Dropped Packet Problems in Stepping Stone Detection Method

Mohd Nizam Omar, Lelyzar Siregar, and Rahmat Budiarto

NAv6 Centre, Universiti Sains Malaysia, Penang, 11800 Penang, Malaysia

Summary
This paper discusses one of the issues that are not covered by current stepping stone detection based researches. Although dropped packet problems are well-known problem in real network environment, all of the stepping stone detection researches just assume that dropped packet problems do not occur. Stepping stone detection research already in complex condition where each enhancement of the research attempts to solve problem such as encryption, delay and chaff without focused to solve dropped packet issues. For that situation, this research believes that to make sure that dropped packet problem can be solved; it should begin from the beginning of the stepping stone research. In other words, to solve dropped packet problem, we must solve it at the early stage of stepping stone research without combine it with current complex condition of stepping stone research. For that purpose, experiment has been executed by using NS-2 to prove the existence of the dropped packet in stepping stone detection environment. The experiment has been setup for a basic stepping stone detection environment. From the result obtained, it shows that dropped packet definitely influences the detection of stepping stone.

Key words:
Intrusion tracing, stepping stone detection, dropped packet

1. Introduction

Internet has become more important than before but, at the same time, Internet attack has increased significantly [1]. Attacker can use intermediate host as their stepping stone before attacking the real target [2]. This compromised host gives the advantages for attacker to hiding their track. From that problem, Staniford-Chen and Herberlein [3] as the pioneer in stepping stone research introduces "thumbprint" technique to detect stepping stone intrusion. In this technique, "thumbprint" represents as a summary of the content of a connection. The comparison of two different "thumbprints" determines either the two different connection is stepping stone or not. Because of research by [3] depends on content of packet; their approach will not work in encrypted connection. Zhang and Paxson [2] overcome this problem by proposing a time-based method on interactive sessions. In their research, distinctive characteristics, such as packet size and timestamps to identify a connection is used. The advantage of the technique is this technique does not require tightly synchronized clock and robust against retransmission however the disadvantages is not available in real-time, and vulnerable to intruder manipulation such as random delay and chaff perturbation. Research by Yoda and Etoh [4] proposed approach similar to [3] research. The different is they used "deviation" as to deviate two different connections. From their research, they come from the observation that the deviation for two unrelated connections is large enough to be distinguished from the deviation of connections in the same connection chain. They approach also prone to the same problem as [3] methods . Research by Wang et al. [5] proposed an approach named "Inter-packet delay (IPD)". They used IPD characteristics to detect stepping stones. IPD may be preserved across many router hops and stepping stones. As previous researches are prone to delay perturbation, Wang and Reeve [6] introduce active watermark scheme that can overcome that problems. The watermark is used to adjust the inter-packet delays of selected packets in the flow. Strayer et al. [7] introduce "State Space" method. In this method, they make two assumptions. First, the likely hood of one transmission being a response to a prior transmission generally decreases as the elapsed time between these transmission increases. Secondly, the inter arrival times between a fixed even and any other event is approximately Poisson distributed.

Dohono et al. [8] introduces method that known as "Multiscale". In their research, they use wavelets and similar multiscale methods to separate the short-term behavior of the streams. However, this method requires the intrusion connections to remain for long periods and experiment has not been executed as to test their finding. Blum et al. [9] propose method known as "Detect-Attack". In their research, they analyze algorithms for stepping stone detection by using ideas from Computational Learning Theory (COLT) and the analysis of random walks. The research successfully obtain provable upper bounds on the number of packets needed to confidently detect and identify stepping stone streams with proven guarantees on the false positives. At the same time, they also examine the consequences when the attacker inserts chaff into the stepping stone traffic and provide lower

Manuscript received Feb 4, 2008.
Manuscript revised Feb 19, 2008.

bounds on the amount of chaff that an attacker would have to send to evade detection. None of these previous research can effectively detect stepping stone when delay and chaff perturbations exist either or/and delay and chaff exist simultaneously in the connection. For that reason, Zhang et al. [10] proposed a research that can overcome the problem when there are delays and chaff exists simultaneously.

From the first research in detecting stepping stone by [2] to the latest research at this moment [10], it is proved that research in stepping stone area has been in complex condition. From detecting stepping stone is done by just looking at their content [2], detecting stepping stone becomes complex by the existence of perturbation such as delay and chaff [10]. However, it is also proved that previous research just take for granted of the existence of dropped packet problem. This can be looked at most of the research although they realize that dropped packet might occur, but they just assume that dropped packet does not occur. For that reason, this research will proof that dropped packet will affect the overall stepping stone processes through the experiment.

