ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

A Review of Literature on Design and Detection of Network Covert Channel

Nishant D. Rohankar, A. V. Deorankar, Dr. P. N. Chatur
M. Tech, Govt. College of Engg, Amravati, Asso. Professor, Govt. College of Engg, Amravati, HOD, Computer Science & Engg. Department, Govt. College of Engg, Amravati

Abstract —A network covert channel is a mechanism that can be used to leak information across a network in violation of a security policy and in a manner that can be difficult to detect. Covert channels exist in most communication systems and allow individuals to communicate truly undetectable and exchange hidden information. These are used for the secret transfer of information. Encryption only protects communication from being decoded by unauthorized parties, whereas covert channels aim to hide the very existence of the communication. Initially, covert channels were identified as a security threat on monolithic systems i.e. mainframes. More recently focus has shifted towards covert channels in computer network protocols. The huge amount of data and vast number of different protocols in the internet seems ideal as a high-bandwidth vehicle for covert communication. Trapdoors are unintended design with a communication system that exists in network covert channels as a part of rudimentary protocols. Again, many applications of covert channel are of malicious or unwanted nature and therefore pose a serious threat to network security. So, there is need to detect covert channels. This paper presents a survey of network covert channel design and their detection.

***Keywords*— Covert Channel, Detection, Network Security, Trapdoors**

I. INTRODUCTION

The first definition of covert channel was given in [1]. A covert channel is a mechanism that can be used to violate a security policy by allowing information to leak to an unauthorized process [2]. The main characteristic of a covert channel is the aim to hide the fact that a transmission is taking place. Compare this with cryptography where the goal is to transfer data readable only by the receiver. In cryptography there is no intention to camouage the communication. A covert channel may encrypt the data sent through it, but it mainly seeks to disguise its transmission. Steganography is the oldest form of covert channel. It is the act of embedding a secret message within a larger message so that others cannot discern the presence or contents of the secret message. In the cyber world, steganography specifically means hiding information in text or multimedia files (like image or video files) in a way that is unnoticeable by an average user. Only special expertise and tools may lead to its detection. Statistical analysis is one way to detect this type of steganography. A diverse range of individuals and groups has found reason to utilize covert channels for communication and coordination. Whenever an unauthorized access is gained, the next typical intruder's aim is to obtain information from the compromised host (e.g. password files, private documents, cookies etc.) and to sends there. In order to hide the fact of data transmission by malicious software, covert channels can be applied. The general idea of covert channels relies on the idea that information can be transferred in unused fields of network protocols or it can change any uncritical data in a network protocol. Many programs for creation of covert channels are developed. It should be emphasized that often even ordinary employees may want to use covert channels to bypass their company firewalls in order to access internet resources. Furthermore recent attempts by some governments to limit the freedom of speech in the Internet have led to proposals for using covert channels to circumvent these measures [2, 3]. In countries that forbid (strong) encryption of data, covert channels can be used to secure the information transport (although this is not strong security in the cryptographic sense).Network administrators can use covert channels to secure network management related communication by hiding it from hackers [4]. Again this is not strong security in the cryptographic sense. Honeypots, which are computer systems set up as trap for hackers, can also use covert channels to export logged data in real-time hidden from the attacker [5]. Since security analysts first started thinking about covert channel communication, two terms have been introduced, storage and timing channels. A storage channel involves the direct or indirect writing of a storage location by one process and the direct or indirect reading of the storage location by another process. The storage resource [such as unused bits in a packet header or the padding fields in a data-gram] is shared between the two subjects. In a way steganography can be seen as a form of storage channel. A timing channel involves a sender process that signals information to another by modulating its own use of system resources (e.g., CPU time) in such a way that this manipulation affects the real response time observed by the second process. Specifically, this can be done by modulating the wait time between packet transmissions (the inter-packet delays).In the

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

following section, we provide information on survey of covert channels. We present work done on design and detection of storage and timing channel.

