time teacher at the Autonomous University of Yucatán since 2000 in the Networking department in Tizimin Mexico. He has participated in software engineering development projects

DOI 10.17148/IARJSET.2018.5611