Indirect DNS Covert Channel based on Base 16 Matrix for Stealth Short Message Transfer

M.A. Ngadi, S.N. Omar, and I. Ahmedy

Faculty of Computer Science and Information Systems,
University Technology Malaysia, Skudai, Johor 81310, Malaysia

Abstract

Covert Channel are the methods to conceal a message in the volatile medium carrier such as radio signal and network packets. Until now, covert channels based on the packet length produce abnormal packet length when the length of the message is long. Abnormal packet length, especially in the normal network will expose the covert channels to network security perimeter. Therefore, it motivates the study to propose a new method based on reference matrix to hide the secret message in DNS request. Normal DNS request packet was collected from the campus network. The proposed packets length covert channel was compared with normal DNS request packets. The study found that the new purpose covert channels produce normal DNS packet length according to the campus network.

Keywords: *Covert channels, Packet length, DNS.*

1. Introduction

The encryption is a method where the readable messages are scrambled into unreadable messages through transformations or permutations traditionally or conventionally with a key to protect the information from being readable by unauthorized party [1]. However, encryption alone cannot protect the confidentiality of the message because the unreadable message will attract the attacker to attack the communication channel and try to decrypt the message [2]. Moreover, the encryption itself does not prevent the adversaries from detecting the communication pattern [2].Furthermore, in communication, the encryption itself raises suspicion and triggers further investigation action [3] and knowing there exists a communication between two parties is already valuable information to the attacker [4]. Therefore, these motivate the study to find a method in network communications, where the secret message can be delivered without using an encryption on the network layer up to the application layer and at the same time, preventing the adversary from detecting the communications. It resembles Steganography, where the message is written on a piece of wood and then, waxes the surface of the wood to cover up the message. In network communications, the act of hiding the message in network protocol or communications is called covert channels [5].

Covert Channels are the desirable choices to send secret message based on the stealthiest and volatile of the packets. The stealth is possible to achieve because there are fields in the packet where the characteristics of the fields are random unused or not symmetrically controlled between the network devices within the network where the packets travel; known as Storage Covert Channels [6]. There is also a technique where the data could be hidden between the time arrivals of the packets, known as Timing Covert Channels [6]. The types of Covert Channels depend on the network security perimeter control between the sender and receiver and the stealthiest of the Covert Channels. The later is more preferable because it can resist some security perimeter. Additionally, the volatility of the packet's leverage is the stealthiest against Steganography without leaving any trail to be audited, because the packets will be destroyed after being used or processed according to defined criteria, which further motivates the use of covert channels [5,7]. Moreover, the motivations to use covert channels is supported with the quantity of data that can be transferred through covert channel annually, which can be as huge as 26Gb of data, although the data being transferred is only one bit at a time.8]. Therefore, there is no reason to doubt the capability of the Covert Channels in sending secret messages over the networks.

On the other hand, what makes the covert channels useful is the stealth of the covert channel [9]. The property of the stealth of the covert channels is attributed from the anonymity of covert channels as described in [10], which is subjected to three pillars; plausibility, undetectable and indispensability. Plausibility means the covert channel must be able to exploit the medium in which the packet is in use by the adversary. Undetectable means the amount of the bits sent should not violate the distribution of the normal packet. Indispensability means the adversary must use the medium and will not block the medium on the security perimeter.The indispensability is the most important aspect in stealth property because it will ensure

whether the proposed covert channel is useful or verse versa.

Plausibility and undetectably are co-related. To be truly plausible the covert channels cannot just exploit the packet's unused fields or the randomness of the packets without fulfilling the symmetry of' the packets as in [10,11,12]. In TCP packets, an independent packet could be seen as unfinished TCP handshake activity [13]. In literal meaning, how the data hidden in the packets and the packets operate have a strong correlation with the stealth of the Covert Channel [4,14,15]. Basically, there are two methods to hide and retrieve the data with Storage Covert Channels; indirect and direct methods. Indirect technique hides and retrieves the data by substituting the symbols with the property of the packets whereas the direct technique directly embeds the data into the field of the packet [16]. However, there are ambiguities in the definition of Indirect hidden method.

Hitherto, the direct covert channels have been the favorites and target research in the covert channel's research because the protocol fields to be manipulated are tangible within defined characteristic, based on the survey in [17].

Conversely, direct embedding is not always covert because little attention has been addressed to the communication in the context of the internet [9]. The effect of embedding the data directly into the innocuous protocol fields is it does not hide the fields from traffic analysis, which could map the connection structure to retrieve the uncovered information [9]. Moreover, direct embedding of the data in protocol fields will not be able to resist the network security approach known as the protocol scrubber. Protocol scrubber is a method of an active interposition mechanism to homogenize the network flows by identifying and removing the malicious content in the traffic flow through normalizing the protocol headers, padding and extensions [18]. Therefore, with the sophisticated protocol scrubber mechanism, the direct hidden method could satisfy plausibility of the covertness. However, it could not satisfy the indispensability of covert channels. This, further, forces and motivates the study to look into the indirect hidden method.

Looking back at the previous hidden method, the study is keen on indirect hidden method based on the packet lengths because the packet is directly under the control of the sender. Therefore, there are no malicious attempts to control other state property, and it is not prone to protocol scrubber. However, there are a couple of problems with previous packet lengths. First, the packet length is hard to implement because it could only work in a controlled environment with end-to-end connections as the lengths of the packets depend on the MTU of the routers in the packet travel across the path and the size of MSS. Secondly, as highlighted by Liping in [15], not only the MTU and control environment are the obstacles to packet lengths, but also the normal lengths distribution of the packet lengths.

