DNS ID Covert Channel based on Lower Bound Steganography for Normal DNS ID Distribution

Abdulrahman H. Altalhi[1], Md Asri Ngadi[2], Syaril Nizam Omar[2] and Zailani Mohamed Sidek[3]

**[1] Department of Information Technology, College of Computing and Information Technology
King Abdulaziz University, Jeddah, Saudi Arabia**

**[2] Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia
Skudai, Johor, 81310, Malaysia**

**[3] Information Security Department, Advanced Informatics School (AIS)
Universiti Teknologi Malaysia, Kuala Lumpuer, 54100, Malaysia**

Abstract

The covert channel is a method used to send secret data within a communication channel in unauthorized ways. This is performed by exploiting the weaknesses in packet or network communications with the intention to hide the existence of a covert communication. The DNS identification (DNS ID) method has been exploited by Thyer. However, the major problem in Thyer's implementation is that the encrypted cipher was directly inserted as a DNS ID value, which is abnormal, compared to the normal DNS ID distribution. We have overcome this problem through the application of Steganography to insert the cipher value into the DNS ID. The data set test for normal DNS ID is taken from MAWI. We tested four different message lengths and plotted the distribution graph. We found that the proposed result is normal compared to normal distribution of the DNS ID. Therefore, this method produces a normal distribution for DNS ID covert channel. .

Keywords: *DNS Identification, Covert Channel, Normal Distribution.*

1. Introduction

The covert channel (CC) is a method designed to prevent custodian or network monitoring devices from detecting the information exchanged between two parties. This means, that there should be no way for a warden to observe what is being exchanged between the communicating parties. However, if there is no study performed on the effect of the method used to conceal the secret into the packets, these packets could raise suspicions that will alert the warden. Once the warden has been alerted, the warden may record whatever is being exchanged between the parties and later perform an analysis to grasp that there is secret information in the exchange. This does not mean that the warden has to reveal the information, but it may mean that the warden will be suspicious of the activity.

For example, two parties may be engaged in an exchange of a secret message and protect the information with a cipher. The cipher itself reveals that something important or secret is being transmitted in the exchange – and that could further invoke activity to decrypt the message. In [1], the authors explain the conditions under which the information is more secure via the use of Steganography. Steganography is a method used to conceal the existence of the message in a tangible medium cover, such as a picture, a movie or some music. However, this does not mean the CC is not secure against Steganography. The main difference between CC and Steganography is the medium cover. Steganography hides the message in the file, while CC hides the data in the packets, which are a volatile medium. A packet will be destroyed after it reaches the destination or when it cannot reach its destination. These volatile characteristics have made CC the preferable way to send a secret message.

CC has been used to send malicious messages [2, 3], steal information [4], control a Trojan [5], and leak sensitive information [6, 7]. Albeit, the CC also has good applications, such as protecting anonymity and tracing [7], protecting anonymity and preserving privacy [8] and protecting government information and e-commerce transactions [9].

As of the current research, the physical layer has been exploited up to the application layer for CC [10]. As mentioned in [10], many firewalls have blocked internet traffic to reduce the CC threat. However, as stated in [10, 11], the DNS is less filtered because of the great need for

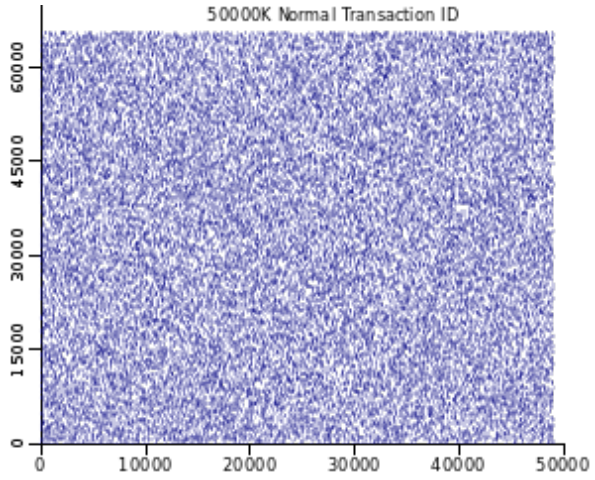

Fig. 1 The normal randomness of the DNS ID distribution taken from MAWI data set.