The important of the dropped packet problem needs to be solved stated by Wang [11] in his PhD's dissertation. Wang [11] listed that dropped packet is the one of the challenges in tracing through stepping stones. Moreover, research by Venkateshaian and Wright [12] used dropped packet as their one of the technique to stop the existence stepping stone detection from detecting stepping stones. These two examples of research give support that dropped packet is the problem that needs to be solved. From the research done on previous stepping stone detection research, there are two main reasons that cause previous research take for granted to drop packet problem; 1) stepping stone detection research only focused on detection on Local Area Network (LAN) environment and 2) stepping stone detection is not in real-time. For the first reason, most of stepping stone research such as [2], [3] and [4] focus on solving stepping stone detection in LAN environment. This can be proved by looking their experiments that run on LAN environment. LAN environment is less prone to dropped packet problem compared to real-time Internet environment. For the second reason, solving stepping stone in non real-time environment such as [2], [3], [4] causing the dropped packet problems can be taken for granted. This is because the calculation can be fixed although dropped packet occurs.

The rest of this paper is structured as follows. Section 2 discusses research terms that used in this research. Section 3 defines dropped packet in network. In Section 4, we discuss further on dropped packet in stepping stone detection environment. Section 5 gives the experiment setup that has been done in this research. In Section 6 and 7, we discuss and analyze the experiment that has been done, respectively. Finally, we summarize the whole work and give future work in Section 8.

2. Research Terms

Before we start more detail on focused discussion, there are several research terms or terminology used that need to know.

A person or program can login from Host 1 to Host n through Host i - 1,… i, i + 1,…, and Host n as shown in Figure 1.

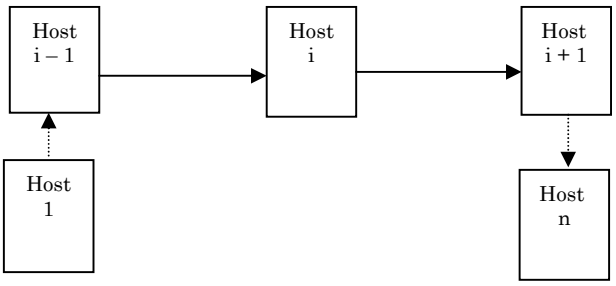

Fig. 1 Current Detecting Stepping Stone Chain Example

Connection occurs when a host logins from one host to another host. Connection is when a given n host H_1, H_{i-1}, …, H_i, H_{i+1},…, H_n is a sequence of connection as a chain $C = <C_1, C_{i-1},\ldots, C_{i+1},C_n>$ where C_i is a connection from Host i to Host H_{i+1}, for i = 1, …, n – 1. Downstream is a direction of user's login direction (according to arrow) and otherwise it is upstream.

Our previous research had divided stepping stone detection algorithm into three different parts. There are capturing [13], identifying [14] and comparing [15]. Although dropped packet can occur in identifying and comparing parts, this paper only emphasizes on dropped packet occurred in capture part because this part is more prone to dropped packet problem compare to identifying and comparing part. Moreover, current research also realizes that dropped packet occurs on connection of the stepping stone. As a result, our dropped packet is only focusing on Host i + 1 as shown in Figure 2. In our finding, dropped packet between Host i – 1 and Host i maybe occurred but if that occurred, it cannot influence the stepping stone detection. It is because in this case, dropped packet just be forwarded to the next host. However, when dropped packet occurred between Host i and i + 1, differentiation processes to obtain stepping stone will be affected.

Current researches on Stepping Stone Detection just focus on detection stepping stone by looking at downstream and upstream connection [20] [21] [22] compare to previous one that doing comparison on each

stepping stone connection [2] [3] [4]. For that reason, our proposed research does what current research is doing. Beside, the current research is more effective, it also enhances the possibilities to detect stepping stone more precise.