At this juncture, indeed, the normal distribution of packet lengths is directly associated with the plausibility and undetectably of the covert channels. The plausibility of the packet length's covert channels could not only be achieved when the length of the packets is associated with a value starting from 1 to 256, because, as shown by Liping in [15], the length of the packets is not randomly distributed between 1 to 256. Moreover, there is a gap between the lengths in normal distribution, which is too odd for 256 lengths differently or randomly. Figure 1 is taken from [15], which shows the normal packet lengths distribution of about 2000 packets. The vertical line is the length. In fact, it is clear that, there are only about 20-30 different lengths among the 2000 packets, which will be appropriate for a covert channel with 256 different length distributions as in [19,20,21]. Therefore, the plausibility of the exploited lengths is not convincing in [19, 20, 21] when compared to the normal packet length distribution.

Certainly, if the plausibility of the exploited lengths is not fulfill, the undetectably could not be satisfied. Liping in [15], proposed a method which used a reference of the lengths to overcome the trouble with too many packet length's distributions as in [19,20,21]., The Liping's model used sixteen different length based on the baseline length of the reference agreeable between the sender and receiver. This means, for every 4 bits of the data, there is additional of at least between one to sixteen bytes on the normal packet length. The problems become worse when the sender and receiver have to update the reference length to contain long messages. It would be noticeable that when the size of the message is huge, the Liping's method turns out from normal length distribution to abnormal length distribution as mentioned in [22]. Therefore, the Liping's method cannot satisfy the plausibility of the distribution of the packet length when the message is too long.

This is another challenge that this research would like to address. With the above problem, this research proposed a method, that whereby, there is no additional length of packets that needs to be added to deliver the message as in Liping's method, and the association of the packet lengths is not subject to one to one association as in [19,20,21] methods. The proposed method will allow an association of one to many. That is , the proposed method will introduce a reference, instead to the length of packet. It

will also, take the data in the packet payload as a reference. With this, it will not produce an odd packet length's distribution and there is no additional packet length to be added, therefore, it could resist against abnormal packet length, which resulted into the ability to deliver a long message within the distribution of the length on the selected packets. Therefore, with the proposed method, this research, is the first, to associate the length of the message with contain in the payloads and is the first, able to send a long message without adding additional lengths and within the normal packet length's distribution.

2. Previous Work

Packet length is classified as an indirect hidden method because there is no direct modification done to the packets except the data is hidden based on the length of the packets. Padlipsky introduced a packet length covert channel in [19]. Padlipsky associated the length of link layer frames with a symbol to conceal the secret message. Ten years later, Girling demonstrated Padlipsky idea in [20]. Girling represented the length of the link layer with 256 symbols. Which, it required 256 different packet lengths, and each packet length represents bytes of information. The experiment was done in the isolated network to eliminate the noise of other packets such as buffering, reblocking and spurious message insertion from high level protocols. In real networks, the controls of block size and packet length are actually being modulated and depend on the network conditions [23]. Conveniently, Padlipsky method could be very effective within the same network segment [24].

Two decades later, Yao and Zhang in [21] used a secret matrix with 256 rows and randomly associated it with the length of packets. The arrangement of the length in the matrix will be transformed according to the agreement between the sender and receiver. The Yao and Zhang method has successfully improved the Girling. However, a study by Liping in [15] shown that, the randomly packet length will trigger the detection because it produces abnormal network traffic.

To overcome the abnormal network traffic, Liping in [15] proposed a method based on a reference of length. Liping's method required the sender and receiver to agree upon the length of the packets that the sender sent to the receiver. The agreeable length of several packets is set as a default reference. To send a secret message, the sender takes the byte of the message and adds it to the length of the reference. To get the byte of the message, the receiver will deduce the received length with the initial reference. However, as mention by Liping in [22], this method was

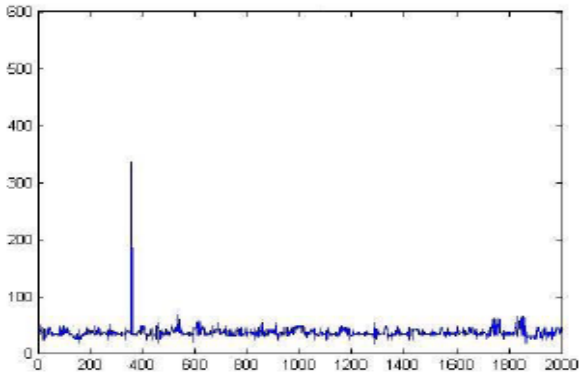

Figure 1: Normal packet length distribution [15]

not efficient when the size of the message is long because the method will update the default reference with every length it received. Therefore, when the message is too long, it will produce abnormal packet length's distribution.

Other noble works on storage covert channels are explained based on the protocol where the covert channels have been exploited.

In [13], Taeshik explained how the IP Identification (IP ID) field being exploited to embed ASCII alphabetic. The method multiplies the ASCII in hex value with 255, since 255x255 is 65535 which just fine to fix into 16 bits fields. However, the method use could trigger suspicious when the same letter in a word occurs. Then Ahsan in [25] improved the method using Toral Automorphism System that used pseudo random sequence to make sure the modified IP identification is random. Yogi Metta's in [6] proposed to exploit the IP ID fields by XOR the byte of the secret message with the IP's version and IP's header length, then, the result will be concatenated with a random number to cover the remaining 8 bits. However, as mention by Murdoch, covert channels with randomly the number in IP ID fields can be detected because it's not by default random [26].