II .NETWORK COVERT CHANNELS

A. *Storage channel: Design and Detection*

Many tools exist for setting up network covert channels using a variety of protocols including TCP, IP, HTTP and ICMP. The data section of packets is the easiest place to convey covert information, due to its easiest place to convey covert information, due to its large size and because it is relatively unstructured compared to headers. Modifying the packet payload is outside the scope of this paper as it falls in the realm of steganography or watermarking. Covert channels can be encoded in *unused or reserved bits* of frame or packet headers. Unused header fields that are either designed for future protocol improvements or in general go unchecked by firewalls and network intrusion. This is particularly problematic if protocol standards do not mandate specific values or receivers do not check for the standard values. Handel proposed a covert channel using the unused bits of the IP header's Type of Service (TOS) field or of the TCP header's Flags field [6]. Kundur suggested using the IP header's Don't Fragment (DF) bit as a covert channel [7]. The DF bit can be set to arbitrary values if the sender knows the Maximum Transfer Unit (MTU) size of the path to the receiver and only sends packets of less than MTU size. Hintz proposed transmitting covert data in the TCP Urgent Pointer (used to indicate high priority data) that is unused if the URG bit is not set [8]. Wolf proposed covert channels in header fields of multiple (now obsolete) protocols such as Token Ring [9]. Lucena identified a number of covert channels in various IPv6 header fields such as Traffic Class and Flow Label [10]. The *IP Identification (ID)* header field is used for reassembling fragmented IP packets. The only requirement from the IP standard is that each IP ID uniquely identifies an IP packet for a certain time period [11]. The Fragment Offset is used to determine in which sequence the fragments need to be reassembled. Rowland proposed multiplying each byte of the covert information by 256 and directly using it as the IP ID [12]. Ahsan proposed transmitting covert information in the high eight bits of the IP ID (as the XOR of the data and a secret key) and generate the low eight bits randomly [13]. TCP sequence numbers are used to coordinate which data has been transmitted and received guaranteeing reliable transport. The first sequence number selected by the client is called the *Initial Sequence Number (ISN).* The ISN must be chosen such that the sequence numbers of new incarnations of a TCP connection do not overlap with the sequence numbers of earlier incarnations of a TCP connection [14]. Rowland outlined an indirect channel called the bounce channel (Fig. 1). Instead of sending the ISN directly in a TCP SYN packet to the receiver, the sender sends the TCP SYN packet to a bounce host with a spoofed IP source address set to the intended destination. Upon reception of the SYN packet the bounce host sends a SYN/ACK or SYN/RST to the receiver with the acknowledged sequence number equal to the ISN+1. The receiver decrements the ACK number and decode the hidden information. Rutkowska developed a covert channel based on TCP ISNs for Linux, where the covert information is encrypted in the ISN fields so that the resulting distribution is uniform random [15]. However, Murdoch and Lewis pointed out that all the previous proposed ISN techniques produce a different distribution than the real operating system implementations [16]. They developed ISN covert channels tailored for Linux and Open BSD, where the ISN distribution of the covert channel looks like the normal ISN distribution. Jones et al. proposed a covert channel based on the IP header *Time to Live (TTL*) field as solution to trace back IP flows without using the source address field [17]. In their approach, routers modulate the TTL field of packets so that downstream receivers can unambiguously identify their upstream router. The covert channel is used for marking instead of general purpose communication. Qu and Lucena proposed techniques for embedding covert information into the TTL [18] and the IPv6 Hop Limit field (IPv6 equivalent of the IP TTL). Neither scheme takes into account typical initial TTL values chosen by the sender and normal TTL variation occurring in networks. Zander et al. analyzed initial TTL values and normal TTL variation and proposed an improved covert channel encoding in the TTL field that is harder to detect [19].Application level protocols can also be used as a carrier for covert data. Bauer proposed using covert channels in web traffic of uninvolved users to enlarge the set of users and improve the degree of anonymity.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

