The Implementation of Covert Channel in IPV6 using Linux Kernel

Simran D Munot[1,], Mitali J Yevale [1], Nikhil M Raut [1], and Dhananjay M Dakhane[2]*

[1] Department of Computer Engg, Ramrao Adik Institute of Technology, Nerul, Navi-Mumbai, India
[2] Professor, Department of Computer Engg, Ramrao Adik Institute of Technology, Nerul, Navi-Mumbai, India

Abstract. Nowadays indirect communication is used as a mechanism to bypass the direct communication network channel. Such indirect communication channels prove as a threat to information-sensitive systems that don't allow leakage of information like Military communication. Developing a covert channel in the network requires some carriers like TCP Headers or Protocols. The only challenge is that it possesses strict semantics. If it is manipulated by even a single bit, the entire semantics will change. The packet will be considered as a forged packet and will get rejected automatically by a firewall. To avoid this, we have proposed a unique way of covert communication - TCP and IPv6 Referencing Model. In our proposed referencing model, instead of embedding any bit and destroying the semantics of the header, we are going to use the referencing method to send the covert bit from sender to receiver end. Thus the packet will appear a part of normal distribution and gets transmitted securely within the channel. Here we are using the TCP Sequence Number (SQN) to make this possible.

1 INTRODUCTION

The communication channel is defined as a mechanism to transmit any message from the sender device to the predetermined host device. Currently, the communication channels can be categorised into two: Overt and covert communication channels. The overt communication channel is defined as the mechanism which is followed ethically to establish communication between two communication devices. The covert communication channel is defined as the false mechanism that is followed by the devices to establish a communication channel between them. Today, a plethora of devices are now internet protocol (IP) enabled. Increased complexity in communication has made the task challenging for the network administrators to define a security policy that ensures an ethical way of communication between the host and the destination. Defining a new and own proprietary protocol will open up the possibilities of a security breach. As stated by Zander and Van Horenbeek in their respective studies - The real-world applications and the academic research on this topic has a huge gap; the current academic research is uncertain due to the applicability of some of the approaches on detection of such unethical communication channels. Now the question is which carrier to utilize for covert communications? Given in the present era, the choice of protocols, Internet Protocol version 4 (IPv4) currently, is one of the most pervasive protocols in deployment. It has become a good carrier for covert channel exploits because of its use in most of the communication over the globe. The IPv6 is designed for the successor of IPV4. As IPv4 is exhausting its address space for future use. It is very obvious that the focus will shift towards the IPv6 as a carrier for network covert channels. The covert channel in IPv6 is more secure like buffer overflow attacks

2 RELATED WORK

To identify the possible covert channels, R.Kemmerer in 1983 shared a resource matrix methodology that was used to analyse a constructed resource matrix. [13]

In a network environment in an OSI ISO framework, Girling, on a local area network (LAN) in 1987 first studied covert channels. He identified two storage covert channels and one covert timing channel on the LAN environment. [12]

In this protocol there are some reserved, padding and undefined fields which are used in TCP/IP protocol, In 1989 Wolf implemented the covert channels in LAN protocols. [15]
For a different TCP/IP to tunnel through, T. Handel of Stanford University in 1996 permitted a proprietary protocol that could open a security breach. The real-world applications and the academic research on this topic has a gap as pointed out by Zander [1] and Van Horenbeeck [2]; the current academic research is uncertain due to the applicability of some of the approaches on detection. Now the question is which carrier to utilize for covert communications? Given in

* Corresponding author: dhananjay.dakhane@rait.ac.in

the present era, the choice of protocols, Internet Protocol version 4 (IPv4) currently, is one of the most pervasive protocols in deployment. It has become a good carrier for covert channel exploits because of its uses in most of the communication over the globe. The IPv6 is designed for the successor of IPV4. As IPv4 is exhausting its address space for future use. It is very obvious that the focus will shift towards the IPv6 as a carrier for network covert channel. The covert channel in IPv6 is more secure like buffer overflow attacks. It stated his work on Hiding the Data in the OSI Network Model. All the possible loopholes in the OSI layers were analysed by him. It also stated that for implementing covert channels for exploiting hidden channels within the standard design of network communications protocols, this paper gave a basis to the development of a tool kit.

'Covert_tcp.c' scheme was proposed by C.H. Rowland in 1997 in which he gave the three methods which revolve around the IP-ID exploitation of IPv4 header, the sequence number and acknowledgement fields of the TCP header.

As it is a simple embedding method in these fields for covert communication which can be easily identified by SVMs and achieved a classification accuracy of up to 99% [4].