Yogi Metta's in [6] theoretically explain how the value of DF could be use to send a message. The method could successfully implement if and only if we know the MTU of each router. Enrique's in [27] used the IP Offset field to embed the data. The only problems with IP offset field are when the DF is set and there is a data in IP Offset field. This would trigger and IDS or IPS. In [28] Zander demonstrated how TTL is manipulated to send a value 1 or 0. The TTL method is very suitable to send a small amount of data, except. It needs careful study on the variant of Operating System in the network because Fyodor mentions each Operating System use different TTL to identify them. Abad in [29] theoretically described how the Checksum value could carry the data, though; the Windows NIDS

layer will discard the packet if the checksum value is wrong. Moreover, the checksum value will change when the packet goes through the router or NAT.

RFC 792 portrayed Internet Control Message Protocol (ICMP) as a method to help and notified system when an error occurs somewhere in the network path. Most common uses of ICMP are ICMP type 0 (echo reply) and 8 (echo request), which is known as ping. These echo requests and replay could carry 56 bytes of data, and it is ubiquitous among an operating system. On August 1996, Daemon9's demonstrated the ICMP covert channel by exploiting the payload of ICMP type 0 and 8 with malicious data. This is possible because most of the firewall and network perimeter didn't check its payload contents and would allow it to pass [30]. Moreover, the payload of ICMP could carry an arbitrary data [31], therefore, there is numerous ICMP covert channels being exploited and published, such as: Loki [32], ICMP bounce tunnel [33], Ping tunnel [34] and 007Shell [30]. Latest, Zouher in [35] has used ICMP covert channel to send a file and message by exploiting the record router IP header. A lot of ICMP covert channel means the security professional should emphasize their security parameter to limits the ICMP packets. Nevertheless, ICMP will be a great Covert Channel in LAN but not on over the net. This is because most of the firewall will not allow inbound ICMP packet to enter their network [17].Unless used for outbound traffic, ICMP covert channel will be applicable.

Rowland in [11] had shown the basic of TCP covert channel by exploiting the TCP sequence number fields (32 bits). He used the same method as he did for IP Identification, just by multiples the 255x255x255xASCII value. This does not mean Rowland method is naïve, because the method presented is just to shows how the TCP sequence number could be exploited. Ahsan presented method that is more advanced later in [25], where the author encodes the secret message using Toral Automorphism Algorithm. Ahsan's divided the TCP sequence number into two 16 bits. The high 16 bits are used to embed the secret message while the lower 16 bit was generated by the random number. Rutkowska in [36] proposed a more robust method by encrypted the data and XOR it's with one-time-pad key. However, Murdoch specified that, in Rutkowska method. The TCP sequence number didn't exhibit the structure of the TCP ISN as expected in Linux and there is a flaw in the use of DES for encryption, which allows the recovery of the plaintext by statistical information [26]. Therefore, Murdoch later comes out with a more advanced method to show that the covert TCP sequence number look like real TCP sequence generated by Operating System. Murdoch's proposed a method where the data is encrypting with the block cipher that is running in counter mode, which produced different pseudo random sequence for each rekey interval. This 15-bits value than is inserted into the ISNs. The 16-bit field in ISN is set to zero and the rest 15 bits are generated by an RC4 pseudo random number generator [26]. Notes, TCP is the connection oriented therefore the stateful firewall can keep track the TCP state. The exploitation of TCP for covert channel over the net is not viable, unless in LAN because nowadays network perimeter firewall could keep track the TCP connections. Therefore, a single TCP sequence packet will look like a port scan packets.

Despite the concentration on TCP sequence number, Chan in [37], proposed a method call partial acknowledgment to exploit the TCP Acknowledgment number (TCP ACK). Their method is calls partial acknowledgment because the value of TCP ACK is less than ISN +1, as in normal TCP operations. To send a message, let say M, the partial acknowledgment number will be ISN + 1 – M. Therefore, to get back the message, the receiver need to get the value of M by subtract the next ISN+1 – ACK. However, this method is not efficient in the network environment with stateful firewall because the ACK number is less than the ISN + 1. Another problem is, for each secret message, they have to make sure, for each TCP packet, is set with the minimum size of MSS. This will result with a lot of TCP ACK between the sender and receiver despite the OS could handle TCP packets with more payloads.

UDP based on its design principle, is to exchange message with a minimum of protocol mechanism and session management. UDP is connectionless protocol. Therefore, there is no session control to make sure the packets reach the destination. There are few fields could be exploited on UDP for covert channels. The only possible covert channel field on UDP is the source port, and this would be applicable on LAN because the source port will be modified when the datagram goes through the NAT firewall. Conversely, UDP has been used to carry another internet protocol such as IP [38][39] and TCP (Simon) to evade the firewall. Thereby, the covert channel could exits on the protocol on top of UDP stack.

2.1 DNS

DNS or Domain Name System' is used to translate human-readable hostname to numerical IP address and vice versa [40]. In the design, DNS protocol was on top of UDP protocol. This gives advantages to DNS, which, there will be no overhead on the services resource and network perimeter, as there is no connection tracking or session to process. Moreover, there are fields in the DNS protocol, which allows huge bytes to be carried especially in the

query and response [41]. The researcher used DNS as a method to bypass the security perimeter] has exploited these advantages [41. Besides, until now, DNS is less filtered by the organization security perimeter, and this is further proof, when the captive portal allows the DNS to make a query to the internet, although the user hasn't authenticated to the network, which means, the DNS query is independent of the identity of the requestor. This further encouraged the used of DNS query and response, instead, as a simple method to bypass the firewall as a method to tunnel through a network. The used of DNS as a tunnel is the further study by Merlo in [41], which do the comparison on the performance of six DNS tunnels; NSTZ [39], DNSCat, Iodine, TUNS [42], Dns2TCp and OzyManDNS [38].