Internet access. DNS is used to translate the mnemonic name of the server to its corresponding IP Address. DNS is built on top of the UDP protocol, which means that it is connectionless and has low reliability. As reported in [12], based on a study performed on DNS queries for two weeks, the minimum number of queries per second is 11 and the maximum number is 90 queries per second. The high number of queries per second is because duplicated queries are allowed for the purpose of increasing the reliability of the response as required in the relevant RFC. When duplicated DNS queries are received, the DNS server will respond to the queries based on the autonomy field in DNS, which is the DNS ID. The DNS ID can be viewed as an authentication key for each DNS request. As stated in the RFC, the DNS ID should be random enough to make sure that each query has a unique DNS ID.

The unreliability of the DNS protocol and the randomness of the DNS ID give the researcher an opportunity to study the applicability to exploit it for CC. A previous technical report [13] mentions the ability to embed 16 bits of concealed values in the DNS ID. Later, in [11], Thyer elaborates and shows how the DNS ID can be used to send a secret message. Thyer developed a method with a plain insertion and cipher insertion, which prevents the warden from noticing or being able to recognize the hidden information in the DNS ID. However, merely inserting a block cipher string into the DNS ID violates the randomness distribution of the DNS ID. Figure 1 below shows the normal randomness distribution of 50,000 K DNS ID taken from MAWI data set.

For that reason (based on Figure 1), the challenge is not just to craft an encrypted message and embed it into the protocol field but to develop a method that does not violate the property characteristic of the protocol field exploited as shown in [14]. Murdoch shows that the embedded method must adhere to the characteristics of the original design.

Therefore, we would like to present a method based on the characteristic of LSB to embed 8 bits of a secret message into the lower bound or 8 bits of less significant field of DNS ID. We protect the message by encrypting it with a block cipher.

The rest of the paper is organized as follows. Section 2 will discuss the related studies and discussions in the literature review. This is followed by the presentation of the overview of the design in section 3. Section 4 will show how the DNS ID is used to implement the covert channel. And section 5 will discuss the findings and tests. Lastly, the study conclusions are presented and the related improvements are explained in section 6.

2. Related Works and Discussions

The study of the covert channel was originated in 1973 by Lampson. It was then known as the subliminal channel [15]. The first use of a covert channel for a secret purpose was applied when the United States carried out a mission to calculate how many Minuteman missiles they had in a 1000 silos - without revealing which silos actually contained missiles [16]. From 1978 until today, more than a dozen research studies have been performed on covert channels. In this study, we would like to review the work related to covert channels performed on IP, ICMP, TCP, UDP and the Application layer.

2.1 IP Protocol

In [17], Taeshik explained how the IP Identification (IP ID) field can be manipulated to embed ASCII alphabets. This method was used by Rowland to multiply the ASCII as a hex value with 255 because 255x255 is 65535, which is the value of 16 bit fields. The proposed method was excellent in concealing data in the IP ID because the data resembled the value of an IP ID. Note that the initial intention of Rowland was to prove that the IP ID can be exploited to carry a secret message. The design was excellent for sending unique characters. However, the design seemed suspicious if closely monitored, as in the case of the use of duplicate characters. Then, Ahsan in [18] improved the method using a Toral Automorphism System that used a pseudo random sequence to ensure that the modified IP identification is random.

