Intelligent Host-based Stepping Stone Detection Approach

Mohd Nizam Omar and Rahmat Budiarto

Abstract— **This paper intends to introduce an implementation of a novel Self-Organization Map (SOM) in Host-based Stepping Stone Detection (SSD). Previous works have introduced Artificial Intelligence (AI) approaches such as Artificial Neural Network (ANN), however we found that the approaches are complex due to the requirement of variable to be known and tested to detect a stepping stone. SOM provides unsupervised capability in learning process. This feature helps to decrease the complexity of the AI approach. Moreover, this paper uses packet arrival time instead of Round Trip Time (RTT), which in turn reduces CPU usage as well as improves network load balancing. Through a series of real-time experiment, we show that our novel SOM approach is able to detect the stepping stone by only looking into the number of involved connection chain. In addition, the usage of SOM in Network-based SSD had been proven can detect the stepping stone in our previous research paper.**

Index Terms—**Self-Organizing Map (SOM), Stepping Stone Detection, Stepping Stone Intrusion, Tracking Intruder.**

I. Introduction

Internet has become more important than before however, 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 has given some advantages for attacker to hiding their track.

According to Zhang and Paxson [2], Stepping Stone Detection (SSD) is a process to find a connection chain of stepping stone. Since the first research on SSD by Staniford-Chen and Heberlein [3] to current research by Wang et al. [4], many related issues appear. For example, research by Wang [5] provides an active SSD system. Research by Yoda and Etoh [6] introduced SSD that is robust on encrypted connection. Research by Zhang et al. [7] on the other hand focused on solving active perturbation problems such as chaff, and delay. Avrim Blum et al. [8] in their research try to detect stepping stone by introducing the confident bound. These researches focus on statistical approaches to detect stepping stone.

Recently, realizing the importance of AI techniques, research on SSD begin to use AI techniques such as Neural Network (NN) [9], data mining [10], and so forth. The usage of AI techniques can be considered as a mean to overcome the problem that exists on statistical-based approaches such as high CPU usage and network occupation. Moreover, the usage of AI techniques hopefully can overcome the active perturbation problems that had become a focus to most researchers in this field.

The usage of Self-Organizing Map (SOM) on this research can be considered from the unsupervised capabilities as compared to other NN approaches [11]. By using SOM, Direct Stepping Stone (DSS) and Indirect Stepping Stone (ISS) can be identified easily. The introduction of DSS and ISS hopefully becomes the beginning point for better stepping stone detection researches. From the results obtained from the experiment, it is proved that SOM will give only one direction on DSS and more than one direction on their node if there is ISS.

The rest of this paper is structured as follows. Section 2 gives all of the research terms that used in this paper. Section 3 explains host-based SSD. Section 4 describes Self-organizing map (SOM). In Section 5, we discuss further on stepping stone detection with SOM before the Data Reduction and Pre-procession discussed in Section 6. Section 7 explains the overall experiment process then, followed by their result. Finally, we summarize the whole works in the conclusion sections and give future works in Section 9.

II. Research Terms

Before we start more detail on focused discussion, there are several research terms or terminologies used that need to know. A person or program can login to a network from Host 1 to Host n through Host i - 1,… i, i + 1,…, and Host n as shown in Figure 1.

Connection occurs whenever host logs from one host to another hosts. A connection is when given n host H1, Hi-1, …, Hi, Hi + 1,…, Hn is a sequence of connection as a chain C = <C1, Ci-1,…, Ci +1,Cn> whereby Ci is a connection from Host i to Host Hi + 1, for i = 1, …, n – 1. Downstream is a path of user's login directions (according to arrow) or otherwise if the arrow direction goes on other way, it is called as upstream.

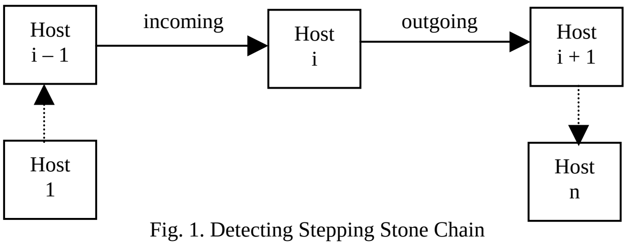

Fig. 1. Detecting Stepping Stone Chain

Manuscript received July 1, 2009. This work was supported in part by the Universiti Utara Malaysia and Minister of Higher Education Malaysia for the scholarship for one of an author.

