Steganography Using TCP / IP's Sequence Number

Jamal M. Kadhim and Abeer E. Abed
Department of Computer Science, College of Science, Al Nahrain University, Baghdad-Iraq.
Corresponding Author: aroma.jesus@yahoo.com.

Abstract

The exchanging process inside LAN or through the Internet may be exposed to be stolen, altered or damaged by a baleful person who was represented as real threats to transport process and also to information especially if this information was sensitive, important and must be accessed by only authorized person. Therefore this data must be secured against such threats. Steganography is classified as the art of hiding information in a suitable carrier like image, text, audio and video. It can also be applied to network protocols whether to header or payload of the packet. In this paper, steganography was implemented on the header of IP packet protocol and/or TCP segment protocol by exploiting the sequence number field of this header for hiding information inside it. In addition, four characters would be sent in every connection. So to increase the amount of information which could be sent, the number of connections must be increased. [DOI: 10.22401/JNUS.20.4.16]

Keywords: Steganography, TCP/IP, hiding information, and Sequence number.

1- Introduction

After computer appearance and pervasion in most institutes like companies, universities, and homes, the network appeared. Since these places had limited hardware resources like printers, scanners...etc, so individuals especially employees needed to share these resources, so they had no need to have for example a private printer. Also, after growth of most institutes which was represented by having different branches distributed in different places like companies and universities, the individuals really need to communicate and at the same time to exchange important information to do their work, so the need to connect these institutes by network became necessary [1].

Network in simple case connects at least two computers using connection. Network may be built in homogeneous form (i.e. every computer in network has the same operating system, Network Interface Card (NIC) … etc) like Local Area Network (LAN's) or in heterogeneous form (i.e. every computer has different operating system, … etc) like network of network (Internet) . As a result, these developments in communication means leads to exchange huge information. The exchanging process inside LAN or through the Internet may be exposed to be stolen, altered or damaged by a baleful person who was represented as real threats to transport process and also to information especially if this information was sensitive, important and must be accessed by only authorized person. Therefore this data must be secured against such threats. Many ideas was suggested under security concept for protecting data from these threats such as hiding content of the message sent which was named cryptography [2] or concealment the existence of such message which was named steganography. This paper shows what is steganography historically and recently, TCP/IP protocol suite, methods that utilized TCP protocol header for steganography, implementation of steganography in TCP/IP, the proposed system, practical implementation, results and conclusions.

2- Steganography

Previously, especially in ancient Greece, Herodotus (c. 486-425 B.C.) was told about how a message was sent for inciting mutiny against the Persians. So to guarantee that, anyone could not notice that, Histiæus chose most faithfull slave and shaved his head, tattooed it with the required message and waiting until his hair had regrown, then he was sent. Another story was told by him was that warning message was sent by Demeratus, a Greek at the Persian court to Sparta included that Xerxes, King of Persia wants to foray it. So, he inscribed the message on tablet and

covered it with wax in which was appeared as if it was not used [3]. Recently, with computers and information technology emergence, hiding information takes another direction by using text, image, audio and video which represents suitable carrier for transferring secret information. Also, Protocols of TCP/ IP protocol suite can be utilized to transfer these information which was known as Network Steganography [4] and was introduced firstly by Krzysztof Szczypiorski in 2003 through implementing hidden data in HICCUPS [5]. Before that, the concept that was known as covert channel produced by Lampson in 1973 which provides desirable environment. It means utilizing unused fields of network protocols or changing uncritical data [6]. This leads to create carrier for hiding information far away protocol specification. Because the requirement of steganography for carrier to hide information, the network steganography can be implemented through existence of covert channel [7].

It was obvious whether in historical or modern steganography that this process needs essential elements for hiding which was represented in :

1. Carrier: also known as cover-object and defined as the place where message was included and conceal existence of it.
2. Message: represented the data that the sender wants to be secret.
3. Password: also known as stego–key that represented decoding key which was known only by recipient and hence would be extracting hidden data from cover object [8].