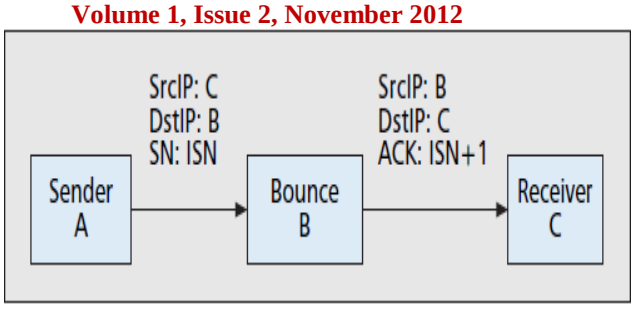

Fig.1 The TCP Initial Sequence Number (ISN) Bounce Channel

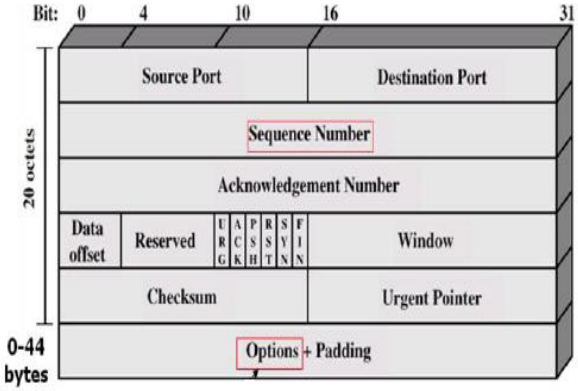

Fig. 2 TCP Header

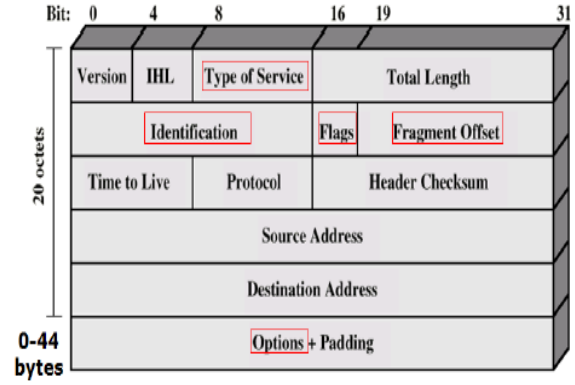

Fig. 3 IP Header

The information is hidden in JavaScript/HTML and transported through the use of JavaScript redirects. An observer who cannot look into the content transported by HTTP cannot distinguish between harmless web surfers and the covert senders/receivers. Feamster *et al.* proposed infranet i.e a framework to use covert channels in HTTP to circumvent censorship [20]. Web servers participating in infranet receive covert requests for web pages encoded as a sequence of HTTP requests to harmless web pages and return the content hidden inside harmless images using steganography. Bowyer proposed a very similar mechanism to communicate with Trojans behind firewalls [21]. A Trojan on the compromised system sends HTTP requests to a web server, with the covert data encoded as URL parameters. The web server returns innocent looking web pages with images that contain hidden data (steganography). Dyatlov, Castro, Kwecka and Van Horenbeeck proposed various methods for embedding covert channels into HTTP protocol headers [22–24]. These encompass encoding covert data into header field values, the order of header fields, the use of lower or upper case, the presence or non-presence of optional header fields, the use of multiple white spaces, and new nonstandard header fields. More recently Castro *et al.* have also developed a method for transmitting covert information through HTTP cookies [25].Eugene Tumoian and Maxin Anikeev make use of NUSHU, a proof of concept tool for TCP/IP passive covert channel creation. NUSHU for Linux is one of the newest tools for covert channel implementation. It has been developed by Joanna Rutkowska. NUSHU is a proof-of-concept tool for Linux using 2.4 or 2.6 kernel. It provides a passive covert channel (PCC), i.e. it does not create new traffic, but uses data of the existing one. Figure 2 and figure 3 illustrates the possible ways of creating covert channels in TCP and IP headers. In TCP