Mohd Nizam Omar is a Ph.D candidate at School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, MALAYSIA (corresponding author to provide phone: 60175387991; fax: 6047331315; e-mail: niezam.cod07@student.usm.my).

Rahmat Budiarto is an Associate Professor at School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, MALAYSIA (e-mail: rahmat@cs.usm.my).

Other terms that need to be rectified here are Direct Stepping Stone (DSS) and Indirect Stepping Stone (ISS). DSS means that the connection of the stepping stone has been established by direct hosts before the target hosts. For example, in Figure 1. DSS is connection established between Host i and Host i-1. ISS on the other hand is a connection that established using more than one host before reaching the target hosts.

Our previous research had divided SSD algorithm into three different parts. There are capturing [12], identifying [13] and comparing [14]. However, in case of host-based stepping stone detection, only one host, incoming and outgoing packet streams of the host is involved. Therefore, if we refer to Figure 1, Host i, incoming and outgoing packet of Host i is involved in the host-based stepping stone detection. We will explain about host-based stepping stone detection in details on the next sections.

III. HOST-BASED STEPPING STONE DETECTION

Many researchers such that focus on detecting compromised hosts have already being conducted by [15], [16] and [17]. The compromised hosts contain connection chains of multiple hosts. SSD is a system that has capability in identifying these connection chains. Wang et al. [18] divides tracing approaches into host-based and network-based. Each of these categories can further be classified into active or passive. Table 1 shows the overall classification of existing tracing approaches.

Table 1. Existing Tracing Approach

	Passive	Active
Host-based	DIDS CIS	Caller ID
Network-based	Thumbprint ON/OFF Deviation	IDIP SWT IPD

Snapp et al. [19] develop Distributed Intrusion Detection System (DIDS), a host-based tracing mechanism that keep track of user in the network and account for all activities to network-wide IDS. Research by Jung et al. [20] also studies a host-based and passive based tracing mechanism called Caller Identification System (CIS).

Caller ID, research conducted by Air Force [21] is an host-based approach. Both DIDS and CIS use passive approaches where network packets need to be captured continuously. However, it is different from Caller ID where tracing is executed when an intrusion is occurred.

For network-based tracing, Staniford-Chen and Herberlein [16] are the pioneer. In this approach, a small quantity of information to summarize a connection is used. Then, research by Zhang and Paxson [15] correlated the connection based on the distinctive timing characteristics of interactive traffic. Yoda and Etoh [6] introduce correlation schemes that count the minimum average of delay gaps between the packet streams of two TCP connections that is known as deviation. All of these network-based tracing are passive because of their passively behaviors of monitoring the traffic. It is different with active network-based tracing such as Intrusion Identification and Isolation Protocol (IDIP) and Sleepy Watermark Tracing (SWT), both use active approaches which tracing only executed when the intrusion is occurred.

IDIP developed by Schanckenberg [22] uses active approaches to trace the incoming path and source of intrusion. Alternatively, research by Wang et al. [18] proposed a framework called ‘sleepy’ because it does not introduce overheads when no intrusion is detected. Research by Wang [23] also chooses active network-based stepping stone tracing by developing inter-packet delay concepts validated through encrypted connection.

Host-based approaches have the advantages from the accurate tracing methods. By looking into its audit logs especially its incoming and outgoing flows of network connection [24-25], the existence of stepping stone connection can be identified.

Network-based approached on the other hand has the capability to tracing an intruder without participation of monitored hosts. However, there is the possibility that network information can be changed or spoofed easily. Research such as [26-28] and attempt to solve the problem on spoofed information on network flows. Therefore, to determine that the information is not spoofed, the information on host-based approaches is used. Hybrid approaches as published [29] as mutual cooperation between host-based and network-based SSD are used to solve this problem. By considering the advantages from the accurate tracing methods on host-based stepping stone detection, this research intends to use one of the AI techniques as known as SOM as to detect the stepping stone in the host-based environments. The usage of SOM techniques on network-based SSD was successfully described in previous research [43].