However, there is a major different between tunnel and covert channels. Generally, tunnel main intentions are just to bypass the security filtering and not to fade the communications [43]. Further, in tunneling, the clients and server have to keep tract of the connections between each other while the connection is active, which, results in high traffic between the communications node [42]. Moreover, in tunnels, the method used to carry their data is not to hide their data appearance as in covert channels. Therefore, the data is being encoded in non-compliant to Base64 encoding [41]. Albeit, the problem highlighted with the tunnels, is not to be highlighted that the tunnel is not good, but to support that, DNS is the good choice for covert channels because, as stated in [41], until now, DNS is the less filtered protocol, which means. It can be used for the purposed covert channels, which meet the indispensability property.

3. The DNS Reference Model

3.1 The step to send and received the message

The study subdivided the entire process into the method similar to OSI model for better understanding as follows:

• Level 0; starting from Alice's side, Clear Message (M) is the readable message that Alice wishes to send to Bob.

• Level 1; Alice's M is encrypted (Em) with a block cipher algorithm and stored the Em in the queue.

• Level 2; The indirect algorithm will associate the corresponding Cm with standard URL name.

• Level 3; The Sp will be injected into the network that will passes through the protection network as normal DNS query packets.

• Level 3; On the Bob side, all received datagram will be picking up and stored in memory stack.

• Level 2; Indirect Detection module will determine the correct Sp. The correct Sp will be processed and the byte of the message will be stored on the stack until the end of the flow control is found.

• Level 1; together with the Sk, the Decryption Decoder will decode Em recover the sent message.

• Level 0 if the Em is successfully decoded, Bob will be able to read the M which sent by Alice.

To be note; for the rationale of the SCCF as in Figure 1, the study assumed the follows conditions:

• Reasoning for the purpose of security. Cm is encrypted using Symmetric encryption algorithm. The Sk is only known to Alice and Bob.

• The objective of this DNS Covert Channel is to hide the secret messages in Sp through Indirect Algorithm.

• The process of the transmission Sp is in sequence order with ideal timing and overt network.

• In some situation, when Sp needs to travel across multiple networks; Sp must not be detected by other nodes. Therefore, Sp must not show any different between normal DNS packets against Sp DNS packets.

• assuming there is no Sp loss because of buffer unavailability or network congested.

3.2 The indirect reference algorithm

The stego method used in DSCCF is based on URL name to represent the Base 16 values. The URL hostname could be any agreeable hostname that is normally used in the network where the sender resides. The preferable solution is to choose the URL that normally requests by the clients in the network. Importantly, the DNS query's datagram should not exceed 300 bytes and 512 bytes for the response datagram as stated in [45]. Albeit, there are certain conditions, which allow the DNS query to specify the response datagram can exceed 512 bytes by using the OPT Resource Record [46]. However, as stated in [42], the length of the URL should not exceed 140 bytes.

The indirect reference module will process the cipher in block size of 4 bits. For each block, the module will find the corresponding Base 16 values. The row of the corresponding Base 16 is the current amount of block being processed. Notes that, the amount of blocks will be mod with 16. As a result, the value will be in between 0 to 15. This value is the row that process will find the position of Based 16 value. Once the value of Base 16 is found in row[n], the module will generate the DNS packets with corresponding URL name and store in stack. The process of covert the Em to corresponding URL name will end when all the byte in the Em has been processed. line.

Table 1

DNS Samples		A	B	C	Average Packet length (%)
Number of Packets		34607	398067	649962	
Time (Minutes)		280	64	121	
Length	40-79	64.82	51.42	50.55	55.6
	80-159	35.18	48.58	49.45	44.4
	160-319	0	0	0	0

4. The Experiment

The proposed indirect algorithm was tested based on the URL name collected from campus network. The proposed schema was compared with [20], [21] and [15] models.

4.1 DNS Dataset

The DNS dataset was collected on-campus network. The study used tcpdump to collect the packets. Table 1 shows the three different samples that have been captured on three different times.

Sample A has about 34,607 thousand standard DNS query packets with average packet's length in between 40 to 79 is about 64.82 percent and 80 to 159 with 35.18 percent. Sample B has about 398,067 thousand standard DNS query packets with average packet's length in between 40 to 79 is about 51.42 percent and 80 to 159 with 48.45 percent. Sample C has about 649,962 thousand standard DNS query packets with average packet's length in between 40 to 79 is about 50.55 percent and 80 to 159 with 49.45 percent.

The most important point with this data, the covert channels will have about 119 packets different lengths to operate and not to exceed 159 packet lengths. Otherwise, its will show an abnormality in the networks. Moreover, it could be tolerated to say that, it should be normal for the covert channels with plus or minus 5 percent of the average packet length in between 40 to 79 and 80 to 159.

4.2 Experiment Setup

The implementation of the proposed covert channel was done using the winpcap library to send and capture the standard DNS query. To capture the packets for the purpose to be validated, evaluated and monitor, the study used Ethereal and Wireshark. The message length of the cipher is 88 bytes. The message was encrypted using 256 block cipher algorithm with shared key. The Oracle Virtual Box was used to running the Windows XP operating systems with 100Mbps.