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

header, fields like sequence number and options can be used to place covert data. Similarly in IP header, fields like Type of Service, Identification, flags, fragment offset, and options can be used to place covert data. It is generally admitted that covert channels can not be completely eliminated. Although it could be appreciably reduced by design and careful analyzes. *Detection theory* [26] for covert channels embedded in various protocols is relatively a new area of research. Covert Channel detection is to actively monitor the illegal information flow or covert channel in the network stream. Covert Channel Identification is to identify a couple of resources used for covert channeling, especially this happens in the case of storage based covert channels. Focus here is on active monitoring the malicious activity on the network stream and not the identification of resources. Various authors across the globe have categorized detection into three important categories - *Signature Based Detection* - It involves searching specific pre-defined patterns in the network stream and when same appears, trigger alarm process. Best example for kind of channel it detects is Net Cat which is a reverse-shell communication between the internal network and a public network. *Protocol Based Detection* - It involves searching the protocols for anomalies or violations while monitoring the network stream. This requires understand the protocol specification described in their RFC's and detector must be knowledgeable to scans covert vulnerable fields in the protocol header. The best example for channel that can be found is Covert TCP tool which manipulates sequence number field in TCP and IP identification in IPv4 packet for the covert communication. *Behavioral Based Detection* - It involves creation of user profiles and reference profiles with respect to network stream in a legitimate environment. These reference profiles are later applied to the production environment for lateral comparison of real time user profiles with reference profiles. Best instance is writing arbitrary data in any packet using steganographic techniques. Senda Hammouda, Lillia Maalej, Zouheir Trabelsi[27] proposed TCP/IP covert channel detection system. They analyze various fields in TCP/IP header and detect possible covert channels. In IP protocol header,during checking of *Covert channels using the "Type of Service" field*, it is considered that the use of this field would be suspicious when it is set to a non null value. It is also mention that the bits 6 and 7 in this field are specified to be set to zero. So to detect a non legitimate use of this field, it is necessary to check the value of these fields. *Covert channel using the "IP identification" field*-the values of this field are generated randomly, by the TCP/IP stack. But for packets belonging to the same connection and the same flow, the "IP identification" field is incremented by 1. Thus, the detection of this particular hidden channel could be made by saving packets of various connections, calculating the difference between the IP IDs values of consecutive packets of each connection and test if it is equal to 1. *Covert channels using the IP options: "Strict source routing", "Loose source routing", "Strict source routing" and "Record route*-First of all, these IP options require contents of IP addresses. So to detect covert channels using the IP options, it is required to check the validity of the inserted IP addresses in the IP options. In TCP protocol header, during checking of *Covert channel using the "Urgent Pointer" field*-the "*Urgent Pointer*" field is interpreted only if the URG bit is set. Thus, if this bit is not set, the "Urgent Pointer" field would have null content. On the other hand, if the URG bit is not set and this urgent pointer field contains a non null value, it would be the detection of a covert channel. Eugene Tumoian, Maxim Anikeev [28] provides a way to detect covert channel created by Initial Sequence Number (ISN). They were made use of NUSHU tool for ISN generation. They collect ISNs generated by the standard stack. The collected ISNs form a training set. The ISNs generated by standard stack are considered as normal ISN. These are different from the ISNs generated by NUSHU tool. Fig.4 shows the neural network training.

Neural network which act as a classifier is trained for normal ISNs and the ISNs generated by NUSHU tool, so that, it will classify normal ISNs and the ISNs generated by NUSHU tool during testing. Whenever the training is completed, SYN-packets are being intercepted in the controller network (other packets are not needed). The neural network tries to predict the successive ISN. As soon as the current ISN is also intercepted, it can be compared with the predicted value. Similarity measure between the two values characterizes how well the network data matches the constructed model. If the difference is higher than the chosen threshold, it will consider that the ISN was not generated by the original stack. Hamming distance between the two binary numbers is chosen as a similarity measure. This process is shown on Fig.5. So, the ISN which deviates from normal ISN will be easily considered as a result of covert channel.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