3- TCP/ IP Protocol Suite

TCP/ IP protocols suite consists of many protocols but its name came especially from two main protocols, Transmission Control Protocol (TCP) and Internet Protocol (IP). IP is responsible for providing addressing and routing globally. So, it supports universal onnectivity. Also, it supports fragmentation [9]. Most important characteristic is that it is best effort or connectionless. The meaning of best effort is that the datagram could be lost, corrupted or reached out of order [10], while connectionless means that before transmitting the data, there was no exchanging control information (known as handshake) to establish end – to end connection which is provided by connection oriented protocols for example TCP. So, it does not provide acknowledgments or retransmission when packet lost or out of order and when connection oriented was required, IP must depends on protocols of the upper layer that provided it [11]. Datagram consists of header and payload. The length of header is variable and ranges between 20-60 bytes proportion to the existence of option field. This variation results in variable length datagram. The header of IPv4 is shown in Fig.(1) [10].

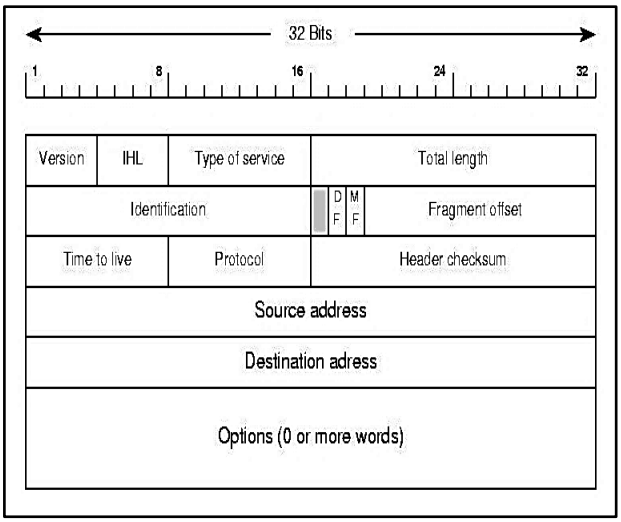

Fig.(1) : IP header format.

TCP is a connection oriented protocol, meaning that the data must be transferred after the connection is established. In addition to establish connection and data transmission, the connection must be released. Because it uses sequence number to ensure delivery correctness and confirmed no data was lost when network failure occur, so it was considered a reliable protocol. Also, it is stream-orientation because it uses buffer in sending and receiving and this enable application to write very small or an amount of data and divide it into appropriate size [9]. It is also process–to–process communicated through using port numbers [12]. Port numbers from (1- 1023) are system ports. Port numbers from (1024- 49151) are registered ports while ports from (49152-65535) are private ports. TCP segment consists of header part and data part as shown in Fig.(2).