IV. SELF-ORGANIZATION MAP IN ARTIFICIAL INTELLIGENT

Artificial Intelligent (AI) according to [30] can be described as one of computer science fields that focused on the automation of intelligent behaviors. Artificial Neural Network (ANN) or Neural Network (NN) on the other hand can be defined as an effective approach for classification [9]. Kohonen Self-Organization or Self-Organization Map (SOM) is one type of NN besides of Feedforward Neural Network, Radial Basis Function (RBF) network, Recurrent Network and so forth [31]. One of the interesting capability of SOM is it can classify data without a supervision [32].

AI concepts have been applied into many fields in computer sciences. One of computer sciences field that applied AI techniques is network security. In this case, AI has been used in Intrusion Detection System (IDS) [33], Firewall [34], and so forth. In SSD fields itself, the usage of AI techniques has attracted many researcher. A researcher conducted by Jianhua and Shou-Hsuan [10] used data mining methods to find the round-trip time from the time-stamps of TCP send and echo packets. On the other hand, research by Han-Ching and Shou-Hsuan [9] choose NN techniques as to detect stepping stones.

Realize that SOM has a special capability from it unsupervised capabilities to classify data, we are intent to apply SOM as to detect stepping stone. Moreover, if the successful results from both Lichodzijewski et. al [35] and Albert et. al [36] have been referred, SOM theoretically can also be used to solve the detecting stepping stone problems. The success or fail of SOM to solve stepping stone problems will be answered in Section 8 (Result).

V. Stepping Stone Detection with Self-Organizing Map

The main goal of this research is to automate the process of detecting stepping stone efficiently. In order to develop this, characteristic of Indirect Stepping Stone (ISS) connections and Direct Stepping Stone (DSS) connections need to be identified. ISS connection is defined as connection that go through more than one host before reach the target host compare to DSS connections where the connection is direct or on other hand a connection is from the closest host. Intrusion can be executed through ISS or DSS. Intrusion through DSS is not become a problem because tracing can be done easily. However, intrusion on ISS becomes a problem because attacked host only can trace the intrusion only to the nearest host.

Compared to network-based stepping stone approaches, host-based stepping stone approaches has limited source of data to determine whether it is ISS connection or DSS connection. Host-based stepping stone approaches only have the information on incoming and outgoing at the host itself.

Research by Kwong [37], and Jianhua and Shou-Hsuan [38], [10], [39] define that connection used more than three hosts is categorized as stepping stone connections. In fact, how many hosts that have been compromised are not accurately determines that the connection is intrusion connections or non-intrusion connection. Only by using tool like IDS can determine either the connection is intrusion stepping stone connections or non-intrusion stepping stone connections. In this paper, it is assumed that a connection being compromised by more that two. Currently, it is suggested that a connection compromised by three hosts is considered as a stepping stone connections.

Kwong [37] explains the relationship between time and packet data when interactive sessions are occurred. As shown on Figure 2, Host 1 issues a character packet containing letter l. Host 2 forwards the packet to the final Host 3. After executing the packet, Host 3 sends reply echo back to Host 1 through Host 2. In this case, Host 1 logs three packets at different time t_q, t_a and t_e.

SOM plays the important roles to classify packet data that in and out through a host to obtain the information on the existing of stepping stone connections.

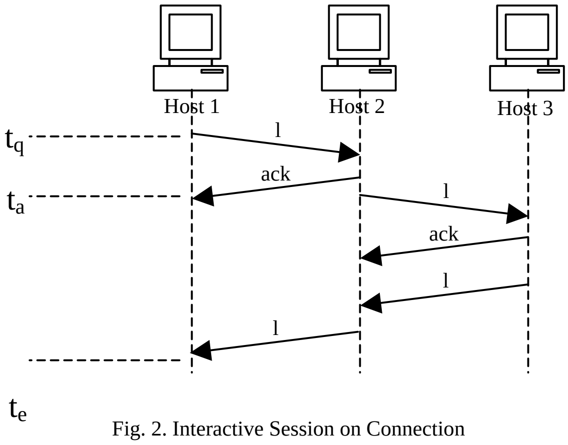

Fig. 2. Interactive Session on Connection

Besides its unsupervised learning capabilities, SOM also has an efficient update scheme and the ability to express topological relationships. This behavior makes it is very convenient for expressing the different between ISS connections and DSS connections. The simple hypothesis is that the stepping stone connections will be look in sparse regions of the topology of SOM visualizations.

