International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:5, 2009

Intelligent Network-Based Stepping Stone Detection Approach

Mohd Nizam Omar[1] Rahmat Budiarto[2]

[1]College of Arts and Sciences, Information Technology Building, Universiti Utara Malaysia, Sintok, Malaysia
[2]School of Computer Sciences, Universiti Sains Malaysa, Penang, Malaysia
Emails: niezam@uum.edu.my, rahmat@cs.usm.my

***Abstract*—** This research intends to introduce a new usage of Artificial Intelligent (AI) approaches in Stepping Stone Detection (SSD) fields of research. By using Self-Organizing Map (SOM) approaches as the engine, through the experiment, it is shown that SOM has the capability to detect the number of connection chains that involved in a stepping stones. Realizing that by counting the number of connection chain is one of the important steps of stepping stone detection and it become the research focus currently, this research has chosen SOM as the AI techniques because of its capabilities. Through the experiment, it is shown that SOM can detect the number of involved connection chains in Network-based Stepping Stone Detection (NSSD).

***Keywords*—** Artificial Intelligent, Self-Organizing Map (SOM), Stepping Stone Detection, Tracing 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 hide 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], there are many related issue appears. 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, Researches 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 SOM on this research can be considered from the unsupervised capabilities that compared to other Neural Network approaches [11]. By using SOM, Direct Stepping Stone (DSS) and Indirect Stepping Stone (ISS) can be identified easily. Experiment results prove that SOM show its different ways of node relationship for each existence of connection chains.

The idea behind the use of SOM in the SSD environment is translated into two set of experiments, Host-based and Network-based. This is suitable as definition by Wang et al. [12] that divides the overall SSD into two categories.

The contents of this paper are divided into eight sections; in Section 2, all the terms or terminologies that used in the paper will be described precisely. In Section 3 the Stepping Stone Detection (SSD) will be explained. Section 4 will describes about Artificial Intelligent (AI) in SSD approaches. Discussion pertaining SOM as a new solution in SSD will be presented in Section 5. Then, the details of experiment and results will be provided in Section 6. In Section 7, analysis of the results will be discussed. Finally, Section 8 provides summary and future works.

II. TERMINOLOGY

Before starting on focus discussion, there are several research terms or terminologies that need to know. A person or program can log-ins to a network from Host 1 to Host n through Host i - 1,… i, i + 1,…, and Host n as shown in Figure 1.

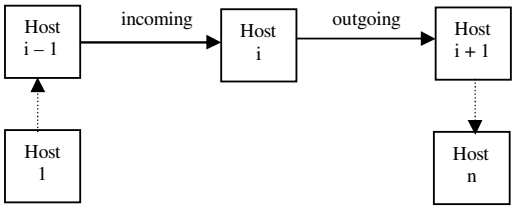

Fig. 1 Current Detecting Stepping Stone Chain Example

Connection occurred whenever a host logs from one host to another hosts. Connection is when 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},…, C_{i+1},C_n> whereby C_i is a connection from Host i to Host H_{i+1}, for i = 1, …, n – 1. Downstream is a path of user's login directions (based on arrow direction) or

otherwise, if the arrow direction goes on other way, it is called as upstream.

Two other research's 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 a connection established between Hosts i and Host i-1. ISS on the other hand is a connection established using more than one host before reaching at the target hosts. In the event of Host l to connect the Host n, for instance Host 1 using Host i-1, Host i and Host i + 1 before reaching Host n. Since Host l uses more than one host to connect to the target host, this is called ISS.

In previous research, SSD algorithm has been divided into three different parts. There are capturing [13], identifying [14] and comparing [15] parts. Capturing part is responsible to capture the network flows according to the requirement of SSD. Identifying part executes its task by identifying the items in network packet that need to be identified. Lastly in comparing part, each identified items in previous parts will be compared to each other. The output from the comparison processes is the correlation of the stepping stones. The explanation about SDD, Host-based SSD and Network-based SSD will be continued in the next sections.