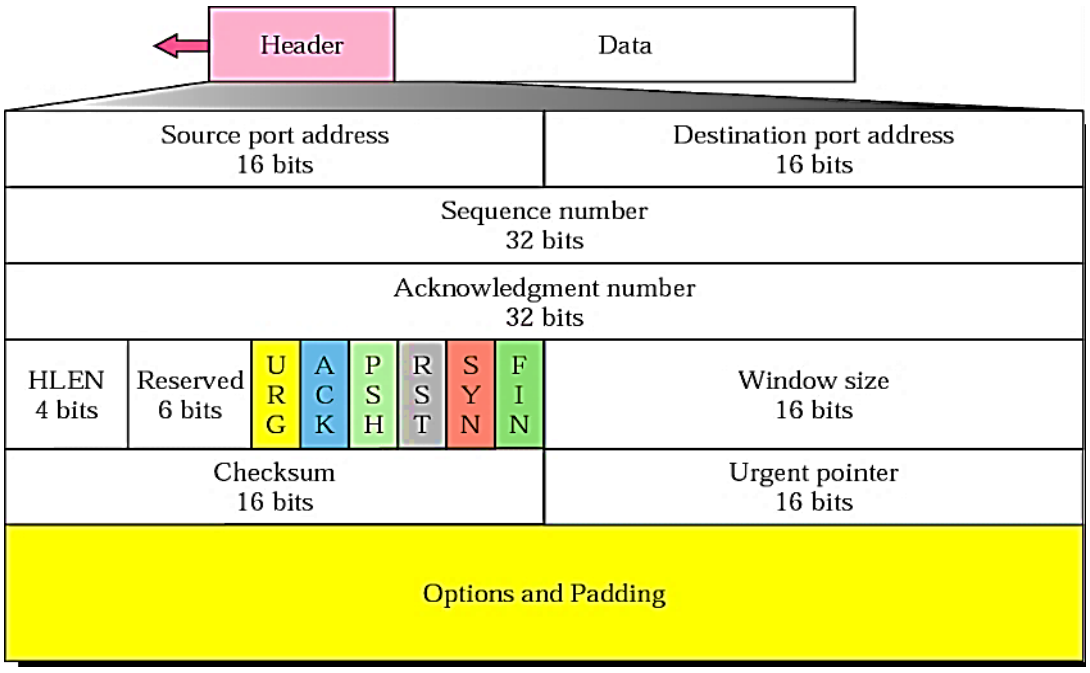

Fig.(2) :TCP header format.

4- Literature Review

Rowland used *Initial Sequence Number Field (ISN)* of TCP for hiding the information. It was done by *multiplying* ASCII of each character with (65536 * 256) to generate number which was placed as sequence number value for each connection. On the receiver side, opposite process applied to get the character by *dividing* sequence number's value on (65536 * 256). The big disadvantage, when every character is transferred through a connection, is that many requests was made to connect to the server without receiving SYN/ ACK packet would attract the attention [13].

Joanna implemented steganography by changing the sequence number field in the packet which is generated by compromised the system using NUSHA tool. NUSHA sender did not generate its own packet, but instead it modifies the sequence number and the acknowledgement number that were generated by the operating system by saving the TCP state, so when packet with the modified initial sequence number was sent, the original initial sequence number was stored and when a reply was received which included the acknowledgement number (modified sequence +1), it must be replaced with the original sequence +1. This made NUSHA process complicated since it must store every sequence and acknowledgement numbers. In other side, recipient could be applying a simple sniffer by staying on trusted gateway or using technique that force traffic toward its interface network card to capture all the packets [14].

Ciobanu suggested SCONeP (Steganography and Cryptography Over Network Protocols) and using ISN (Initial Sequence Number) after solving the issue which appeared in [13] by sending RST (Reset) packet to abort a connection instead of ACK packet after 4-bytes will be transmitted. Identification field was also used after DF(Don't Fragment) -bit was assigned to 1 to avoid modifying it by kernel since it must have value (1) when packet do not need fragmentation [7].

Singh implemented steganography by using identification field (16-bit) with initial sequence number field (32-bit) for disguising (6-bytes) of character after encrypted it using an algorithm which was chosen by the sender and whether to compress it or not is determined by the sender [4].

Biswas in his research, sequence number was used as a carrier for RSA/ DES key that was used to encrypt the data then, the ciphertext was embedded in the data field. The receiving packet is captured through the wireshark application. After the ciphertext was taken from the data field and obtaining the key from the sequence number field, the ciphertext is decrypted to get the data [15].

5- TCP/ IP based Steganography implementation

Sequence number field was chosen for steganography. It means that TCP and IP headers must be created manually. Because in client – server architecture, operating system's kernel was taking care of adding required headers for data. Some fields of the two protocols headers must remain unchanged through data transmission while the others must be changed. The unchanged fields of IP protocol header are version, header length, and type of service. Other fields like identification, flags, fragment offset (that specified for fragmentation strategy was changed through fragmentation), total length, time to live, checksum, source address, and destination address would be changed. Protocol field had (6) value that pointed to payload of IP header is TCP segment ignoring other values of this field. In other side, fields of TCP header like source port, destination port, sequence number, acknowledgement number, header length, flags, checksum, and window field should be changed through transmission. Regardless of which operating system was used, root privilege level access should be used since custom header of packet was created. Sending packet that included the hidden data in sequence number field was in SYN packet. It is the first packet in three way handshake process. Receiving the packet to extract the hidden data from, was in SYN-ACK packet which represents the second packet in three way handshake and also acts as a reply to the connection request in SYN first packet.