Yogi Metta in [19] theoretically explains how the value of DF could be used to send a message. The method can successfully be implemented if we know the MTU of each router. Cauich and colleagues in [20] used the IP Offset field to embed the data. The only problem with the IP offset field occurs when the DF is set and there is data in IP Offset field. This would trigger an IDS or IPS. In [21], Zander and colleagues demonstrated how the TTL is manipulated to send a value - 1 or 0. The TTL method is

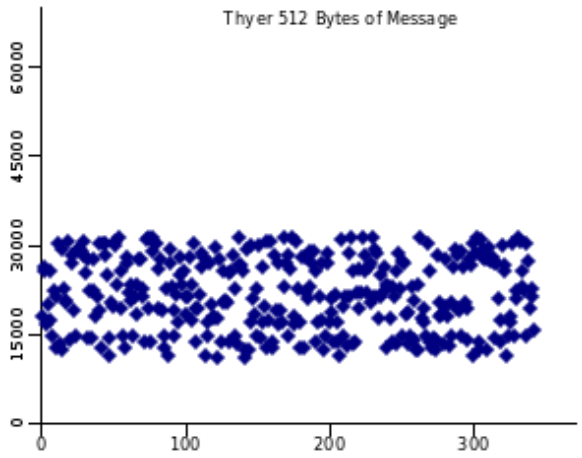

Fig. 2 The randomness distribution of the DNS ID taken from Tyher CC when embedded 512 bytes of a message

very suitable for sending a small amount of data however the variations of Operating Systems in the network need to be carefully studied because Fyodor in [22] mentioned that each OS uses different TTLs to uniquely identify the OS. Abad in [23] theoretically described how the Checksum value could carry the data, although the kernel will discard the packet if the checksum value is wrong. Moreover, the checksum value will change when the packet enters the router. On the other hand, the checksum CC is desirable in a LAN because the detection of checksum CC in a LAN will be very difficult.

2.2 ICMP Protocol

RFC 792 (which describes the Internet Control Message Protocol, abbreviated as ICMP) was designed to help to notify the system in the event that an error occurred in the network path. Its most common use is ICMP type 0 (echo reply) and 8 (echo request). Daemon 9 demonstrated the ICMP covert channel by exploiting the payload of ICMP type 0 and 8. The ICMP payload by default could carry 56 bytes of data. Therefore, the number of ICMP covert channel has increased, such as in Loki [24][25], ICMP bounce tunnel [26], Ping tunnel [27] and 007Shell [28]. Later, Zouher in [29] used an ICMP covert channel to send a file and message by exploiting the record router IP header. The transmission of many ICMP covert channels means the security professional should optimize their security parameters to limit the ICMP packets.

ICMP will be a great CC within the LAN because most of the firewall will not allow inbound ICMP packets to enter their network. Unless it is used for outbound traffic, an ICMP covert channel will be applicable.

2.3 TCP Protocol

Rowland in [30] has shown the basics of a TCP covert channel by exploiting the TCP sequence number (SEQ) fields (32 bits). He used the same method as he did in IP Identification: just multiply the 255x255x255x ASCII value. This multiplication result is fitted within the TCP SEQ field. Again, Rowland's purpose is just to show that the TCP SEQ field is can be exploited for CC. Rutkowska, then, in [31], shows an advanced method by embedding the cipher into the TCP SEQ so that the TCP SEQ field will resemble the normal characteristic of the TCP SEQ field. Later, Murdoch, in [32], shows a better method that resembles the original design of the TCP SEQ field. The Murdoch method fixed the problem in Rutkowska's method by interpreting each TCP SEQ field as an independent field. Therefore, there is no issue when there is no data to be sent.

2.4 UDP Protocol

UDP was designed to exchange messages with minimum protocol overload processing. The only possible covert channel field on UDP is the source port, and it is only applicable on LAN. Conversely, UDP has been used to carry another internet protocol, such as IP [33][34] and TCP. rare.

2.5 DNS

We found that most of the DNS exploits work by bypassing the firewall with the use of a tunnel. DNSTX and DNScat are the tunnels that make use of the DNS query field to carry their data. The DNS query field (as noted in [35]) is used to carry a domain name. The format for the domain name is obvious, so this method is also applicable for sending a secret message over the net. It is not sneaky, as the unusual data in a DNS query is easily detected using a Network Monitor.