Table 2

Author's Model		Girling	LAWB	Liping	Propose
Number of Packets		88	88	194	178
Length	40-79	0	12	12.89	60.67
	80-159	58	60	80.41	39.33
	160-319	30	16	13	0

4.3 Experiment Result

In this section, the study discusses the findings based on two comparisons. First, the study compares the statistical of the DNS length to shows the percentages based on the group length as stated in the Table 1. Then, the study compares the distribution of the length as discuss in subsections 4.3.2. Lastly, the study presented the bandwidth that could be achieved.

4.3.1 Packet length comparison

Table 2 shows the results of Girling's, LAWB's, Liping's and propose the schema after successful sending the secret message to Bob.

4.3.1.2 Packet efficiency

Based on the number of DNS packets used to transfer the secret message, the results in Table 2 shown that, the Girling's and LAWB's method, only required 88 packets, which is better than propose and Liping's method. This is because, in Girling's and LAWB's method. The length of DNS packets is directly associated with ASCII's table value. The propose method is better than Liping's method because Liping's method required Alice's to send preliminary DNS packets to Bob as reference length.

4.3.1.2 Data transfer efficiency

Regarding the data-transfer efficiency, Girling's and LAWB was better than propose and Liping's because Girling's and LAWB method can carry one bytes per packets. The propose and Liping's method carries 4 bits of data per packet.

While Liping's and the propose method required more packets than Girling's and LAWB's method, this doesn't mean that Liping's and the propose method is not efficient. Without tempering the packets with secret data, propose and Liping's method was better than timing's covert channel with four bits of data per packet. In timing covert channel, each packet can only transfer one bit of

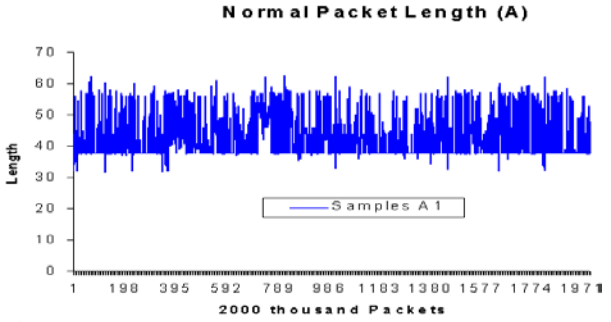

Figure 2

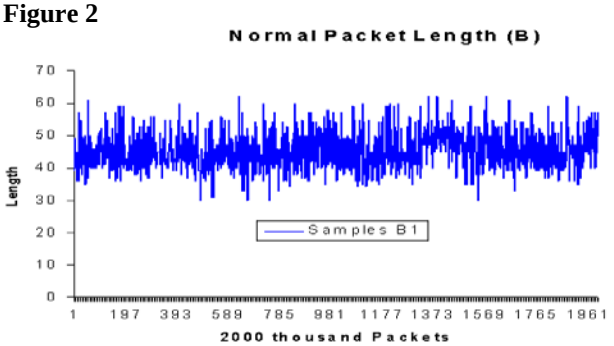

Figure 2

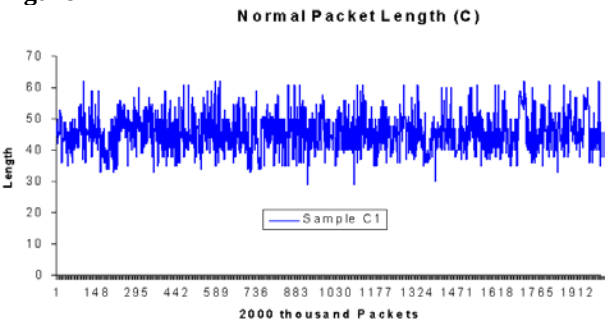

Figure 3

data, which, will require 512 packets to transfer 88 bytes of data.

4.3.1.3 Packet length percentage

The only method where the packet lengths are within the range of 40 to 159 lengths is the propose method. This is because, in the propose method, packet lengths are just a measurement to make sure the covert packets are within the normal packet lengths range. Unlike Girling's, LAWBs and Liping's, where the data was hidden based on the different range of packet lengths, which are bound to the availability of the range of packets in that particular protocol. Liping's method, in [15], based on their test bed HTTP data from Clarknet, was available with only 400 ranges of different HTTP lengths. The limited ranges of length are not limited to Liping's method. Girling's and LAWB was bounded to the same problem where their method could only be used with the range of 55 packet length. Albeit, in HTTP of Clarknet samples, there are 400 packet lengths range. Therefore, the propose method is better, because lengths, is not the limits.

4.3.2 Normal packet length distribution comparison

In normal distribution packet lengths, the study looks into the length of the UDP packets that used to envelop the DNS protocol. The analysis of normal length was done on three DNS dataset as stated in sections 4. Based on the samples, the study analyzes 2000 thousand and 200 packets of each sample and plots a normal distribution of the packet length's graph as in figure 4, 5,6,7,8, and 9. The normal distribution of packet lengths is based on 2000 packets is enough to show the distribution of packets based on a comparison done in [15], however, the 200 normal distribution of packet lengths is required to give the explanation for the results of the experiments based on 88bytes of a cipher message which required 196 packets of DNS to conceal the cipher message.