5.1 System Model

The structure of the proposed system for Steganography using TCP/IP's sequent number is illustrated in Fig.(3). It consisted of two models. The sender (client) and receiver are the two models respectively.

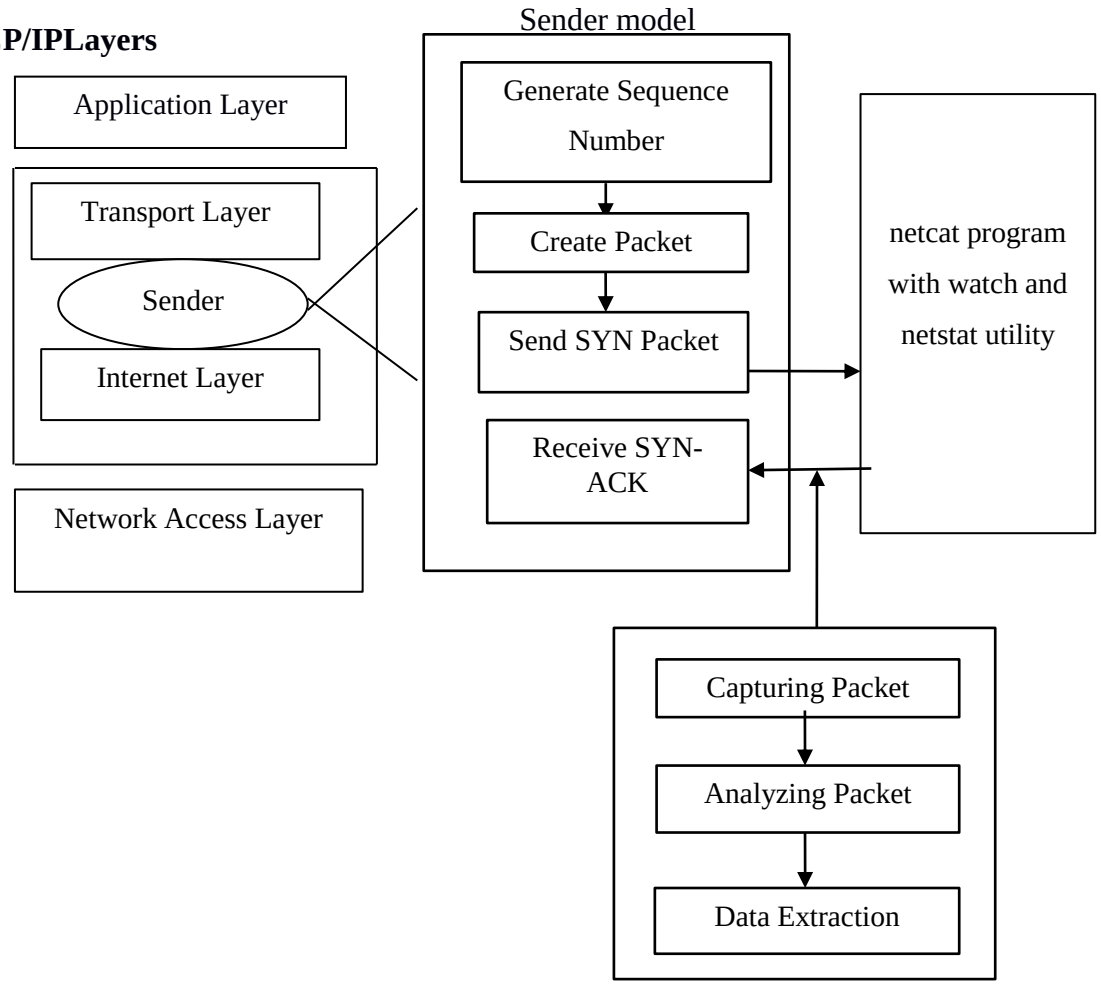

Fig.(3): System model.