Thyer in [11] shows a reasonable level of stealth against the DNS tunnel model. However, the DNS tunnel can be used to send high bandwidth data. Tyher exploits the DNS ID field as the medium carrier to hide the 16 bits of secret data. In essence, the 64 bit cipher is reasonably random. However, the analysis performed with 512 bytes of the message showed that the sub-group of 16 bits from 64-bit ciphers was not randomly distributed. Figure 2 shows the result of the Tyher method in sending 512 bytes of a message. The message was encrypted using the Blowfish encryption algorithm. The figure shows that the secret message was randomly distributed within the range from 11,000 K to 32,000 K. This indeed was different from the normal DNS ID distribution as shown in Figure 1. However, this does not mean that the Thyer CC is easily detected or blocked because the act of embedding the

secret in the DNS ID with a cipher string still produces a good degree of randomness.

Within the DNS header field, we found that the DNS Identification is unique because the field is generated using a pseudo-random method, and the value will not change along the network path until it reaches its destination [36]. Moreover, DNS is carried by UDP. UDP is connectionless, which means there is no tracking mechanism, such as the sequence number or acknowledgement number in TCP.

3. DNS ID CC Design

In this section, we give an explanation of how we design the DNS ID CC that uses the DNS ID as its medium carrier to carry a secret message.

3.1 An Overview of DNS ID CC

To simplify the explanation of the proposed DNS ID CC, we divide the processes into levels or ladders. On the encoder site, the level will start from zero, which indicates the first step or process and will increase subsequently until the embedded process is completed. On the decoder side, the level will start in descending mode until the message is readable to the receiver. In this CC design, we assumed Alice and Bob are communicating openly through the overt network to send their secret messages. The message is protected with the block cipher encryption algorithm. They share a secret key (Sk), which is used to encrypt and decrypt the message. Therefore, with this strategy, the CC design can be explained as follows:

- Level 0: On Alice's side, the message m is the secret message Alice wants to deliver.
- Level 1: Alice encrypts m with the Blowfish algorithm and stores the cipher C in the list.
- Level 2: The CC will divide the C into a sub-group of 8 bits. The sub-group is processed in sequence. The CC will activate the pseudo-random generator and initialize the random seeds. Then, it will generate the random number in the range of 0 to 65535. This random number is the DNS ID that will be used to embed the bytes of the sub-groups. The CC will embed the byte in the lower bound of the DNS ID.
- Level 3: CC will build the DNS packets with the embedded DNS ID. This packet is inserted in the list.
- Level 4: This is the stage where the packet is inserted into the network where the destination is Bob's IP address.
- Level 4: On Bob's side, Bob listens to the DNS port and takes the DNS packets that arrived on a fixed time lapse.
- Level 3: Bob will extract the lower bound DNS ID and store it in the C string. The C string is ready to be decrypted after the complete DNS packet has been received.
- Level 2; The C string will be decrypted with Sk.
- Level 1: The m is ready for Bob. categories

4. Experiment

The experiment that will be used test the proposed CC will result in DNS ID values with the DNS ID from the MAWI data set and the DNS ID values from Tyher model. Four different sizes of a message will be used as the comparisons.

4.1 Dataset Analysis

The DNS data set was based on the MAWI data set captured with tcpdump on a Mac 2008. The size of the tcpdump file after decompressing it is about one gigabyte. We then filter the DNS standard query from the dump file and we obtain approximately 49,056 K of DNS queries. We then further extract the DNS ID of each query and then run the descriptive statistical analysis on the DNS ID. We found that the mean value is 32,778, and the standard deviation (SD) is 18,883. This means that the value is largely dispersed in the range of +/-18883 from the mean. This DNS ID value will be the benchmark to determine whether the proposed method of DNS ID CC is dispersed with the same distribution. Figure 1 shows the distribution of the MAWI DNS ID data set distribution. Notice that the DNS ID is scattered within values from 0 to 65535.

4.2 Experiment Results

Our results will be discussed by showing the graphs of the DNS ID distribution and the Mann-Whitney U test on four different message sizes.