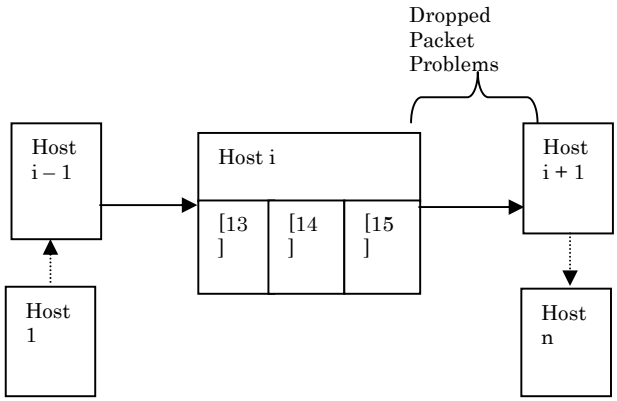

Fig. 2 Proposed Detection Stepping Stone Research Fields

3. Dropped Packet in Network

Previous researches on detecting stepping stone focus on interactive application such as Telnet, SSH, rlogin and so forth. However, the explanation of Telnet is needed here because of the usage of the Telnet is the basic of stepping stone detection. Furthermore, the goal of this research is to prove the existence of dropped packet in a basic of stepping stone environment. Telnet is one of TCP/IP application beside of e-mail, FTP and World Wide Web [23]. Telnet used interactive communication data between client and server. Telnet runs over connection-oriented Transmission Control Protocol (TCP). When a client wants to access a particular server, it initiates a TCP connection to the appropriate server, which responds to set up a TCP connection using the standard TCP three-way handshake. The connection of client server becomes more complex when it involves a stepping stone connection. Ordinary telnet connection just involves connection between client and server but in stepping stone telnet connection, when a packet is sent from first host to destination host, the intermediate hosts will forward the packet to destination host. When the packet reaches at destination host, it sends an echo back. Dropped packet that occurs between intermediate hosts and destination host will influence the overall stepping stone detection finding. This problem will be proved and shown in this research.

Two examples of dropped packet-based research other than stepping stone detection research are research by Nanyin and Micheal [29] and Ibrahim and Lang [30]. Research conducted by Nanying and Micheal studies about the implication of dropping packet from the front of a queue. Ibrahim and Lang [30] investigate the interaction among short and long TCP flows and how TCP service can be improved by employing a low-cost service differentiation scheme. These two of example show that dropped packet become an issue that need to be solve.

Dropped packet is one of issues in TCP/IP. It is discussed extensively in [16], [17], [18] and [19]. Dropped packet during the transmission are retransmitted automatically either because of the sender have not received the packet and acknowledgement or on request of the receiving server. Receiver will receive the retransmission of the same packet until an acknowledgement is received or the connection timeout expires [23]. In this case, stepping stone detection algorithm will face with redundancy of the network packet that needs to be captured.

4. Dropped Packet in Stepping Stone Detection

To detect stepping stone, there are three basic steps involved. As discussed in Section 2, they are packet capture, identification and comparison. For detecting the stepping stone, each step must be followed. Dropped packet makes the packet capture steps fail and this will influence the next steps. For that reason, it is important to solve the dropped packet problem. This research only focuses on proving the effect of dropped packet on packet capture steps by conducting an experiment on NS-2 simulation.

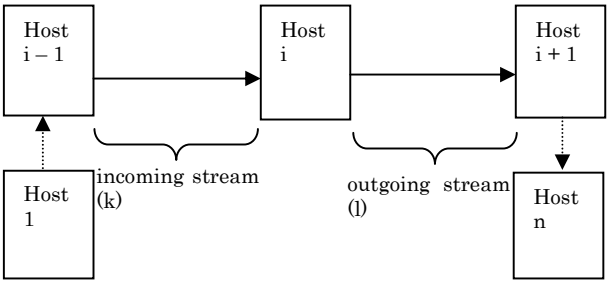

Fig. 3 Dropped Packet Problems in Stepping Stone Detection Environment

Figure 3 shows the overall dropped packet problem in stepping stone detection. Let i is stepping stoned host (host that being used to capture both incoming and outgoing streams) where k is incoming stream from previous host of i, i – 1 and l is outgoing stream to next host of i, i + 1. Dropped packet, dp problem is the problem when there is dropped packet occurs in connection timeout, ct period. Here, dropped packet can occur in ct times. As described before, there is a requirement to differentiate both k and l. Let stream k and l have a list of packet as described below

Stream k: k1, k2, k3,…, k(n+1) where (n > 1)
Stream l: l1, l2, l3, …, l(m+1) where (m > 1)