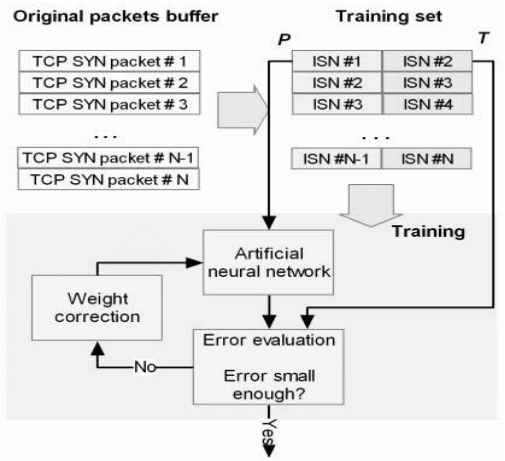

Fig.4 Neural Network Training

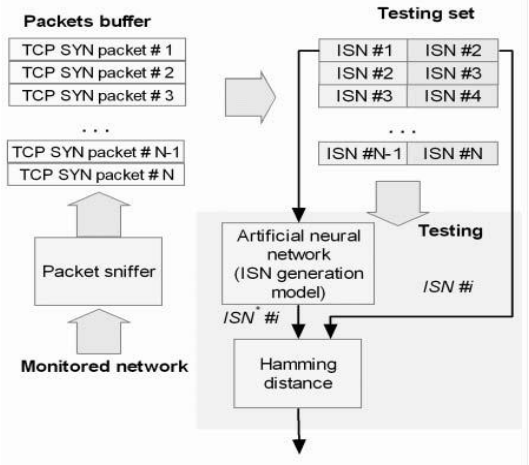

Fig.5 Similarity Measure Calculation

B. Timing Channel: Design and Detection

Less attention has been placed on network timing channels. These channels convey information through the arrival patterns of packets, rather than through the contents of the packets themselves. There are two types of covert timing channels: active and passive. In terms of covert timing channels, active refers to covert timing channels that generate additional traffic to transmit information, while passive refers to covert timing channels that manipulate the timing of existing traffic. In general, active covert timing channels are faster, but passive covert timing channels are more difficult to detect. On the other hand, active covert timing channels often require a compromised machine, whereas passive covert timing channels, if creatively positioned, do not. Network timing channels include packet sorting channels [29, 30], in which the order of packet arrival conveys information, and timing channels in which the reception or absence of packets within specific time intervals carries significance. Serdar Cabuk, C. Brodley[31] developed first IP covert timing channel. In a timing channel, the receiver and sender agree a priori on a timing interval and the starting protocol (either a particular time or in response to a network event, such as the first packet sent). During each time interval the sender either transmits a single packet or maintains silence. The receiver monitors each interval to determine whether a packet was received or not. The result is a binary code where a 1 represents the detection of packet in an interval and a 0 represents the absence of a packet .This process is shown in figure 6. The raw data that flows across the channel is binary but the actual interpretation of the binary stream is up to the communicating parties. Regardless of whether the binary stream is the data itself or whether it represents a message, additional bits are usually included in the transmission for three reasons. Firstly, additional parity bits may be appended to the data to add redundancy for error correction due to transmission errors (e.g., errors arising when a packet is lost). Secondly, additional bits may be added for purposes of maintaining synchronization between sender and receiver. Finally, the data may be encrypted in order to add a further layer of privacy and obfuscation. The message for transmission is subdivided into smaller blocks of binary data, referred to as frames. Frame consists of data bits, synchronization bits, and error-correcting bits. While all the frames are of equal length, the actual length, as well as the interval between frames, is influenced by parameters of the encoding scheme and the network. Cabuk [32] later designed a more advanced covert timing channel based on a replay attack, which is refer to as time replay covert timing channel(TRCTC). TRCTC uses a sample of legitimate traffic S_{in} as input and replays S_{in} to transmit information. S_{in} is partitioned into two equal bins S_0 and $_{S1}$ by a value $t_{cutoff.}$ TRCTC transmits a 1-bit by randomly replaying an interpacket delay from bin S_1 and transmits a 0-bit by randomly replaying an interpacket delay from bin S_0. Thus, as S_{in} is made up of legitimate traffic, the distribution of TRCTC traffic is approximately equal to the distribution of legitimate traffic. Gianvecchio et al. [33] developed an automated framework for building model-based covert timing channels, which is referred as MBCTC, to mimic legitimate traffic. MBCTC fits a sample of legitimate traffic to several models, such as Exponential or Weibull, and selects the model with the best fit. MBCTC then uses the inverse distribution function and cumulative distribution function for the selected model as encoding and decoding functions. MBCTC transmits by generating pseudorandom interpacket delays using the inverse transform method of variate generation to transmit hidden messages, i.e., messages are encoded using the inverse cumulative distribution function and decoded using the cumulative distribution function. Berk et al. [34] implemented a simple binary covert timing channel based on the Arimoto-Blahut algorithm, which computes the input distribution that maximizes the channel capacity.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

Sellke et al. [35] showed that with independent and identically distributed (i.i.d.) traffic as cover, it is theoretically possible to create "provably secure" covert timing channels, i.e., covert timing channels that are computationally non-detectable. The same basic proof can be used to show that TRCTC is computationally nondetectable for i.i.d. cover traffic when its input messages are XOR'd with cryptographically secure random numbers.