VI. Data Reduction and Pre-processing

To determine that only an important data is used by SOM, data reduction and pre-procession steps are executed. For the data reduction, packet data will be filtered towards to output Telnet-based data that occurred on the host. Data is further focused on the time information that captured by Wireshark [40] when there is incoming packet data. Compared to previous research that chosing many type of data to detect the stepping stone, the usage of only time type of data can be considered as an effort to reduce the number of overhead on processing. This will make SOM easier to execute the expression.

VII. experiment

As describe before, the experiment is divided into two main parts. First, detecting stepping stone connections using SOM by only involves DSS connection. Then it is followed by detecting stepping stone connections using SOM by involving more than one stepping stone. Although there are different set of experiment, both parts have the same experiment flows. The different can be looked only on the data input either for DSS or ISS connection detections.

Experiment begins with the usage of Telnet Scripting Tool v.1.0 [41]. This scripting tool is used as to guarantee the uniform patterns of telnet operations during the execution of the experiment. Moreover, there is no such dataset that suitable to use in this research. From our observation, previous researches do not publish their dataset because of the security and privacy concerns. Previous research such as Jianhua and Shou-Hsuan [38 - 39] also has their own dataset in their research.

Experiment is run in controlled environment (in LAN) as to avoid any interference with outside networks. Experiment is firstly run on DSS connections and then followed by stepping stone connection that involved more than one stepping stone connections or on other hand, ISS. During the experiment, Wireshark will be used as to capture network packets that incoming and outgoing on the monitored hosts.

Before Wireshark is executed, filter has been set so that only captures the needed network packets (Telnet-based packet data).

After Telnet Scripting Tools finished its execution, the packet that had been captured is converting into text-based form. This is as to provide next processes to get the appropriate information that needed. In this process, only the time information is obtained. This time information from the experiment is transferred into m type of file as to be use in Matlab 6.1[42] software.

In the Matlab 6.1 software, the time information is used as the input to create, train and lastly plots the SOM. Matlab gives just a simple solution to create, training and the visualize SOM. The result from the visualization will be taken as the result on this research and will be discussed in next sections.

VIII. RESULTS

The result can be divided into two categories and a comparison together with an explanation between these categories will be discussed in this sections. In DSS connection, the arrival time in monitored host is shown in Figure 3. From the graphs, it shows that packet data will arrive at the host with almost similar pattern of arrival time.

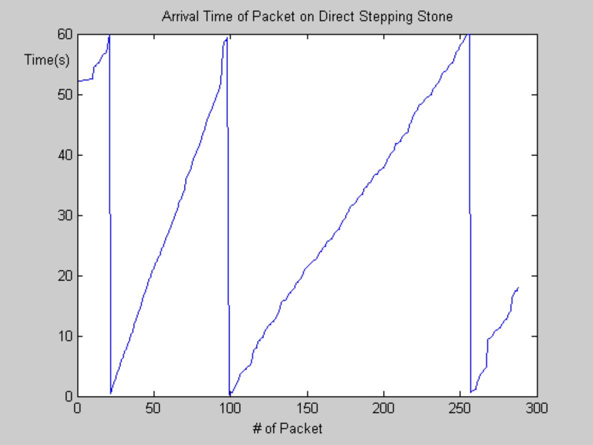

Fig. 3. Arrival Time of Packet on Direct Stepping Stone

In this graph, packet arrives at host between 100 and 150 packet in a minute. The arrival can be looked as smooth without any interruption.

By applying SOM techniques on the same data, it shows that there is only one possible direction can be followed. It is done by train the data using 100 times of epochs. Figure 4 shows the overall results.