III. STEPPING STONE DETECTION (SSD)

There are many researchers such as [2], [3], [6] and so forth focus on detecting compromised hosts. These compromised hosts contain connection chains of multiple hosts. SSD system is the system that identified these connection chains. Wang et al. [12] divides tracing approaches into two categories host- and network-based. Each of these categories can further classified into active or passive. Table 1 shows the overall classification of existing tracing approaches.

TABLE I
CLASSIFICATION OF EXISTING TRACING APPROACHES

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

A. Host-based SSD (HSSD)

Snapp et al. [16] 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. [17] also studies a host-based and passive based tracing mechanism but he eliminates centralized control by utilizing a truly distributed model called Caller Identification System (CIS). Through this system, each host keeps a record about its view of the login chains so far.

Caller ID, is a concept introduce by [18] is another active host-based approach. In this approach, Caller ID utilizes the same break-in techniques used by intruders to break into the hosts along the connection chains reversibly. 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.

B. Network-based SSD (NSSD)

For network-based tracing, Staniford-Chen and Herberlein [3] is the pioneering. In this approach, correlation techniques utilize a small quantity of information is used to summarize the connection. Then, research by Zhang and Paxson [2] correlate 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 known as deviation. All of these network-based tracing are passive because of its behaviors passively monitor 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 that developed by Schanckenberg [19] uses active approaches to trace the incoming path and source of intrusion. Boundary controllers collaboratively locate and block the intruder by exchanging intrusion detection information. While it does not require any boundary controllers to record any connection for correlation, its intrusion tracing is closely coupled with intrusion detections. Research by Wang et al. [12] on the other hand is a proposed framework which called sleepy mode because it does not introduce overheads when no intrusion is detected. The target will inject a watermark into backward connections of the intrusion and wakes up intermediate routers along the intrusion paths. Research by Wang [20] also chooses active network-based stepping stone tracing by develop Inter-packet delay concepts which valid through encrypted connection.

Host-based approaches have the advantages from the accurate tracing methods. By looking into its audit logs especially on its ingoing and outgoing flows of network connections [21], [22] 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 a possibility that network information can be changed or spoofed easily. This can be seen in SSD research fields itself. Research such as Omar et al. [23], Wang and Reeves [24], and Venkateshaiah [25] 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. What can be seen here is the mutual cooperation

between host- and network-based SSD. The solution on this problem had been solved by using hybrid approaches as published [26].

By looking the advantages from the independent properties of network-based approaches, this research intends to use one of the Artificial Intelligent techniques as known as Self-Organization Map (SOM) as to detect the stepping stone in the network-based environments. The usage of SOM techniques on host-based SSD was successfully described in previous research [27].

IV. ARTIFICIAL INTELLIGENT IN SSD APPROACHES

Artificial Intelligent (AI) according to Michael [28] can be described as one of computer science fields that focus 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 [29]. 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 [30]. One of the interesting capabilities of SOM is its capability in classifying data without any supervision [9]. This capability actually had been embedded in this proposed intelligent approach in this paper.

AI concepts have been applied into many fields in computer sciences. One of computer sciences fields that applied AI techniques is network security. In this case, AI has been used in Intrusion Detection System (IDS) [31], Firewall [32], and so forth. Even, in SSD fields itself, the usage of AI techniques has attracted many researchers. This can be seen at a research conducted by [10] that used data mining methods to find the round-trip time from the time-stamps of TCP send and echo packets. Research by Han-Ching and Shou-Hsuan [9] on the other hand choose NN techniques as to detect stepping stones.

From study in SSD fields, there is no such a research that used SOM as the approach to detect stepping stone. Realize that SOM has a special capability from it unsupervised capabilities to classify data, SOM is applied as to detect stepping stone. Moreover, the successful results from both Lichodzijewski et al. [33] and Albert et al. [34] 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 7 (Result and Analysis).

V. SELF-ORGANIZING MAP (SOM) AS A NEW SOLUTION

The main goal of this research is to automate the process of detecting stepping stone efficiently. The process can be achieved by using SOM, as SOM has a capability to classify data without supervision [30]. 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 does 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 to the nearest host.