4.2.1 Randomness Distribution Comparison

Our results will be discussed by showing the graphs of the DNS ID distribution and tIn this experiment, we test four message sizes that are in the range of 64 bytes, 128 bytes, 256 bytes and 512 bytes. As mentioned in Section 3.1, the message will be encrypted using the Blowfish encryption algorithm. After encryption, the sizes of the cipher were in the range of 88 bytes, 345 bytes, 689 bytes and 1369 bytes. Figure 3 shows the randomness distribution of the DNS ID based on the result from the proposed CC and Tyher CC. The labels a, c, e, and g are the randomness distribution graphs showing the results of the Tyher CC DNS ID distribution.

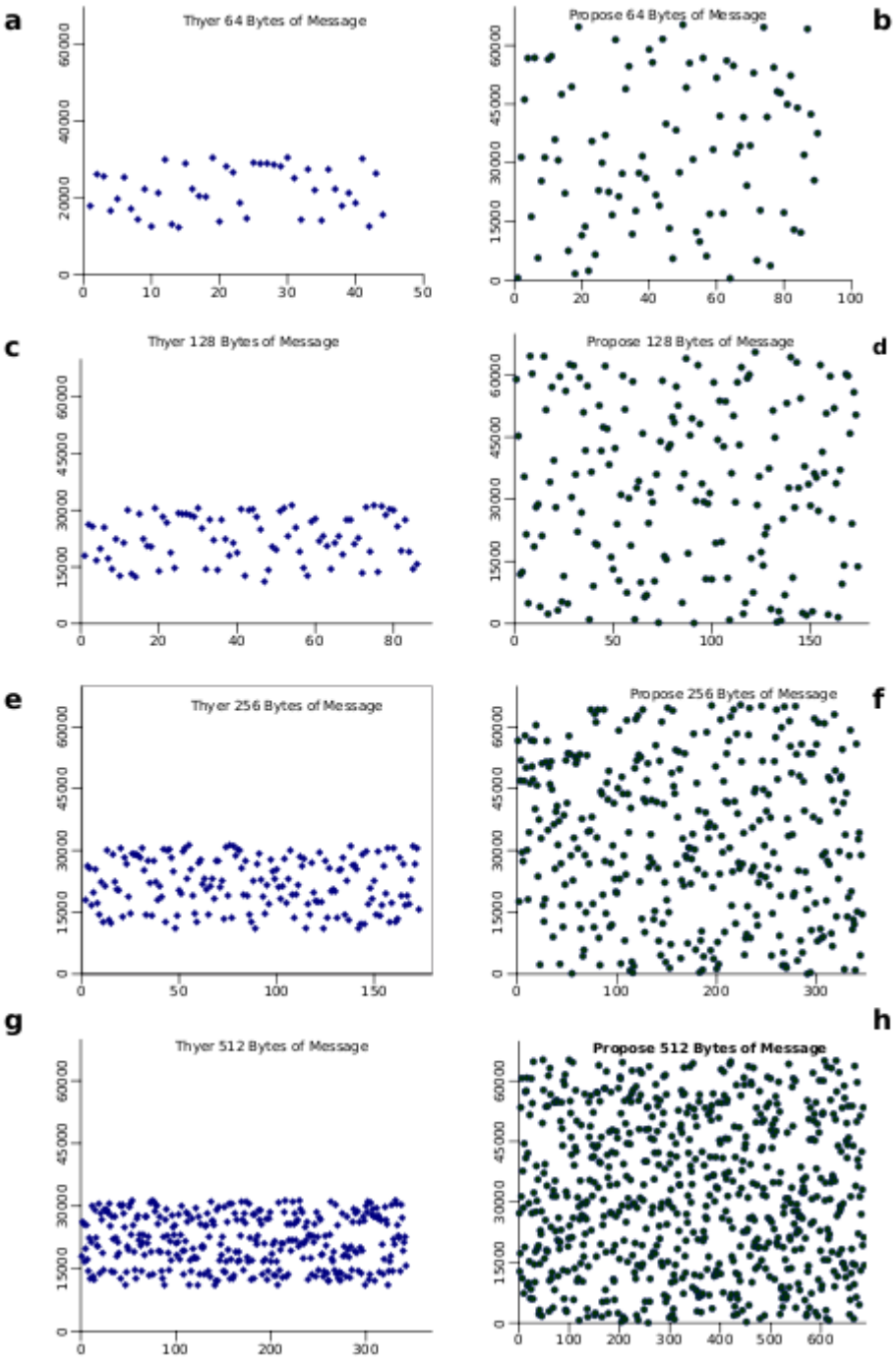

Fig. 3 The result of the DNS ID values based on 4 different messages sizes ranging from 64, 128,256 and 512 bytes. In the left side of the figure, labels a, c, e, and g are the result from the Tyher CC, while labels b, d, f, and h are the result of the DNS ID values from the proposed CC

The labels b, d, f and h show the randomness distributions of our CC DNS ID results. The label a shows the 64 byte message, c the 128 byte message, e the 256 byte message and g the 512 byte message. The label b is for 64 bytes, d for 128 bytes, f for 256 bytes and h for 512 bytes.

Note that the number of the DNS ID is different between the proposed CC and the Tyher CC. In [11], with the Tyher CC design, the embedded method used the entire 16 bit DNS ID field to conceal the message, while we used only the first 8 bits of the DNS ID to conceal the message. For that reason, as shown in the result, for each message size,

the Tyher method only requires half of the packet size compared to what is required by our method.

Based on the results in Figure 3, we can see that for each message size, the randomness distribution of our result is far more dispersed than the Tyher CC. Moreover, our CC result was dispersed in a range that resembled the dispersal range from the MAWI data set. To confirm this, we further investigated the SD values. The SD value for our 64 byte message is 18,727, with a mean value of 32,891. The difference between our mean value and the data set is 133, and the difference with our SD is 156. This proves that, for 64 bytes of a message, though the number of the packets required is doubled, the value of the difference between the proposed method and the data set is trivial compared to the Tyher CC.