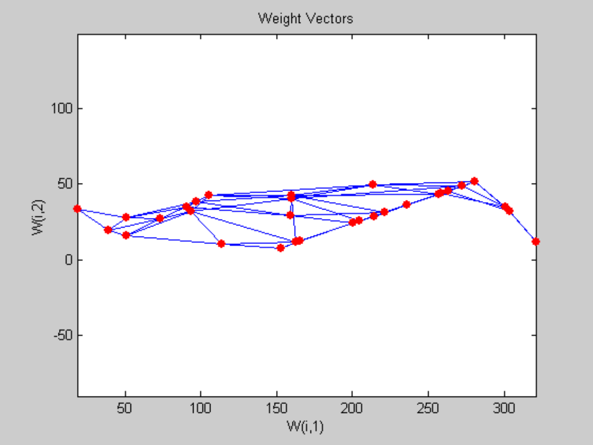

Fig. 4. Node of Arrival Time By Using SOM Techniques

It is different to ISS connections. By looking at Figure 5 below, it shows that the arrival time of a packet is unpredictable. Although the packet data is only monitored in one host, the arrival time of packet is unpredictable. This is support the theory that describe as in Figure 2. The arrival time is not only involved to the nearest host but also forwarded packets from the other hosts.

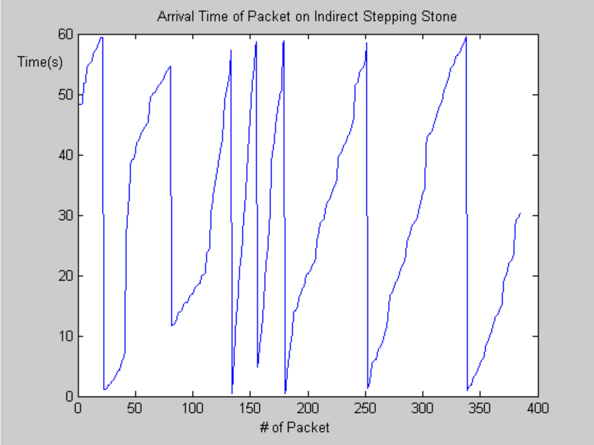

Fig. 5. Arrival Time of Packet on Indirect Stepping Stone

The arrival time here can be observed between 100 packets per minutes, 50 packets per minutes and less than 50 packets per minutes. This shows the unpredictable of packet arrival time compared to DDS connections.

By applying SOM techniques, the node of arrival time is drawn as shown in Figure 6. By using the same of epochs used by DSS, Figure 6 shows the relationship of the node. In this figure, it is shown that there is three possible ways of node that can be followed. This on the other hand also reveals that there are three stepping stones used in this ISS connection.

For the comparison in both graphs on SOM techniques, it is clearly shown that SOM can be used to detect the DSS and ISS. As shown in Figure 4 and Figure 6, node of SOM in DSS gives only one possible direction can be followed. It is different in ISS where there is three possible directions can be found in the graph.

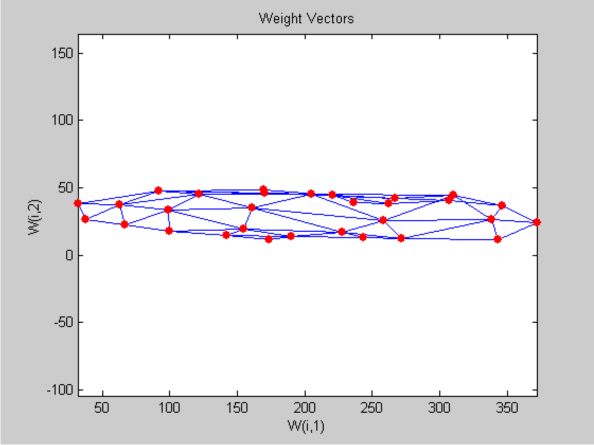

Fig 6. Node of Arrival Time By Using SOM Techniques

As summary, SOM techniques successfully differentiate between direct and indirect stepping stone connection and at the same time reveal the number of connection chains that have been involved. The differentiation between DSS and ISS become an important factor to provide a better stepping stone detection.

IX. CONCLUSION AND FUTURE WORKS

The usage of AI techniques on stepping stone detection researches is a new field that needs more exposure. Although there are a few researches that apply AI techniques, our AI approaches that introduces to differentiate between DSS and ISS is the pioneer research towards to provide better stepping stone detection.

Through the experiment, it is shown that SOM can be used in host-based stepping stone detection. By SOM capabilities, stepping stone on host-based can be identified easily. Unsupervised capabilities that embedded in SOM itself makes stepping stone detection is more easily. From the result obtained form the experiment, SOM successfully distinguishes the DSS and ISS.