If comparison to network-based stepping stone approaches have been made, 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 [35], Jianhua and Shou-Hsuan [36], [10], and Jianhua et al. [37] 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 determined that the connection is intrusion connections or non-intrusion connection. Only by using tools like IDS can determined either the connection is intrusion stepping stone connections or non-intrusion stepping stone connections. These problems will be explained and discussed in future papers and for current paper; an assumption has been made that a connection being compromised by more than two hosts can be a candidate as stepping stone connections. Previous research has suggested that three compromised hosts are intrusion stepping stone connections. However, from the literature, a minimum number of compromised hosts have the possibilities as candidates as intrusion stepping stone connections.

As explaind by Kwong [35], Figure 2 shows the relationship between time and packet data when interactive sessions are occurred. In Figure 2, Host 1 issues a character packet containing letter l. Host 2 forward 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.

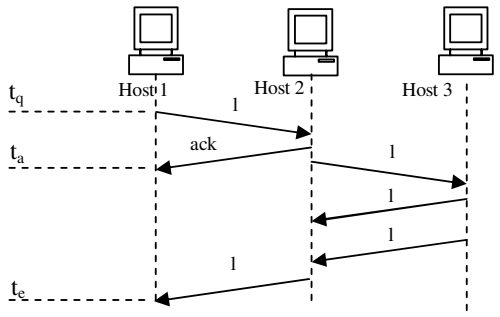

Fig. 2 Interactive session on connection

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. Firstly, SOM is used to detect and visualize the characteristic of DSS connections. This is achieved by executing SOM on normal or direct stepping stone connections. Then, SOM will be trained itself on stepping stone connection environments as to detect the existence of stepping stone connections.

SOM is used as the intelligent techniques here based on its capabilities on unsupervised learning techniques for data analysis and visualization. Moreover, SOM 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 considered in sparse regions of the topology of SOM visualizations.

VI. Experiment And Results

To determine that only an important data is used by SOM, data reduction and pre-procession steps is 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 captures by Wireshark [38] when there is incoming packet data. Compared to previous research that choosing many type of data as to detect the stepping stone, the only time type of data that used in this research can be looked as an effort to reduce the number of overhead on processing. This will make SOM easier to execute the expression. Moreover, it is possible for the future works that proposed technique will be applied on-line.

The experiment begins by the usage of Telnet Scripting Tool v.1.0 [39]. 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 an observation, previous researches do not publish their dataset because of the security and privacy concerns. Previous research such as Jianhua and Shou-Hsuan [36] also has their own dataset in their research. Moreover, using own dataset will let the flow of the network traffics is known and this is also agreed by Staniford-Chen and Herberlein [3].

Experiment is run in controlled environment (in LAN) as to avoid any interference with outside networks. During the experiment, Wireshark will be used as to capture network packets that flow on each host. Before Wireshark is executed, filter has been set so that it only captures the needed network packets (Telnet-based packet data).

After Telnet Scripting Tools finished its execution, the packet that had been captured is converted 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 in to m type of file as to use in Matlab 6.1 [40] 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, train and visualize the SOM. The result from the visualization will be taken as the result of this research and will be discussed in the next sections.

The testbed used in this paper is shown in Figure 3. From Figure 3, it is shown that only four hosts are involved. So, there are only three possible connections can be existed.

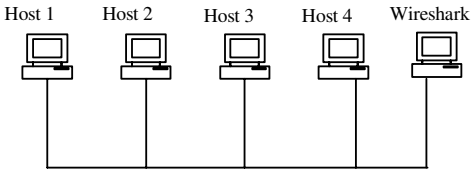

Fig. 3 Experiment Testbed

In this experiment, as shown in Figure 3, Host 1, Host 2, Host 3 and Host 4 are connected to a hub. Another one host known as Wireshark also is connected to the rest of Host. Wireshark is a host that only captures the entire of network traffic. Wireshark is located in the network segment so that it can see all of the network traffic that flow. Before the result obtained by using SOM technique is further discussed, packet arrival time on each host is shown.