Detection scheme of covert timing channel concerned with detecting only the presence of a covert channel, and are not designed to infer the contents of such channels. There are two broad classes of detection tests: shape tests and regularity tests. The shape of traffic is described by first order statistics, e.g., mean, variance, and distribution. The regularity of traffic is described by second-order or higher order statistics, e.g., correlations in the data. Peng et al. [36] showed that the Kolmogorov-Smirnov test is effective to detect watermarked interpacket delays, a form of timing channel [37]. The watermarked interpacket delays are shown to have a distribution that is the sum of a normal and a uniform distribution. Thus, the Kolmogorov-Smirnov test can be used to determine if a sample comes from the appropriate distribution. The Kolmogorov-Smirnov test determines whether or not two samples (or a sample and a distribution) differ. The Kolmogorov-Smirnov test is distribution free, i.e., the test is not dependent on a specific distribution. Thus, the Kolmogorov-Smirnov test is applicable to different types of traffic with different distributions. In the Kolmogorov-Smirnov test, the distance between the test sample and the training set is measured that represents legitimate behavior. Thus, if the test score is small, it implies that the sample is close to the normal behavior. However, if the sample does not fit the normal behavior well, the test score will be large, indicating the possible occurrence of a covert timing channel.

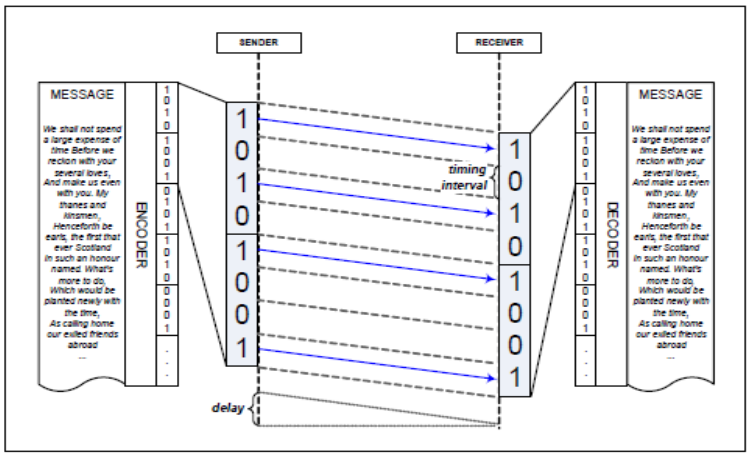

Fig. 6 IP Covert Timing Channel. The Example Text is First Encoded with a Coding Scheme and then Bit By Bit Sent to the Receiving End. The Message is rebuilt By Decoding The Bit Stream

The Kolmogorov- Smirnov test statistic measures the maximum distance between two empirical distribution functions:

$$KSTEST = \max | S1(x) - S2(x) |,$$

where S1 and S2 are the empirical distribution functions of the two samples. Cabuk et al. investigated a method of detecting covert timing channels based on regularity. This detection method, referred to as the regularity test, determines whether or not the variance of the interpacket delays is relatively constant. This detection test is based on the fact that for most network traffic, the variance of the interpacket delays changes over time, whereas with covert timing channels, if the encoding scheme does not change over time, then the variance of the interpacket delays remains relatively constant. In the regularity test, the standard deviation of the normalized standard deviations of sets of 100 packets is measured. If the regularity score is low, then the sample is highly regular, indicating the possible existence of a covert timing channel. For the regularity test, a sample is separated into sets of w interpacket delays. Then, for each set, the standard deviation of the set σ_i is computed. The regularity is the standard deviation of the pairwise differences between each σ_i and σ_j for all sets i < j

$$regularity = STDEV(\frac{|\sigma i - \sigma j|}{\sigma i}, i < j, \forall i, j).$$