We further evaluated the SD and the mean value for the 128, 256 and 512 byte messages of our DNS ID values. For 128 bytes, the SD value is 19,862 and the mean value is 32,432. The SD value for 256 bytes is 18,816 with mean value 32,502. For 512 bytes, the SD value is 18,599 with a mean value of 31,574. We can see that the difference between our mean value and the data set is within the range of +/-1200. Moreover, the difference with the SD is within the range of +/- 1000. Therefore, this proves that the proposed CC DNS ID value is not significantly different from the data set, which is supported by the graphs that show the distribution of the randomness (Figure 3 for the labels b, d, f, and h).

4.2.2 Mann-Whitney U Test

The Mann-Whitney U test is a non-parametric test to test whether the independent sample had an equally large value. In our case, the SD was dispersed widely from the mean value. The results shown in Table 3 further support the analysis we made in Section 4.2.1 and confirmed the results in Figure 3(f) that there is a randomly distribution of the data set test sample. Note that the z-value and the p-value are slightly decreased if compared with the results in Table 2. However, this does not show a significant difference compared with the sample test, as the p-value and the z-value are still high. Albeit, there is a high increase in p-value and z-value for Tyher CC, which show a wider range than the DNS ID of the test data set.

Table 1: The MWU test for 64 bytes of message

64 Bytes	Z-Value	P-Value
Proposed CC	0.48	0.64
Thyer CC	2.59	0.01

Table 2: The MWU test for 128 bytes of message

128 Bytes	Z-Value	P-Value
Proposed CC	0.07	0.94
Thyer CC	2.96	0.003

Table 3: The MWU test for 256 bytes of message

256 Bytes	Z-Value	P-Value
Proposed CC	0.43	0.67
Thyer CC	5.34	0.0001

Table 4: the MWU test for 512 bytes of message

512 Bytes	Z-Value	P-Value
Proposed CC	1.72	0.09
Thyer CC	8.16	0.0001

The result in Table 4 shows a decrease in the p-value and z-value of our DNS ID for 512 bytes of message. However, this is far better than the Thyer CC DNS ID results, which show an increase of more than 30% from the results in Table 3. Overall, the results from the MWU U-tests have shown that the proposed CC was spread within the random distributions of the test data set. Therefore, we can conclude that the MWU U-test results were consistent with the conclusion in sub-section 4.2.1. The MWU test also further supports the DNS ID values we plotted in Figure 3(b, d, f, h), which shows that the DNS ID values are widely spread in the range of 0 to 65535.