The practical techniques and uses of internet photography were investigated by Kamran Ahsan in 2002. To covertly communicate supplementary data, Internet steganography is the exploitation of Internet elements and protocols. For a demonstration of this theory, he worked on the Network layer of TCP/IP stack and analysed the two existing covert channels. He also studied the relation of bandwidth with covert communication by violating the systems security policy. This indeed increased the performance of the network applications [8].

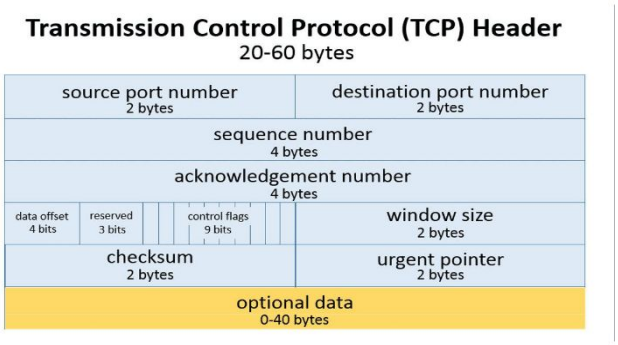
In 2011, Grzegorz Lewandowski started his work in the paper "Network-aware Active Wardens in IPv6". The main focus of the paper was the answers to the important questions of covert channel theory. The answers were about the possibility of the existence of the covert channel in the IPv6 protocol and the counter security measures to prevent such unethical channels. [16]

Using the "IP Next Generation Protocol", widely referred to as IPv6, as well as its associated protocol ICMPv6, Geoffrey Ackerman, Daryl Johnson and Bill Stackpole in 2015 proposed a covert channel in a paper titled "Covert Channel Using ICMPv6 and IPv6 Addressing." [17]

3 PROPOSED REFERENCE MODEL

Our aim is to establish a connection over the network by creating a covert channel using TCP and IPv6 Header and decrease the asynchronous command channels between the compromised system and its master.

Every data packet is associated with a solitary 4-byte Sequence Number which is present in the TCP header. The main objective of this field is to identify the sequence of the data packets and redirect them to the proper destination host. At the receiver end, this number is also used for reassembling the data packets.



The possible combinations for the Sequence number of TCP headers are around 4294967296 values. The operating system must have a particular mechanism for the identification and generation of Sequence Number.

For implementation purposes, we propose to develop a Linux Kernel Module(LKM) at the sender's as well as the receiver's side.

As the packet enters the transport layer the programs can register with the Netfilter hooks (five predefined hooks). As the packet progresses through the stackn(Fig 3.1), it triggers the kernel modules and registers itself with the required hook. Depending upon the state of the packet (incoming or outgoing), the hooks will get triggered.

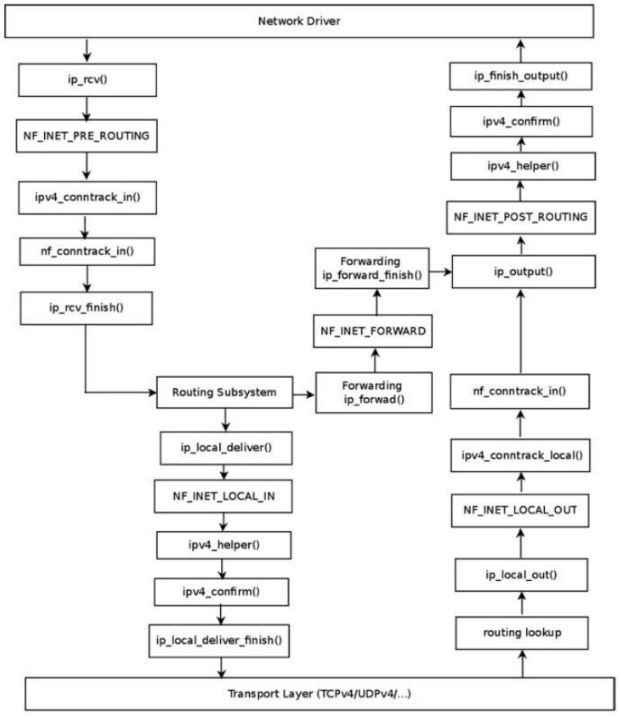


Fig.3.1: Flow of Data Packet From Network Layer to Transport layer.

The five pre-defined Netfilter hooks have well-defined points within the networking stack. They are as follows:

- NF_IP_PRE_ROUTING: Before making any routing decision (destination address), this hook gets triggered.
- NF_IP_LOCAL_IN: If the incoming data packet has the destination address of the local system then only the hook is triggered, otherwise the data packet is passed to the next hook.
- NF_IP_FORWARD: When an incoming packet is being forwarded to another host, this hook gets triggered.
- NF_IP_LOCAL_OUT: As soon as the packet enters the network stack, this hook gets triggered to create outbound traffic.
- NF_IP_POST_ROUTING: After routing takes place, this hook is triggered by any outgoing data packet or network traffic.