Cabuk investigated a second method of detecting covert timing channels, referred to as ε-similarity, based on measuring the proportion of similar interpacket delays. The ε-similarity test is based on the fact that IP covert timing channel creates clusters of similar interpacket delays at multiples of the timing interval. Steven and Wang proposed entropy based approach to detect covert timing channels. They calculate entropy, conditional entropy,

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

and corrected conditional entropy. The entropy test estimates the first-order entropy, whereas the corrected conditional entropy test estimates the higher order entropy. Entropy test score is calculated. If the entropy test score is low, it suggests a possible covert timing channel, because the sample does not fit the appropriate distribution. If the conditional entropy test score is lower or higher than the cutoff scores, it suggests a possible covert timing channel. When the conditional entropy test score is low, the sample is highly regular. When the conditional entropy test score is high, near the first-order entropy, the sample shows a lack of correlations.

III. CONCLUSION AND FUTURE WORK

In this paper, we have given an overview of covert channels in computer network protocols. We have given review of different mechanism for creating and detecting covert storage and covert timing channels. There are number of protocols that can be used as carriers to make covert storage channel. In covert timing channel there is a problem of synchronization. Both sender and receiver need to be synchronized. Therefore, most of the time storage channel is preferred. The work on the covert channel can be extended by considering different ways of creating covert channel. Instead of using single field in header consider multiple combinations so that it became strong covert channel and detecting such strong covert channel will be a major challenge. If it is possible to detect such channel, definitely it will help in securing future computer network.

REFERENCES

[1] Jonathan Millen, "20 years of covert channel modeling and analysis," IEEE Symposium on Security and Privacy, 1999.

[2] N. Feamster et al., "Infranet: Circumventing Web Censorship and Surveillance," Proc. 11th USENIX Security Symp., Aug. 2002

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

[4] D. V. Forte et al., "SecSyslog: An Approach to Secure Logging Based on Covert Channels," Proc. First Int'l. Wksp. Systematic Approaches to Digital Forensic Engineering, Nov. 2005, pp.248–63

[5] The Honeynet Project, "Know Your Enemy: Sebek —A Kernel Based Data Capture Tool," tech. rep.,v 2003,http://www.honeynet.org/papers/sebek.pdf

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

[7] D. Kundur and K. Ahsan, "Practical Internet Steganography:Data Hiding in IP," Proc. Texas Wksp. Security of Information Systems, Apr. 2003.

[8] A. Hintz, "Covert Channels in TCP and IP Headers," 2003,http://www.defcon.org/images/defcon-10/dc-10-presentations/dc10-hintz-covert.ppt

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

[10] N. B. Lucena, G. Lewandowski, and S. J. Chapin, "Covert Channels in IPv6," Proc. Privacy Enhancing Technologies (PET), May 2005, pp. 147–66.

[11] J. Postel, "Internet Protocol," RFC 0791, IETF, Sept. 1981. http://www.ietf.org/rfc/rfc0791.txt

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

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

[14] J. Postel, "Transmission Control Protocol," RFC 0793, IETF,Sept. 1981, http://www.ietf.org/rfc/rfc0793.txt

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

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

[17] E. Jones, O. Le Moigne, and J.-M. Robert, "IP Traceback Solutions Based on Time to Live Covert Channel," Proc. 12th IEEE Int'l. Conf. Networks (ICON), Nov. 2004, pp.51–57

[18] H. Qu, P. Su, and D. Feng, "A Typical Noisy Covert Channel in the IP Protocol," Proc. 38th Annual Int'l. Carnahan Conf. Security Technology, Oct. 2004, pp. 189–92.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

[19] S. Zander, G. Armitage, and P. Branch, "Covert Channels in the IP Time To Live Field," Proc. Australian Telecommunication Networks and Applications Conf. (ATNAC), Dec. 2006.

[20] N. Feamster et al., "Infranet: Circumventing Web Censorship and Surveillance," Proc. 11th USENIX Security Symp., Aug. 2002.