5. Conclusions and Future Works

In this paper, we have undertaken studies and implemented a capable DNS ID CC that uses the lower bound of the DNS Transaction ID field to conceal secret values to be transmitted across a network. The solutions we have developed have three main advantages compared to previous studies. First, it can conceal a message inside the DNS ID without violating the random characteristic of the DNS ID. Second, the CC method did not leverage large significant differences from the sample data set, which means it is very difficult for any IDS or IPS to detect that the DNS ID is an object cover that carries a concealed message. Third, the lower bound embedding has successfully proved that it will not affect the normal randomly distribution of the DNS ID.

For the near future, our studies will focus on the need to design a method that can receive and validate CC packets against non-CC packets.

Acknowledgments

The author would like to thank the King Abdulaziz University for supporting the research and National Science Fellowship of Malaysia for the PHD scholarship.

References

[1] Petitcolas, F., Anderson, R., Kuhn, M.: Information hiding-a survey. Proceedings of the IEEE. 87, 1062-1078 (1999).
[2] Trabelsi, Z., Jawhar, I.: Covert File Transfer Protocol Based on the IP Record Route Option. Information Assurance and Security. 5, 64-73 (2010).
[3] Maalej, L., Hammouda, S., Trabelsi, Z.: Towards Optimized TCP/IP Covert Channels Detection, IDS and Firewall Integration. ACSAC. p. 1–5 (2008).
[4] Lewandowski, G., Lucena, N.B., Chapin, S.J.: Analyzing network-aware active wardens in IPv6. ,, Systems Assurance Institute, Syracuse University, Syracuse, NY 13244, United States (2007).
[5] Wang, C., Ju, S.: The new criteria for covert channels auditing. Presented at the (2004).
[6] Moskowitz, I., Newman, R.: Practical Covert Channel Implementation through a Timed Mix-Firewall. Presented at the (2008).
[7] Knight, G.S., Smith, R.: Predictable Design of Network-Based Covert Communication Systems. Presented at the (2008).
[8] Lin, C., Kuo, S., Yarochkin, F., Dai, S., Huang, Y.: Introducing P2P architecture in adaptive covert communication system. First Asian Himalayas International Conference on Internet. pp. 1-7. ,, Kathmandu (2009).
[9] Trabelsi Z., E.H.: Traceroute based IP channel for sending hidden short messages. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 4266 LNCS, 421-436 (2006).
[10] Zander, S., Armitage, G., Branch, P.: A Survey of Covert Channels and Countermeasures in Computer Network Protocols. Communications Surveys & Tutorials, IEEE. 9, 44-57 (2007).
[11] Thyer, J.: Covert Data Storage Channel Using IP Packet Headers, (2008).
[12] Bojan, Z.: Security Monitoring of DNS traffic.
[13] M. Smeets, M.K.: Research Report: Covert Channels, http://www.os3.nl/~mrkoot/courses/RP1/researchreport_2006-02-15_final2.pdf.
[14] Murdoch, S.J., Lewis, S.: Embedding covert channels into TCP/IP. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 3727 LNCS, 247-261 (2006).
[15] Lampson, B.W.: A note on the confinement problem. in Pmc. of the Communications of the ACM, October. 16, 10pp613-615 (1973).
[16] Simmons, G.J.: The History of Subliminal Channels. IEEE Journal on Selected Areas In Communications. 26, 452-462 (1998).
[17] Sohn, T., Seo, J., Moon, J.: 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. 313-324 (2003).
[18] Ahsan, K., Kundur, D.: Practical data hiding in TCP/IP. Proceedings of the Workshop on Multimedia Security at ACM Multimedia. 63-70 (2002).