The normal distribution of packet lengths based on 200 hundred packets as shows in figure 7, 8, and 9 are accordingly with their respective 2000 packets length. Therefore, the two distributions based on 2000 thousand and 200 packets will develop to justify packet length's comparisons. The graph in Figure10 is the packet length's distribution that was plotted based on the result from the propose schema. The propose packet length distributions depict in Figure 10 was compared to the 2000 thousand and 200 hundred normal distributions. The study found that the propose result was normal than Girling's, LAWB's, and Liping's. The Girling, LAWB AND Liping schema is depicted in Figure 11,12, and 13. The result proof that the propose distribution's packet lengths are normal to the normal distributions. The normal distribution is shows in Figure 5 and 8. Therefore, propose indirect method based on Base 16 matrix has been successful.

They should be numbered consecutively throughout the text. Equation numbers should be enclosed in parentheses and flushed right. Equations should be referred to as Eq. (X) in the text where X is the equation number. In multiple-line equations, the number should be given on the last line.

4.3.3 Statistical Test

The study further analyzes to propose packet lengths and the three samples as shown in Table 1 with T-test. The T-test [47], is used to measure significant different in their distributions. The study analyzed the UDP packet lengths on each sample and plot the boxplots to get boundaries of the packet lengths.

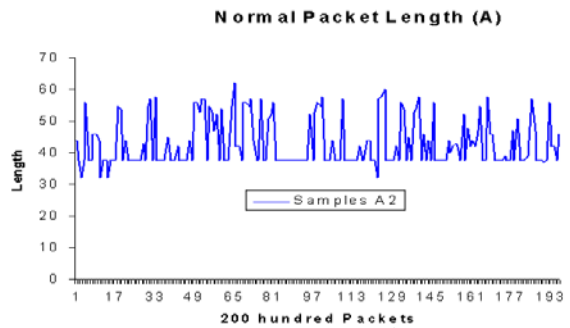

Fig 5: Normal Packet Lengths Distribution of Sample A2

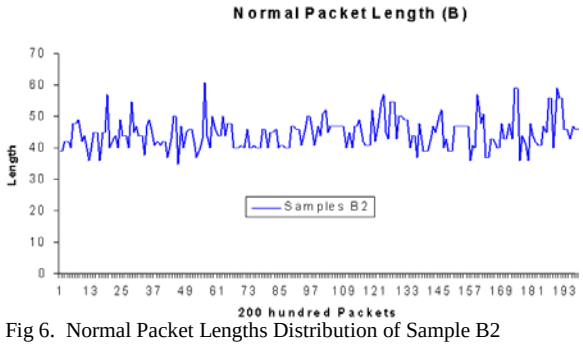

Fig 6. Normal Packet Lengths Distribution of Sample B2

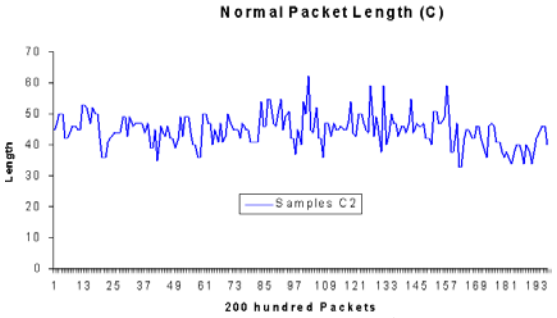

Fig 7. Normal Packet Lengths Distribution of Sample C2

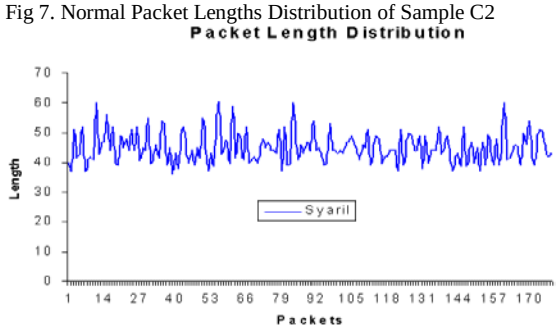

Fig 8. Propose Packet Length Distributions

Based on the boxplots in Figure 12, it could be concluded that the range of the packet length's D (Propose method) didn't same with the range of Packet lengths (A). Subsequently, the hypotheses for the relevant 2-tailed would be of the form:

H: Distribution of D packet lengths didn't same with A.
H: Distribution of D packet lengths is same with A.

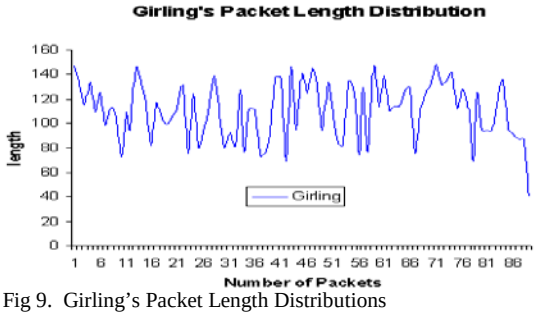

Fig 9. Girling's Packet Length Distributions

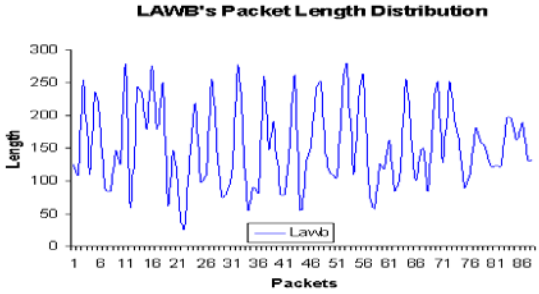

Fig 10. LAWB's Packet Length Distributions

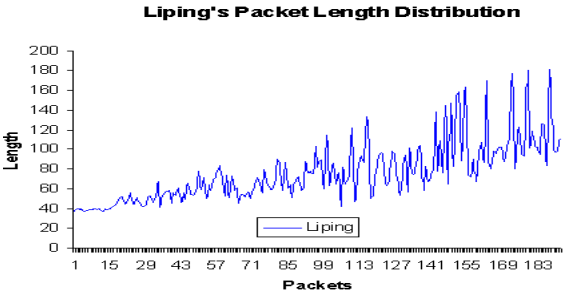

Fig 11. Liping's Packet Length Distribution

The study used Gnumeric statistical software to calculate the tstat, tcrit and p-value, leading to the following conclusions:

As we can expect from Figure 14, there is not enough evidence to reject that the distribution of B and C packet lengths is different from D.. Moreover, based on the p-value for the comparison between C and D, there is sturdy evidence to proof that the distribution of packet lengths between C and D are alike. No less, there is also a correlation between packet lengths for B and D.. Therefore, the t-test has proof that the propose packet lengths covert channel is normal with the campus's packet length distributions.

5 Conclusions

The novel indirect packet length covert channels has been proposed to generated normal packet length which

associated the payload of the packets to more than one symbols as done by previous packet length covert channels. This is done by using a reference matrix of Based 16. The experiment results shows that the covert channel has able to sustain in the upper bound of the average normal DNS packet lengths taken from the campus network and was normal in the packet length distributions which resist the covert channels against abnormal packet lengths observation..

Acknowledgments

This work is supported by National Science Fellowship of Malaysia. Special thanks to Prof Md Dr Mohd Asri Bin Ngadi, Ismail Hamedy and Mohd Nasri Mat Isa.

References

[1] L. Frikha and Z. Trabelsi, "A new covert channel in WIFI networks," Risks and Security of Internet and Systems (CRISIS 08), 2008, p. 255–260.
[2] S. Hamdy, W. El-Hajj, and Z. Trabelsi, "Implementation of an ICMP-based covert channel for file and message transfer," 2008, p. 894–897.
[3] G. Armitage, P. Branch, and S. Zander, "Covert channels in multiplayer first person shooter online games," 2008, p. 215–222.
[4] A. El-Atawy and E. Al-Shaer, "Building covert channels over the packet reordering phenomenon," INFOCOM, 2009, pp. 2186-2194.
[5] A. Desoky, "Listega: List-based steganography methodology," International Journal of Information Security, vol. 8, 2009, pp. 247-261.
[6] Y. Mehta, "Communication over the Internet using Covert Channels," 2005.
[7] J. Lubacz, W. Mazurczyk, and K. Szczypiorski, "Vice over IP," IEEE Spectrum, vol. 47, 2010, pp. 42-47.
[8] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating Steganography in Internet Traffic with Active Wardens," Information Hiding, 2002, p. 22.
[9] G. Danezis, "Covert Communications Despite Traffic Data Retention," Security Protocols XVI, B. Christianson, J.A. Malcolm, V. Matyas, and M. Roe, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 198-214.
[10] J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, "Covert Messaging through TCP Timestamps," vol. 2482, 2003, pp. 194-208.
[11] C. Rowland, "Covert channels in the TCP/IP protocol suite. Tech. rep., First Monday," ACM Transactions on Information and Systems Security, vol. 12, 1997, p. Article 22.
[12] S.J. Murdoch and S. Lewis, "Embedding Covert Channels into TCP/IP," The 7th Information Hiding Workshop, 2005, pp. 247-261.
[13] T. Sohn, J. Seo, and J. Moon, "A Study on the Covert Channel Detection of TCP/IP Header Using Support Vector Machine," In Proceedings of the International Conference on Information and Communications Security, 2003, pp. 313-324.
[14] N. Chen, W. Hu, and Z. Xue, "Research of covert channels based on web counters," Shanghai Jiaotong Daxue Xuebao/Journal of Shanghai Jiaotong University, vol. 42, 2008, pp. 1678-1681.
[15] L. Ji, W. Jiang, B. Dai, and X. Niu, "A Novel Covert Channel Based on Length of Messages," International Symposium on Information Engineering and Electronic Commerce, IEEC 2009, 2009, p. 551–554.
[16] H. Khan, M. Javed, S.A. Khayam, and F. Mirza, "Designing a cluster-based covert channel to evade disk investigation and forensics," Computers and Security, vol. 30, 2011, pp. 35-49.
[17] S. Zander, G. Armitage, and P. Branch, "A Survey of Covert Channels and Countermeasures in Computer Network Protocols," Communications Surveys & Tutorials, IEEE, vol. 9, 2007, pp. 44-57.
[18] P.A. Gilbert and P. Bhattacharya, "An approach towards anomaly based detection and profiling covert TCP/IP channels," Proceedings of the 7th international conference on Information, communications and signal processing, Piscataway, NJ, USA: IEEE Press, 2009, pp. 695-699.
[19] M.A. Padlipsky, D.W. Snow, and P.A. Karger, "Limitations of End-to-End Encryption in Secure Computer Networks," Tech. Rep., vol. ESD-TR-78-, 1978.
[20] C. Girling, "Covert Channels in LAN's," Software Engineering, IEEE Transactions on, vol. SE-13, 1987, p. 292–296.
[21] Q. Yao and P. Zhang, "Coverting channel based on packet length," Computer Engineering, vol. 34, 2008.
[22] L. Ji, H. Liang, Y. Song, and X. Niu, "A normal-traffic network covert channel," CIS 2009 - 2009 International Conference on Computational Intelligence and Security, Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, China: 2009, pp. 499-503.
[23] G. Armitage, P. Branch, and S. Zander, "Error probability analysis of IP Time To Live covert channels," 2007, p. 562–567.
[24] A. Epliremides and S. Li, "A network layer covert channel in ad-hoc wireless networks," 2004, p. 88–96.
[25] K. Ahsan and D. Kundur, "Practical Data Hiding in TCP/IP," Proceedings of the Workshop on Multimedia Security at ACM Multimedia, 2002, pp. 63-70.
[26] S.J. Murdoch and S. Lewis, "Embedding covert channels into TCP/IP," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3727 LNCS, 2006, pp. 247-261.
[27] E. Cauich, R. Watanabe, C. Science, and A. Zaragoza, "Data Hiding in Identification and Offset IP fields," In Proceedings of 5th International School and Symposium of Advanced Distributed Systems (ISSADS), 2005, pp. 118-125.
[28] S. Zander, G. Armitage, and P. Branch, "Covert Channels in the IP Time To Live Field," In Proceedings of Australian Telecommunication Networks and Applications Conference (ATNAC), 2006.
[29] C. Abad, "IP Checksum Covert Channels and Selected Hash Collision," Technical report, 2001, pp. 1-3.