After SOM had been proven can detect the ISS connection through the learning process of DSS on Host-based environment, our research will expand this discovery into Network-based environments. The requirement to involve in network-based environment can be looked as the important of network-based SSD itself. Previous research on network-based SSD that gave more focus on network-based should not be taking for granted. The capabilities of SOM will be more tested if applied in network-based SSD because there are a lot of variable needs to detail, observe and study before it can be used easily.

On the host-based site, there are another works can be done. The testing on stepping stone perturbation is not tested in this research. For that problem, this research will be executed the testing on perturbation such as chaff, delay and dropped packet problems. All of these problems have been identified from the previous researches can influence the statistical-based stepping stone detection. Future works also can be looked as to upgrade the overall output on this paper towards to provide a real-time stepping stone detection.

ACKNOWLEDGMENT

The authors would like to express their special thanks to Minister of Higher Education Malaysia and Universiti Utara Malaysia for their scholarship for one of an author to pursue PhD study 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, Denver, CO, 2000, pp. 67-81.

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

[4] X. Wang, S. Chen, and S. Jajodia, “Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems”, Proceeding of the 2007 IEEE Symposium on Security & Privacy (S & P 2007), May 2007.

[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), Zurich, Switzerland, Oct. 2002, pp.244-263.

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

[7] L. Zhang, A. G. Persaud, A. Johnson, and Y. Guan, “Detection of Stepping Stone Attack Under Dalay and Chaff Perturbations”, 25th IEEE International Performance Computing and Communications Conference (IPCCC 2006), Phoenix, USA, April 2006.

[8] A. Blum, D. Song, and S. Benkataraman, “Detection of Interactive Stepping Stone: Algorithm and Confidence Bounds”, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 3224/2004, pg. 258-277, October 1, 2004.

[9] W. Han-Ching, and S. H. Shou-Hsuaan, “Performance of Neural Networks in Stepping-Stone Intrusion Detection”, IEEE International Conference on Networking, Sensing and Control 2008 (ICNSC 2008), Sanya, 6-8 April 2008, pp. 608-613.

[10] Y. Jianhua, and S. H. Shou-Hsuan, “Mining TCP/IP packet to detect stepping-stone intrusion”, *Computer & Security,* Vol. 26(7-8), 2007, pp. 479-484.

[11] T. Kohonen, “The Self-Organizing Map”, *Proceedings of the IEEE,* Vol. 78, Issue: 9, pp. 1464-1480, 1990.

[12] M. N. Omar, M. A. Maarof and A. Zainal, “The Optimization of Stepping Stone Detection: Packet Capture Steps”, Jurnal Teknologi, 44(D) Jun 2006: 1 – 14, Universiti Teknologi Malaysia.

[13] M. N. Omar, M. A. Maarof and A. Zainal, “Identification Steps For The Optimization of Stepping Stone Detection”, ECTI Transaction on Electrical / Electronic and Communication (ECTI 2004), Pattaya, Thailand.

[14] M. N. Omar, M. A. Maarof and A. Zainal, “Comparison Steps for The Optimization of Stepping Stone”, Telematics System, Services, and Application 2004 (TSSA 2004), Bandung, Indonesia.

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

[16] Stuart Staniford-Chen and L. Todd Herberlein, “Holding Intruders Accountable on the Internet”, Proc. 1995 IEEE Symposium on Security and Privacy, Oakland, CA, 1995, pp. 39-49.

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

[18] X. Wang, D. Reeves, S. F. Wu, “Tracing Based Active Intrusion Response”, Journal of Information Warefare, Volume 1, Issue 1, September 2001, pg. 50-61.

[19] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, L. T. Heberlein, C. Ho, K. N. Levitt, B. Mukherjee, S. E. Smaha, T. Grance, D. M. Teal and D. Mansur, “DIDS (Distributed Intrusion Detection System) – Motivation, Architecture and Early Prototype, Proceeding 14th National Computer Security Conference, pg. 167 – 176, 1991.

[20] H. T. Jung, H. L. Kim, Y. M. Seo, G. Choe, S. L. Min, and C. S. Kim, “Caller Identification System In The Internet Environment”, Proceedings of 4th USENIX Security Symposium, 1997.

[21] S. Wadell. Private Commucication. 1994.