[19] Mehta, Y.: Communication over the Internet using covert channels, https://www.cs.drexel.edu/~vp/CS743/Papers/ypm23-hw2.pdf, (2005).
[20] Cauich, E., Watanabe, R., Science, C., Zaragoza, A.: Data Hiding in Identification and Offset IP fields. In Proceedings of 5th International School and Symposium of Advanced Distributed Systems (ISSADS). 118-125 (2005).
[21] Zander, S., Armitage, G., Branch, P.: Covert Channels in the IP Time To Live Field. In Proceedings of Australian Telecommunication Networks and Applications Conference (ATNAC). (2006).
[22] Fyodor: Remote OS Detection via TCP/IP Fingerprinting. Phrack Magazine. 8, (1998).
[23] Abad, C.: IP Checksum Covert Channels and Selected Hash Collision. Technical report. 1-3 (2001).
[24] Daemon9: Loki2 (the implementation), (1997).
[25] Daemon9: Project Loki, (1996).
[26] Zelenchuk, I.: Skeeve - ICMP Bounce Tunnel, (2004).
[27] Stødle, D.: ptunnel - Ping Tunnel, (2005).
[28] Sohn, T., Moon, J., Lee, S., Lee, D.H., Lim, J.: Covert Channel Detection in the ICMP Payload Using Support Vector Machine. Computer and Information Sciences - ISCIS. 2869, 828-835 (2003).
[29] Hamdy, S., Trabelsi, Z., El-Hajj, W.: Implementation of an ICMP-based covert channel for file and message transfer. Presented at the (2008).
[30] Rowland, C.: Covert channels in the TCP/IP protocol suite. Tech. rep., First Monday. ACM Transactions on Information and Systems Security. 12, Article 22 (1997).
[31] Rutkowska, J.: The implementation of passive covert channels in the Linux kernel. Proc. Chaos Communication Congress, Dec (2004).
[32] Murdoch, S.J., Lewis, S.: Embedding covert channels into TCP/IP. The 7th Information Hiding Workshop. 247-261 (2005).
[33] Kaminsky, D.: IP-over-DNS using Ozyman, (2004).
[34] Gil, T.M.: IP-over-DNS using NSTX, (2005).
[35] Mockapetris, P.: DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. RFC 1035, IETF, Nov. (1987).
[36] Bellovin, S.M.: A technique for counting natted hosts. Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment. pp. 267-272. ACM, Marseille, France (2002).

Abdulrahman H. Altalhi is an assistant professor of Information Technology at King Abdulaziz University (KAU). He received a BSc in Computer Science from KAU on December of 1993, MSc on Computer Science from the University of New Orleans on August of 1998. He has obtained his Ph.D. in Engineering and Applied Sciences (Computer Science) from the University of New Orleans on May of 2004. He served as the chairman of the IT department at KAU for two years (2007-2008). Currently, he is the

Vice Dean of the College of Computing and Information Technology of KAU. His research interest include: Networking, Wireless Networks, Computer Security, Software Engineering, and Computing Education.

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.

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 is Information Hiding.

Zailani Mohamed Sidek Received Diploma in Agriculture, University of Agriculture, Malaysia in 1977, BSc in Business Administration from California State University, Fresno, USA in 1982, MSc in MIS from Texas Tech University, USA in 1984, and PhD in Computer Science from Universiti Teknologi Malaysia, Malaysia in 2005. He worked as a Bank Credit Officer in the Agriculture Bank of Malaysia in 1977-1980; Lecturer in the Universiti Teknologi Malaysia, Malaysia in 1982-present; Head of Department in the Faculty of Computer Science & Information Systems, UTM, Malaysia in 1989-1995. He is currently lecturing in the Advanced Informatics School, Universiti Teknologi Malaysia International Campus, Kuala Lumpur, Malaysia.