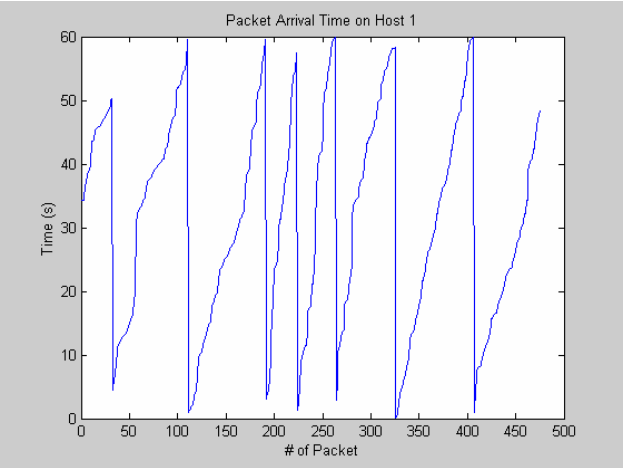

Fig. 4 Packet Arrival Time for Host 1

Figure 4 shows a packet arrival time on Host 1. From the graph, there are approximately 500 packets successfully captured by Wireshark. 50 to 100 packets are successfully being captured in a minute.

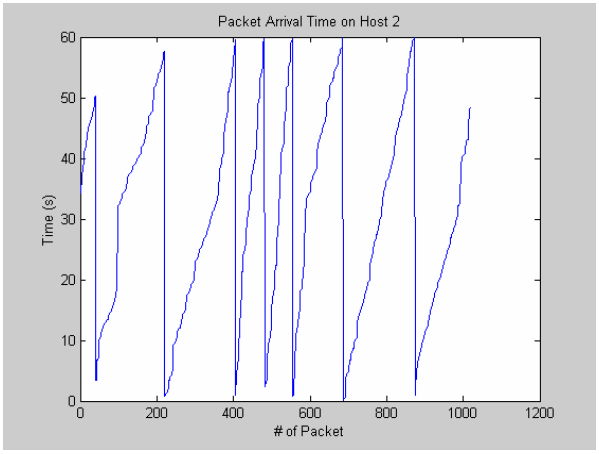

Fig. 5 Packet Arrival Time for Host 2

Compared to Figure 4, graph on Figure 5 shows the number of packet arrival that more than Figure 4. Approximately 1000 packets are successfully being captured. The number of packet that has been captured in a minute is in the range of 50 to 200 packets.

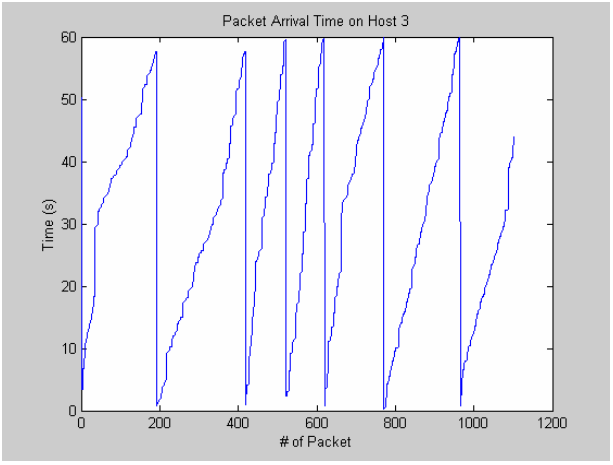

Fig. 6 Packet Arrival Time for Host 3

The number of packet that successfully being captured in Host 3 close to 1200 packets and there are between 100 to 200 packets have been captured in a minute. The details on this can be observed in Figure 6. In figure 7, the number of captured packet is about 550 packets. In a minute, there are 50 to 100 packets successfully being captured.