[22] D. Schnackenbert. “Dynamic Cooperating Boundary Controllers.

[23] X. Wang, D. S. Reeves, and S. F. Wu, “Inter-Packet Delay Based Correlation for Tracing Encrypted Connections Through Stepping Stones”, In. D. Gollmann, G. Karjoth and M. Waidner, editors, 7th European Symposium on Research in Computer Security (ESORICS 2002), October 2002.

[24] Telnet Protocol Spesification, “http://www.ietf.org/rfc/rfc1572.txt”, accessed July 2007.

[25] T. Ylonen, “SSH Protocol Architecture, draft IETF document, “http://www.ietf.org/internet-drafts/drafft-ietf-secsh-architecture-16.txt, accessed June 2007.

[26] M. N. Omar, L. Siregar, and R. Budiarto, “Dropped Packet Problems in Stepping Stone Detection Method”, International Journal of Computer Science & Network Security (IJCSNS), Vol. 8, No. 1, February 2008, pp. 109-115.

[27] X. Wang and D. S. Reeves, “Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays”, Proc. 10th ACM CCS 2003, Washington DC, USA, Oct. 2003.

[28] M. Venkateshaiah, “Evading Existing Stepping Stone Detection Methods”, Master Thesis, University of Texas at Arlington, December 2006.

[29] M. N. Omar, L. Siregar, and R. Budiarto, “Hybrid Stepping Stone Detection Method”, The 1st International Conference on Distributed Frameworkds and Application, 21-22 October 2008, pp. 134-138, Universiti Sains Malaysia, Penang, Malaysia.

[30] F. L. George, *Artificial Intelligence Structures and Strategies for Complex Problem Solving,* 4th Edition, Addison-Wesley, England, 2002.

[31] Wikipedia, Artifical Neural Network, “http://en.wikipedia.org/wiki/Artificial_neural_network”, accessed Disember 2008.

[32] N. Michael, *Artificial Intelligence A Guide to Intelligent Systems,* Addison-Wesley, England, 2001.

[33] S. Jian-Hua, J. Hai, C. Hao, H. Zong-Fen, "MA-IDS: A Distributed Intrusion Detection System Based on Data Mining", *Wuhan University Journal of Natural Sciences (WUJNS),* Vol. 10, No. 1, pp. 111-114, 2005.

[34] I. Yoo, and U. Ultes-Nitsche, "Intelligent Firewall: Packet-based Recognition Againt Internet-Scale Virus Attacks", Proceeding of Conference on Communications and Computer Networks (CCN 2002), November 2002.

[35] P. Lichodzijewski, A. Z-H, Nur, M. I, Heywood, "Host-based Intrusion Detection Using Self-Organizing Maps", Proceeding of the 2002 International Joint Conference on Neural Network (IJCNN 02), USA, pp. 1714-1719.

[36] J. H. Albert, and S. S. Antti, "A Computer Host-based User Anomaly Detection System Using the Self-Organizing Map", Proceedings of IJCNN'00, pp. 411-416.

[37] H. Y. Kwong, "Detecting Long Connection Chains of Interactive Terminal Session", RAID 2002, LNCS 2516, pp. 1-6.

[38] Y. Jianhua, and S. H. Shou-Hsuan, "A Real-Time Algorithm to Detect Long Connection Chains of Interactive Terminal Session", Proceedings of the 3rd International Conference on Information Security, Shanghai, China, pp. 198-203.

[39] Y. Jianhua, S. H. Shou-Hsuan, and D. W Ming, "A Clustering-Partitioning Algorithm to Find TCP Packet Round-Trip Time for Intrusion Detection", Proceeding AINA'06, 18-20 April 2006, 6 pp.

[40] Wireshark, "http://www.wireshark.org", accessed Disember 2008.

[41] Wareseeker, "http://wareseeker.com/freeware/telnet-scripting-tool-1.0/19344/TST10.zip", accessed December 2008.

[42] H. Duane, and L. Bruce, *Mastering MATLAB A Comprehensive Tutorial and Reference,* Prentice-Hall, New Jersey, 1996.

[43] M. N. Omar and R. Budiarto, "Intelligent Network-based Stepping Stone Detection, *Journal of World Academic Science, Engineering and Technology,* Vol. 53, pp. 834 – 841.