[30] T. Sohn, J. Moon, S. Lee, D.H. Lee, and J. Lim, "Covert Channel Detection in the ICMP Payload Using Support Vector Machine," Computer and Information Sciences - ISCIS, vol. 2869, 2003, pp. 828-835.
[31] Daemon9, "Project Loki," Phrack, vol. 49, 1996, p. 6.
[32] Daemon9, "Loki2 (the implementation)," Phrack, vol. 51, 1997, p. 6.
[33] Zelenchuk, "Skeeve - ICMP Bounce Tunnel," 2004, http://www.gray-world.net/poc_skeeve.shtml, 2004.
[34] D. Stødle, "ptunnel - Ping Tunnel," 2005, http:// www.cs.uit.no/daniels/PingTunnel, 2005.
[35] S. Hamdy, Z. Trabelsi, and W. El-Hajj, "Implementation of an ICMP-based covert channel for file and message transfer," Proceedings of the 18th International Symposium on Computer and Information Sciences, 2008, pp. 894-897.
[36] J. Rutkowska, "The Implementation of Passive Covert Channels in the Linux Kernel," Proc. Chaos Communication Congress, Dec, 2004.
[37] E. Chan, R. Chang, and X. Luo, "CLACK: A Network Covert Channel Based on Partial Acknowledgment Encoding," IEEE International Conference on Communications, Dresdan: IEEE, 2009, p. 1–5.
[38] D. Kaminsky, "IP-over-DNS using Ozyman," 2004, http://www. doxpara.com/, 2004.
[39] T.M. Gil, "IP-over-DNS using NSTX," 2005, http://thomer. com/howtos/nstx/, 2005.
[40] P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION," RFC 1035, IETF, Nov, 1987.
[41] A. Merlo, G. Papaleo, S. Veneziano, and M. Aiello, "A Comparative Performance Evaluation of DNS Tunneling Tools," Computational Intelligence in Security for Information Systems, Á. Herrero and E. Corchado, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 84-91.
[42] P. Neyron, O. Richard, and L. Nussbaum, "On Robust Covert Channels Inside DNS," 24th IFIP International Security Conference, Pafos, Cyprus: 2009, pp. 51-62.
[43] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Tunnel Hunter: Detecting application-layer tunnels with statistical fingerprinting," Computer Networks, vol. 53, 2009, pp. 81-97.
[44] F. Petitcolas, M. Kuhn, and R. Anderson, "Information hiding-a survey," Proceedings of the IEEE, vol. 87, 1999, pp. 1062-1078.
[45] D. Hoeflin, K. Meier-Hellstern, and A. Karasaridis, "NIS04-2: Detection of DNS Anomalies using Flow Data Analysis," Global Telecommunications Conference, 2006, pp. 1-6.
[46] P. Vixie, Extension Mechanisms for DNS (EDNS0), 1999.
[47] A. Bhasah, Data Analysis Method, Utusan Publications & Distibutors Sdn Bhd, 2007..

Syaril Nizam Omar is currently a PhD student in the Department of Computer Systems and Communications of the Faculty of Computer Science and Information Systems at the Universiti Teknologi Malaysia. He obtained M.Sc. Information Security from Universiti Teknologi Malaysia (Malaysia) in 2008. He has been involved in lots of academic research since then; presently he is a member of Pervasive Computing Research Group at UTM, while his research interest Covert Channel. He has published in many national and international learned journals.

Ismail Ahmedy is currently persuing PhD in wireless sensor networks at department of computer system and communication, faculty of computer science and information system, universiti teknologi malaysia. He currently hold an academic post in university malaya since 2009. He obtained M.Sc. (Computer Science) from university of queensland, australia in 2009 and B.S.c (Computer Science) from universiti teknologi malaysia in 2006. His research interest is in wireless sensor networks (protocol and signal coverage).

Md Asri Ngadi received his BSc in Computer Science,and the MSc in Computer Systems from Universiti Teknologi Malaysia in 1997 and 1999 respectively, and the PhD degree from Aston University, UK in 2004. He is an associate professor in the Faculty of Computer Science and Information System, Universiti Teknologi Malaysia His research interests are computer systems and security,information assurance and network security.