Sender

Step 1: calling create IPHeader function
- Step a: global definition applied to necessary variables
- Step b: set values to source and destination addresses
- Step c: pass source and destination addresses to IP class's object
- Step d: set result of grouping IPHeader's fields to iph variable
- Step e: return

Step 2: read data to be hidden
Step 3: calling createTCPHeader function
- Step a: global definition applied to necessary variables
- Step b: set values to source and destination ports, empty string to data
- Step c: pass source and destination ports to TCP class's object
- Step d: changing value of sequence number
  - Step 1: converting data's characters to decimal
  - Step 2: convert result of step1 to binary, then set result to bindata variable
  - Step 3: converting source port to binary, then set result to binsrcport variable
  - Step 4: converting destination port to binary, then set result to bindestport variable
  - Step 5: concatenation source and destination ports in their binary representation, then set result to binsrcdest variable
  - Step 6: implementation XOR operation between result of setp3 and step 9, then set result to binsteg variable
  - Step 7: converting binsteg to decimal, then set result to sequence number
- Step 8: return sequence number
- Step g: set SYN bit
- Step h: set length of data to data_length field
- Step i: set result of grouping TCPHeader fields to tcph variable
- Step j: return TCPHeader with the hidden data

Step 4: create packet
Step 5: create Internet raw socket object
Step 6: check for creation error, if true then system exit
Step 7: pass created packet and address to sendto function, then send it
Step 8: return

Capturing and Analyzing at layer 2

Input:

Destination port

Output:

The Hidden data

Begin

Step 1: reading destination port
Step 2: calling capturing and analyzing packet with destination port passing as Input
Step 3: global definition applied to necessary variables
Step 4: create packet raw socket object
Step 5: while condition true
Step 6: receiving packets through recvfrom function
Step 7: extract ethernet header
Step 8 : extract IP header
Step 9: extract TCP header
Step 10: check if destination port == the one passed as input then break, Else go to step 5
Step 11: calling data extraction function
- Step a: converting acknowledgement number to binary, then set result to ackn variable
- Step b: converting source port to binary, then set result to binsrcport variable
- Step c: converting destination port to binary, then set result to bindestport variable
- Step d: concatenation source and destination ports in their binary representation, then set result to binsrcdest variable

Step e: implementation XOR operation between result of step a and step d, then set result to binsteg variable

Step f: converting result of step e to decimal, then set result to dessteg variable

Step g: converting result of step f to data, then set result to hiddendata variable

Step h: return hiddendata

Step 12: return

6- Practical Implementation

The proposed system is implemented using Linux kernel raw sockets. Linux is a good environment provides the easiest access of raw sockets interface. Virtualbox is the tool that is used to setup ubuntu 15.10 for using netcat. Netcat represents the server that listens and waits for a connection request from the client (sender of hidden data).

7- Results

The execution of the program for the method (described above) is shown below. Fig.(4) shows a packet with the hidden data. The sender hide the data using source and destination ports fields of TCP header as stego - key and generate sequence number from it.

```
In [1]: runfile('/home/abeer/Documents/ResearchProgram/finalinjection.py',
wdir='/home/abeer/Documents/ResearchProgram')

data is : from
tcpobj.seqn= 2842043300
Full SYN-packet created ip_header + tcp_header + user_data..
Packet sent to  192.168.0.101
```

Fig.(4) : Sending packet with hidden data.