The order in which the hooks get triggered is determined by the priority number which is assigned to the kernel modules which wish to register themself in any of the five mentioned hooks.

A. Sender-Module

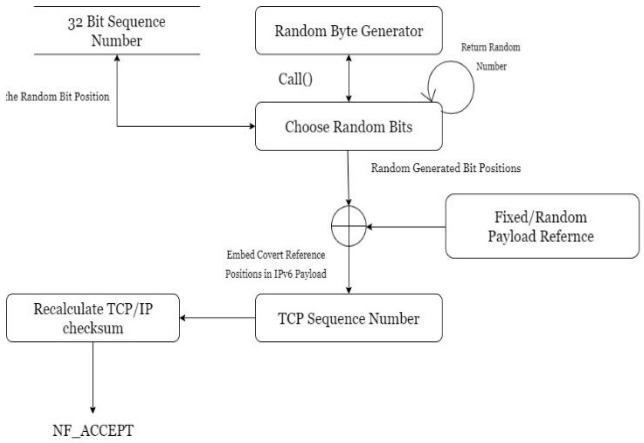


Fig 3.2: Reference Module - Sender Module.

The receiver receives covert messages from the sender through the sender module which is designed in our proposed system. Using the 32-bit Sequence Number field of TCP Header in this module, the referencing technique is used for the covert message. The architecture of the sender module is shown in Figure 3.2. In order to select the reference bit positions and embed these positions into the corresponding IPv6 payload, the Sequence Number field of TCP Header is taken into consideration. Instead of a simulated covert channel, an active covert channel is our proposed model.

32 bits per packet. i.e. total length of covert data to be sent per packet is the bandwidth of our proposed model. To elaborate, the entire 32-bits can be used as a maximum bandwidth.

Here, for all the possible bandwidths the procedure and base principles for covert information transmission are almost the same. In regards to the active covert channel, the intended process can be exploited.

B. Receiver Module

The receiving module will be similar to the sender module concerning the working mechanism. The receiver module is first initiated initially by registering itself inside the Linux kernel. As the data packet arrives at the sender end, travels through the application layer and enters the kernel level our Linux Kernel Model hook gets executed. With the help of the symbol table, the covert bits embedded in the Payload of the IPv6 Header is extracted. The bits are then converted into the actual message by the application level program.

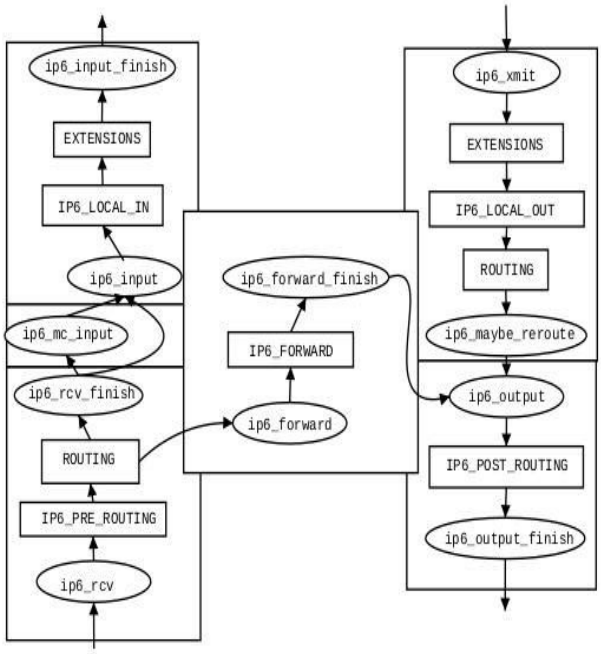


Fig 3.3: IP Routing inside the Linux Kernel.

4. RESULT

Consider the sender and Alice and receiver as Bob. Alice here wants to communicate with Bob covertly and send secure data to Bob. Suppose Alice (Sender) and Bob (Receiver) agree to use our proposed model for Communication, for sending Alice makes use of the CovertSender.ko while Bob uses CovertReceiver.ko

The table below describes the experimental results taken between Alice and Bob during their covert communication. The proposed model can be analysed using different bit sizes and different types of data used during communication.

Table No 1 contains the experimental result of covert communication when the sender sends 8-bit text data per packet, using our proposed model.

Covert Message	Text Data
File Size	2048 bytes
TCP Data Bytes	466944
Packet Received	2056
Covert Data	2048 bytes
Capacity = (Covert Data/TCP Data)	2048/466944 = 0.00438596
Bandwidth per packet	8 bit
Packet Loss	0.00%

Table 1. Results

Table No 2 contains the experimental result of covert communication when the sender sends 16-bit covert text data per packet, using our proposed model.