Because dropped packet problem can occur on stream l, there is a possibility where the number of packet on k is more compare to l in ct time. To differentiate stream at both k and l, stepping stone detection research introduces Differentiate Windows (DW) as the buffer to collect the packet for both incoming and outgoing stream [5]. The size of DW is arbitrary and there is no fix size stated. For example, research by Wang [5] used DW as 5. Differentiate Windows (DWs) with size s is used to determine the number of packet need to capture in k and l. Packets need to be full loaded according to s size before the differentiate processes can be done. Let both k and l has n and m = 5, DWs = 3, so both stream k and l list as below

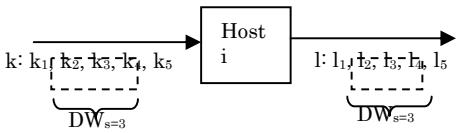

Fig. 4 General Incoming and Outgoing Stream with Packet Data

For the same value on above example, Let dp for l is l3 and The number of packet on l is less because of dropped packet shown in Figure 5.

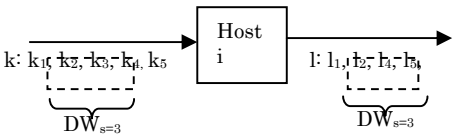

Fig. 5 Incoming and Outgoing Steam with dp is l3

In this case, the number of packet on l is 4 instead of 5. Furthermore, differentiate windows also capture the wrong packet, l2, l4, l5, instead of l2, l3, l4. This will cause the stepping stone detection is not successful in detecting stepping stone. There are many other possibilities of dropped packet problem that can influence the stepping stone detection afford. For example, there is more packet on l retransmitted because of dropped packet occur on l, retransmission on stream k and so forth. But what we show here is the basic on how dropped packet can influence the stepping stone detection.

5. Experiment

The purpose of the experiment is to show, identify and then prove the existence of dropped packet problem in stepping stone detection environment. The experiment is done at the basic level of stepping stone detection environment as to prove the problems of stepping stone detection only without to combine with others complex factor of stepping stone detection. As interactive application used on almost of current stepping stone detection research, this research also doing so. Telnet application [24] used beginning from Host n0 telnet to Host n2 through Host n1 as the stepping stone. Linear topology as shown in Figure 6 explains the overall picture in the experiment.

NS-2 [25] is used as network simulator tools. NS-2 is chosen as their flexibilities to conduct networking simulation [26] and their large acceptable in research environment [27]. Figure 7 shows the real NAM interface that used in the experiment.

In this experiment, packet dropped control is used as to provide packet dropped environment. Two sets of experiments are conducted. For each phase, the following steps are executed.

1. Run NS-2
2. Get Trace Files
3. Get Arrival Time for flow k and l
4. Used DWs=5
5. Find Mini/Max Sum Ratio (MMS)

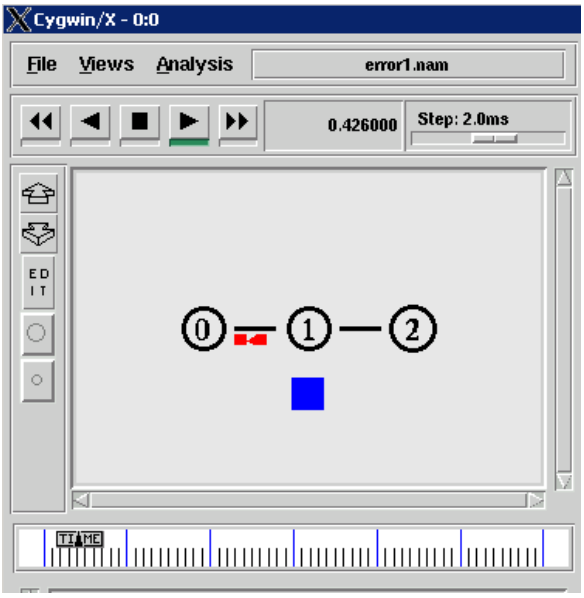

Fig. 7 Real NAM Interface

Each set below is different from the location where is the dropped packet control is located. The same properties of the experiment setting are tabulated in Table 1.

Table 1 Properties of Experiment Setting

Properties	*Setting*
Node	n_0, n_1 and n_2
Connection	TCP
Application	Telnet
Bandwidth	2Mbps
Link	Full-duplex
Delay	10ms
Type of Dropped Packet	DropTail
Arrival Packet Time Pattern	tcplib-telnet.cc

5.1 Set 1: Experiment Control

The first set is used as a control of the overall experiments. There is no dropped packet control used on flow. This set will be used as comparison of dropped packet in other set.