As seen the sequence number value is (2842043300) represents the result of XOR operation between data and stego – key (described previously). Packet analysis on Ethernet is shown in Fig.(5)

```
In [1]: runfile('/home/abeer/Documents/ResearchProgram/raw-ethernet-sniffer1.py',
wdir='/home/abeer/Documents/ResearchProgram')

Please specify a port number to listen on(Default is all): 8080
****
Source Port Number:  28098
Acknowledgment Number:  2842043301
from
```

Fig.(5): Packet analysis on Ethernet.

As seen acknowledgement number value is (2842043301) represents the sequence number +1 and the hidden data extracted from. To ensure that packet is transmitted, the packet sniffing program Wireshark will be used to collect all the transmitted packets of my network.

7- Conclusions

In this paper, a steganography using TCP/IP's sequence number was constructed. The basic requirements for hiding data in such carrier were described. A good mixture for network programming was shown through python as a modern programming language and Linux. TCP/IP header fields were found as a good carrier for sensitive data to be hided. With few minor operations with the field's contents, a secure carrier can be constructed. The serious issue was the restriction of field's sizes, since it is limited to a fixed size. This limitation may restrict the size of data in turn. Therefore, in order to send more data, the sender has to increase the connections.

References

[1] Tanenbaum A. S., Wetherall D. J., "COMPUTER NETWORKS", 4th Ed., Pearson Education, Inc, 8-13, 2003.

[2] [STA11] Stallings W., "Network Security Essentials: Applications and Standards", 4th Ed., Pearson Education, Inc., 8-12, 2011.

[3] [KAT00] Katzenbeisser S., Petitcolas F. A. P. "Information Hiding Techniques for Steganography and Digital Watermarking", Artech House, INC, 3-4, 2000.

[4] [SIN13] Singh J., Sharma L., "Framework for Efficient Secure Steganographic Communication over Network Protocols", International Journal of Advanced

Computer Research, 3 (4), Issue 13,146-150, Dec. 2013.

[5] [SZC03] Szczypiorski K., "HICCUPS: Hidden Communication System for Corrupted Networks", ACS '2003: Proceedings of The Tenth International Multi-Conference on Advanced Computer Systems, 31- 40, 2003.

[6] [ROH11] Rohankar N, D., Deorankar A. V., Chatur, Dr. P. N., "A Review of Literature on Design and Detection of Network Covert Channel", International Journal of Engineering Science and Innovative Technology (IJESIT), 1, Issue 2, Nov.2012.

[7] [CIO06] Ciobanu R., Mihai-Ovidiu Tirsa, Raluca Lupu, Sonia Stan, "Steganography and Cryptography Over Network Protocols", RoEduNet International Conference 10th Ed.: Networking in Education and Research, 2011.

[8] [AMI03] Amin M. M., Ibrahim S., Salleh M., Katmin M. R., "Information Hiding Using Steganography", Universiti Teknologi, Malaysia, 2003.

[9] [DOR16] Dordal Peter L., "An Introduction to Computer Networks", http://intronetworks.cs.luc.edu/, 161, 303, 2016.

[10] [FOR07] Forouzan B. A., "Data Communications and Networking", McGraw-Hill Press, 4th Ed., 583-584, 2007.

[11] [HUN02] Hunt C., "TCP/IP Network Administration", O'Reilly Media, Inc., 3rd Ed., 13, 2002.

[12] [FOR10] Forouzan B. A., "TCP/IP Protocol Suite", McGraw-Hill Press, 4 th Edition, 375, 2010.

[13] [ROW97] Rowland C. H., "Covert Channels in the TCP/IP Protocol Suite", First Monday Journal on the Internet, 2(5), 1997.

[14] [RUT04] Rutkowska, J., "The Implementation of Passive Covert Channels in the Linux Kernel", Chaos Communication Congress, Dec. 2004.

[15] [BIS16] Biswas R., Bandhyapadhyay S. K., "TCP Packet Steganography using SDA Algorithm", Journal of Scientific and Engineering Research, 3(2): 47-51, 2016.