[21] L. Bowyer, "Firewall Bypass via Protocol Steganography," Sept. 2002, http://www.networkpenetration.com/protocol_steg.html

[22] A. Dyatlov and S. Castro, "Exploitation of Data Streams Authorized by a Network Access Control System for Arbitrary Data Transfers: Tunneling and Covert Channels over the HTTP Protocol," tech. rep., Gray-World, June 2003, http://gray-world.net/projects/papers/covert_paper.txt

[23] Z. Kwecka, "Application Layer Covert Channel Analysis and Detection," tech. rep., Napier University Edinburgh, 2006.http://www.buchananweb.co.uk/zk.pdf

[24] M. Van Horenbeeck, "Deception on the Network: Thinking Differently About Covert Channels," Proc. 7th Australian Info.Warfare and Security Conf., Dec. 2006.

[25] S. Castro and Gray World Team, "Cooking Channels," hakin9 Magazine (www.hakin9.org), May 2006, pp. 50–57.

[26] Description of Detection Approaches at http://grayworld.net/projects/papers/html/cctde.html

[27] Senda HAMMOUDA, Lilia MAALEJ, Zouheir TRABELSI, "Towards Optimized TCP/IP Covert Channels Detection, IDS and Firewall Integration", ESRGroups France, 2008, pp.1-5.

[28] Eugene Tumoian, Maxim Anikeev, "Network Based Detection of Passive Covert Channels in TCP/IP", Proceedings of the IEEE Conference on Local Computer Networks 30th Anniversary, 2005.

[29] Kamran Ahsan, "Covert channel analysis and data hiding in TCP/IP", Master's thesis, University of Toronto, 2000.

[30] Kamran Ahsan and Deepa Kundur. "Practical data hiding in TCP/IP". In Proc. Workshop on Multimedia Security at ACM Multimedia, December 2002.

[31] Serdar Cabuk, Carla E. Brodley, Clay Shields, "IP Covert Timing Channels: Design and Detection",ACM transaction information and system security,October,2004,pp.178-187.

[32] S. Cabuk, "Network Covert Channels: Design, Analysis, Detection, and Elimination," PhD dissertation, Purdue Univ., Dec. 2006.

[33] S. Gianvecchio, H. Wang, D. Wikesekera, and S. Jajodia, "Model-Based Covert Timing Channels: Automated Modeling and Evasion," Proc. Symp. Recent Advances in Intrusion Detection, Sept. 2008.

[34] V. Berk, A. Giani, and G. Cybenko, "Detection of Covert Channel Encoding in Network Packet Delays," Technical Report TR2005-536, Dartmouth College, Aug. 2005.

[35] S.H. Sellke, C.-C. Wang, and S. Bagchi, "TCP/IP Timing Channels: Theory to Implementation," Proc. IEEE Conf. Computer Comm., Apr. 2009.

[36] P. Peng, P. Ning, and D. Reeves, "On the Secrecy of Timing-Based Active Watermarking Trace-Back Techniques," Proc. IEEE Symp. Security and Privacy, May 2006.

[37] X. Wang and D.S. Reeves, "Robust Correlation of Encrypted Attack Traffic through Stepping Stones by Manipulation of Interpacket Delays," Proc. ACM Conf. Computer and Comm. Security, Oct. 2003.

AUTHOR BIOGRAPHY

Nishant D. Rohankar has received his B.E. degree from Bapurao Deshmukh College of Engineering, India in 2010 and doing his M.Tech in Government College of Engineering. He has published two national level papers in the field of pattern recognition, neural network and cryptography. His area of research interests includes Cryptography, Network Security.

A. V. Deorankar has received his M.E.degree in Electronics Engineering from Govt. College of Engineering, Amravati, India. He has published nine national level papers and seven international papers. He is also patent of one paper. His area of research includes Computer Network, Web Mining. Currently he is working as an Associate Professor at Govt. college of Enineering, Amravati, India.

P. N. Chatur has received his M.E. degree in Electronics Engineering from Govt. College of Engineering, Amravati, India and PhD degree from Amravati University. He has published twenty national level papers and fifteen international papers. His area of research includes Neural Network, data mining. Currently he is head of Computer Science and Engineering department at Govt. College of Enggineering, Amravati.