5.2 Set 2: Dropped Packet on l

This second set of experiment contains dropped packet properties on the simulation. The properties of the set are tabulated in Table 2

Table 2. Dropped Packet Properties

Properties	Setting
Rate	25%
Random Variable	Uniform
Unit	Packet

These dropped packet properties are used in NS-2 simulation. The generation of dropped packet comes from the used of Error Model [28] in NS-2.

6. Results

Results for both experiments are divided into two sub-topics as shown in the following Sub-sections.

6.1 Set 1: Experiment Control

As shown in Figure 8, the overall packet arrival time for both flows shows a similar pattern.

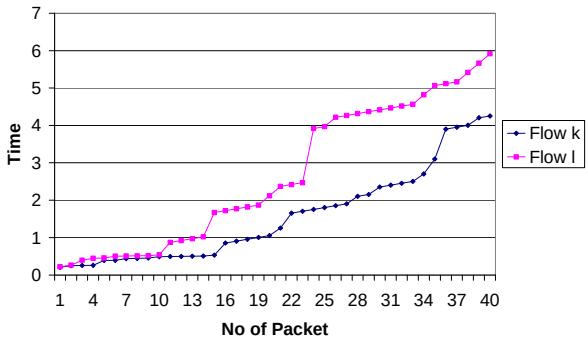

Fig. 8 Packet Arrival Time for flow k and l

The packet arrival time for flow l is differing from flow k just a few milliseconds ahead. This is because of packet is flowing through flow k before it is forwarded to flow l. The similarities of both flows occur because there is no dropped packet in this control set experiment.

6.2 Set 2: Dropped Packet on l

Set 2 provides result where dropped packet occurred.

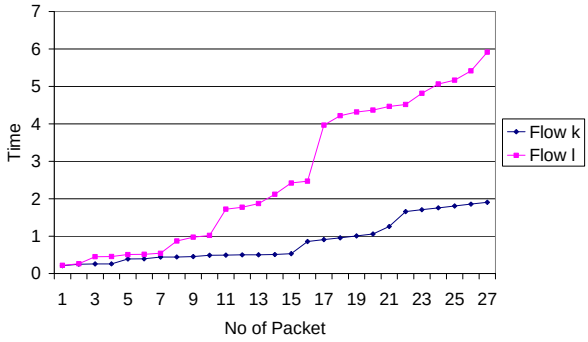

Fig. 9 Packet Arrival Time for flow k and l

Differ from previous experiment set, this second set shows the overall pattern of arrival time for both flows is quite different from previous set of experiment. It is because, in this set, dropped packet is set in flow l. In this experiment, dropped packet is just dropped.

From both set of experiment, it is shown that when there is a dropped packet involved in flow l, the overall arrival time of packet is different each others. Next Section will discuss detail about cause of the different.

7. Analysis

Analysis for result focuses on the similarity of the arrival time for both set of experiments. Analysis of the similarity is chosen because from there, we will know how the two set of data are similar or differ from each other. In this research, the data is the packet arrival time for flow k and l. there are several formulas that can be used to find the similarity of two data. There are Statistical Correlation (STAT) [35], Normalized Dot Product (NPD) [5], and Min/Max Sum Ratio (MMS) [5]. MMS is chosen because it is simple yet effective to finding the similarity of two data [5]. MMS (1) is defined as a one simple matrix to quantitatively express the similarity between two vectors by the ratio between the summation of the minimum elements and the summation of the maximum elements [5]. The formula is shown as in Equation 1.

$$\mathrm{MMS} = \frac{\sum \min(x, y)}{\sum \max(x, y)} \qquad (1)$$

In this formula, the value of MMS is between 0 and 1. Only when x = y, MMS has the value of 1. Therefore, MMS is likely to exhibit unique perfect correlation. This formula is important to determine the similarities between flow k and l. Figure 10 shows the MMS for flow k and l. Set 1 represents MMS for experiment control and Set 2 for dropped packet.