Covert Message	Text Data
File Size	2048 Bytes
TCP Data Bytes	233472
Packet Received	1026
Covert Data	2048 bytes
Capacity = (Covert Data/TCP Data)	2048/233472 = 0.00877193
Bandwidth per packet	16 bit
Packet Loss	0.00%

Table 2: Results

Table No 3 contains the experimental result of covert communication when the sender sends 32-bit covert text data per packet, using our proposed model.

Covert Message	Text Data
File Size	2048 bytes
TCP Data Bytes	116736
Packet Received	513
Covert Data	2048 bytes
Capacity = (Covert Data/TCP Data)	2048/116736 = 0.01754386
Bandwidth per packet	32 bits
Packet Loss	0.00%

Table 3*:* Results

5 CONCLUSION

While establishing a covert communication, one doesn't need to embed the data inside the TCP and IPv6 Header. Neither any bit of the TCP Header nor IPv6 Header has been modified in our proposed system. The syntax and semantics of TCP sequence number of overt communication are not altered to make it possible for covert communication. So that our covert communication is treated as overt communication. So that it will nullify the possibility of detection for passive as well as for active wardens. Hence we claim that our proposed model fulfils the criteria for "Legal and Implementation supported" type of covert channel.

That is the Network cannot distinguish between the covert and overt data packer. Hence the "Indistinguishability" characteristic has been achieved in our proposed system.

The covert channels bandwidth is an important characteristic of communication channels. Since communication is established using a single connection, there is no change in the bandwidth of the communication channel and a single persistent connection is used for communication.

6 References

[1] S. Zander, G. Armitage, P. Branch. Covert Channels and Countermeasures in Computer Network Protocols. IEEE Communications Magazine, December 2007.

[2] M. Van Horenbeeck. "Deception on the Network: Thinking Differently About Covert Channels," in Proceedings of 7th Australian Information Warfare and Security Conference, Decem- ber 2006.

[3] J. Postel, "Internet Protocol," RFC 0791, IETF, Sept. 1981.

[4] C. H. Rowland, "Covert Channels in the TCP/IP Protocol Suite," First Monday, Peer-Reviewed Journal on the Internet, July 1997.

[5] T. Sohn, J. Seo, and J. Moon, "A Study on the Covert Channel Detection of TCP/IP Header Using Support Vector Machine," Proc. 5th Int'l. Conf. Info. and Commun. Security, Oct. 2003, pp. 313–24.

[6] Fyodor: Idle scanning and related IP-ID games (2001). http://www.insecure.org/nmap/idlescan.html

[7] R. E. Best, Phase-locked loops: Design, simulation and applications. McGraw Hill Professional, 5th ed., 2003.

[8] K. Ahsan and D. Kundur, "Practical Data Hiding in TCP/IP," Proc. ACM Wksp. Multimedia Security, Dec. 2002.

[9] S. J. Murdoch and S. Lewis, "Embedding Covert Channels into TCP/IP," Proc. 7th Information Hiding Wksp., June 2005.

[10] E. Cauich, R. Gomez Cardenas, and R. Watanabe, "Data Hiding in Identification and Offset IP Fields," Proc. 5th Int'l. School and Symp. Advanced Distributed Systems (ISSADS), Jan2005, pp. 118–25.

[11] G. Danezis, "Covert Communications despite Traffic Data Retention", tech. rep., ESAT, University of Leuven, Jan.2005, http://homes.esat.kuleuven.be/ gdanezis/cover.pdf.

[12] C. G. Girling, "Covert channels in LAN's," IEEE Transactions on Software Engineering, vol. SE-13 of 2, February 1987.

[13] R. A. Kemmerer, "Shared Resource Matrix Methodology: an IEEE Communications Surveys & Tutorials • 3rd Quarter 2007 57 Approach to Identifying Storage and Timing Channels," ACM Transactions on Computer Systems (TOCS), vol. 1, no. 3, Aug. 1983, pp. 256–77.

[14] T. Handel and M. Sandford, “Hiding Data in the OSI Network Model,” Proc. 1st Int’l. Wksp. Information Hiding, 1996 pp.23–38

[15] M. Wolf, “Covert Channels in LAN Protocols,” Proc. Wksp. Local Area Network Security (LANSEC), 1989, pp. 91–101.

[16] Grzegorz Lewandowski, Syracuse University, Network-aware Active Wardens in IPv6, 2011.

[17] Georey Ackerman, Daryl Johnson, and Bill Stackpole, Covert channel using icmpv6 and ipv6 addressing, in Proceedings of the International Conference on Security and Management (SAM). 2015, p. 63, The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).