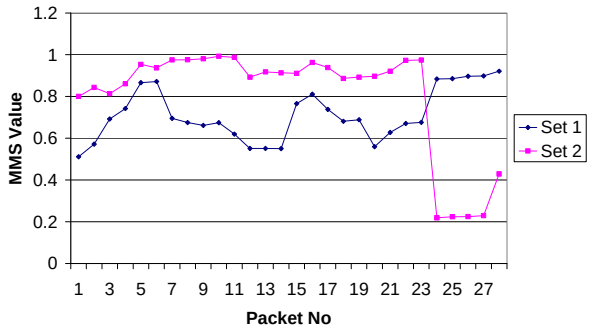

Fig. 10 MMS of Experiment Set 1 and Set 2

Figure 10 shows that MMS for Set 1 (no dropped packet) is more stable compare to Set 2 (dropped packet involved). Moreover, MMS for Set 1 is nearly reached at 1 at the end of the experiment. MMS for Set 2 on the other hand falls into nearly 0.2. As conclusion, it shows that the similarities between two flows are influenced; when dropped packet occurs. This experiment shows us how dropped packet can influence the overall effort to detect the stepping stone.

8. Conclusion and Future Works

Beginning from previous research that assumed dropped packet will not affect their stepping stone detection research to a theory that states dropped packet need to be solved by Xiang [11], this research successfully proves that dropped packet problem is really needed to be solved. Results from experiment through the usage of NS-2 clearly show that dropped packet problem gives effect to the current detection stepping stone methods.

For the future works, we recommend three suggestions. Firstly, as mentioned before that dropped packet on l causes retransmission of a new packet occurs continuously until ACK is received or the connection is expired within ct [23]. For that reason, one mechanism to detect dropped packet should be developed first before stepping stone detection can be done successfully. Secondly, the size of windows used (DW) to capture the packet should be studied further as to determine that the best differentiate process can be done.

Last but not least, the type of information used and chose also need to be tested as to make sure that it is less or independent of dropped packet problem. As to differentiate between stream k and l, we need to revise the method to differentiate them as to make sure that it is robust to dropped packet problem.

Acknowledgment

The authors would like to express their special thanks to Minister of Higher Education Malaysia for their scholarship for one of an author PhD studies in this field.

References

[1] CERT, “Explosion of Incidents”, http://www.cert.org, accessed Jun 2007.

[2] Y. Zhang and V. Paxson, “Detecting Stepping Stones”, Proc. 9th USENIX Security Symposium, 2000, pp. 67-81.

[3] S. Staniford-Chen and L.T. Herberlein, “Holding Intruders Accountable on the Internet”, Proc. 1995 IEEE Symposium on Security and Privacy, 1995, pp. 39-49.

[4] K. Yoda and H. Etoh, “Finding Connection Chain for Tracing Intruders”, Proc. 6th European Symposium on Research in Computer Security (LNCS 1985), 2000, pp. 31-42.

[5] X. Wang, D.S. Reeves, and S.F. Wu, “Inter-packet delay based correlation for tracing encrypted connection through stepping stone”, Proc. 7th European Symposium on Research in Computer Security (ESORICS 2002), 2002, pp. 244-263.

[6] X. Wang and D.S. Reeves, “Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays”, Proc. 10th ACM Conference on Computer and Communication Security (CCS 2003), 2003, pp. 20-29.

[7] W.T. Strayer, C.E. Jones, I. Castineyra, J.B. Levin, and R.R. Hain, "An integrated architecture for attack attribution", BBN Technologies, Tech. Rep. BBN REPORT-8384, 2003.
[8] D.L. Donoho, A.G. Flesia, U. Shankar, V. Paxson, J. Coit, and S. Staniford, "Multiscale stepping-stone detection: Detecting pairs of jittered interactive streams by exploiting maximum tolerable delay", Proc. 5th International Symposium on Recent Advances in Intrusion Detection (RAID 2002), 2002, pp. 49-64
[9] A. Blum, D. Song, and S. Venkataraman, "Detection of interactive stepping stones: Algorithm and confidence bounds", The 7th International Symposium on Recent Advances in Intrusion Detection (RAID 2004), 2004.
[10] L. Zhang, A. G. Persaud, A. Johson, Y. Guan, "Stepping Stone Attack Attribution in Non-Cooperative IP Networks", in Proc. Of the 25th IEEE International Performance Computing and Conference (IPCCC 2006), 2006.
[11] X. Wang, Tracing Intruders behind Stepping Stones, doctoral dissertation, North Carolina State University, Raleigh, 2004.
[12] M. Venkateshaiah and M. Wright, Evading Stepping Stone Detection Under the Cloak of Streaming Media, Tech. Report, Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019, 2007.
[13] M.N. Omar, M.A. Maarof and A. Zainal, The Optimization of Stepping Stone Detection: Packet Capture Steps", Jurnal Teknologi, vol. 44, no. (D), Jun 2006, pp. 1-14.
[14] M.N. Omar, M.A. Maarof and A. Zainal, "Identification Steps For The Optimization of Stepping Stone Detection", Proc. ECTI Transaction on Electrical / Electronic and Communication (ECTI 2004), 2004.
[15] M.N. Omar, M.A. Maarof and A. Zainal, "Comparison Steps for The Optimization of Stepping Stone", Proc. Telematics System, Services, and Application 2004 (TSSA 2004), 2004.
[16] IETF Draft, "SSH Protocol Architecture, draft IETF document, "http://www.ietf.org/internet-drafts/drafft-ietf-secsh-architecture-16.txt, accessed June 2007.
[17] RFC 739, Transmission Control Protocol, University of Southern California, California, 1981.
[18] M.P. Clark, Data Networks, IP and the Internet Protocols, Design and Operation, Wiley, New York, 2003.
[19] IETF Draft, "SSH Transport Layer Protocol, document, "http://www.ietf.org/internet-drafts/draft-ietf-secsh-transprt-18.txt, accessed June 2007.
[20] J. Yang, and S.S. Huang, "Matching TCP/IP to Detect Stepping-Stone Intrusion", International Journal of Computer Science and Network Security (IJCSNS), vol. 6, no. 10, Oct. 2006, pp. 269-276.
[21] K.H. Yung, "Detecting Long Connecting Chains of Interactive Sessions", Proc. International Symposium on Recent Advance in Intrusion Detection, 2002, pp. 1-16.
[22] J. Yang, and S.S. Huang, "A Real-Time Algorithm to Detect Long Connection Chains of Interactive Terminal Session", Proc. 3rd International Conference on Information Security (Infosecu '04), 2004, pp. 198 – 203.
[23] C. Kozierok, "TCP/IP Guide", No Strach Press, October 2005.
[24] The Network Simulator – ns-2, "http://www.isi.edu/nsnam/ns/", accessed July 2007.
[25] I. Downard, Simulating Sensor Networks in NS-2, Naval Research Laboratory, Code 5523, 4555 Overlook Ave, Washinton DC, 20375-5337, 2003.
[26] G. Wittenburg, and J. Schiller, "Running Real-World Software on Simulated Wireless Sensor Nodes", Proc. ACM Workshop on Real-World Wireless Sensor Networks (REALWSN'06), 2006, pp. 7-11.
[27] S. Kurkowski, T. Camp, N. Mushell, and M. Colagrosso, "A Visualization and Analysis Tool for NS-2 Wireless Simulation", Proc. IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication System (MASCOTS), 2005, pp. 503-506.
[28] Y. Nanyin, and G.H. Michael, "Implication of Dropping Packets from the Front of a Queue", IEEE Transaction on Communications, vol. 41, no. 6, June 1996, pp. 846-851.
[29] M. Ibrahim, and G. Liang, Differentiated Predictive Fair Service for TCP Flows, Technical Report, 2000-12, Boston University, Boston, MA, USA, 2000.
[30] M.H. DeGroot, Probability and Statistics, Addison-Wesley Publication Company, 1989.

Mohd Nizam Omar received the B.S(Hons) and MSc in Computer Science from Malaysia University of Technology in 2002 and 2005, respectively. Currently, he is working with Faculty of IT and Computer Science, Universiti Utara Malaysia (UUM). He is a PhD candidate at School of Computer Sciences USM.

Lelyzar Siregar obtained her Bachelor degree in Information System from Gunadarma University, Jakarta, Indonesia, and Master of Information System from Gadjah Mada University, Yogyakarta, Indonesia.
Currently, she is registered as a PhD fellow at the School of Computer Science, Universiti Sains Malaysia. Her research is Quality of Service for IPv6

Rahmat Budiarto received B.Sc. degree from Bandung Institute of Technology in 1986, M.Eng, and Dr.Eng in Computer Science from Nagoya Institute of Technology in 1995 and 1998 respectively. Currently, he is an associate professor at School of Computer Sciences as well as the deputy director of National Advanced IPv6 (NAv6) Center, USM. His research interest includes IPv6, network security, Intelligent Systems. He was chairman of APAN Security Working Group.