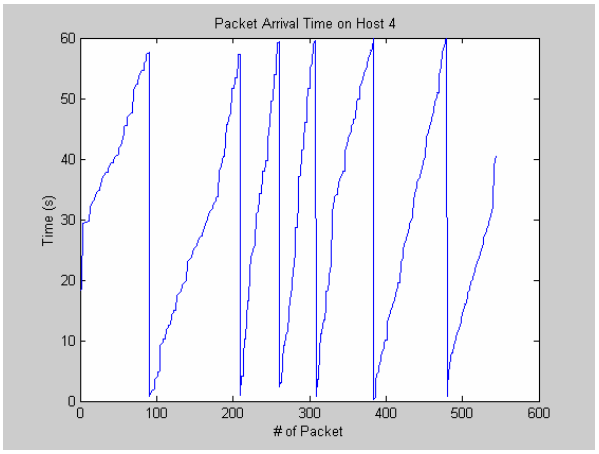

Fig. 7 Packet Arrival Time for Host 4

From the observation of the results in Figure 3 to Figure 7, the overall host can capture from 50 to 200 packets. The number of overall packets that successfully being captured is about 500 to 1000 packets. Through the information that obtained from the arrival time, there is no single information that can be obtained to detect the stepping stone.

VII. ANALYSIS

Each packet arrival time on Host 1, Host 2, Host 3 and Host 4 has been processed by SOM technique. For each involved host, 100 times of epoch is performed. Figure 7, 8, 9, and 10 show results on SOM training for Host 1, Host 2, Host 3 and Host 4, respectively. Figure 8 shows the graphs of 100 times of epoch training by using SOM technique.

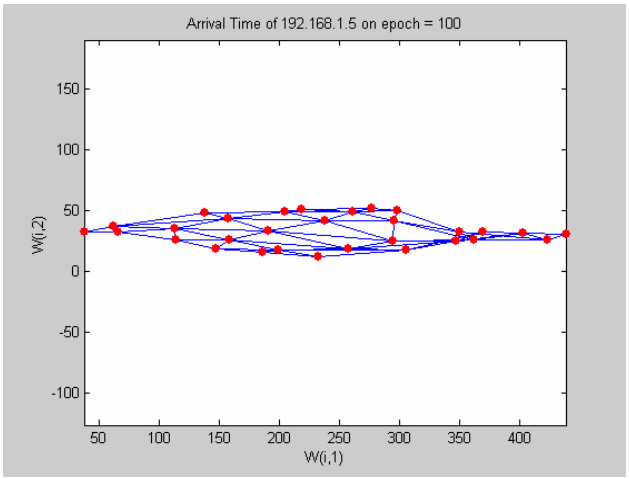

Fig. 8 Node of SOM Training on Host 1

Graph in Figure 8 gives information about the relation of node on the arrival time of packet that received on Host 1. It shows that there are three possible directions can be chosen from the beginning to the end of the graph. From discovery, three possible directions mean that there are three possible

connection chains that connected in Host 1. Figure 9 shows the graph of 100 times of epoch when arrival time of packet touch in or out at Host 2.

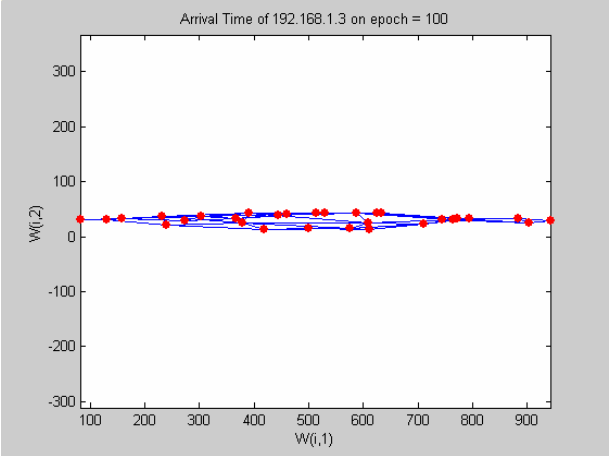

Fig. 9 Node of SOM Training on Host 2

Figure 9 shows that the relation between each node has formed into two possible ways of node. In this case it can be considered as two connection chains that exist in Host 2. Figure 10 below shows the relationship between nodes after SOM technique and 100 epochs have been applied.

The graph in Figure 10 shows that there are two possible directions can be formed. It is suitable as the Host 3 in the experiment is located after Host 1 and Host 2 where two connection chains exist. Figure 10 shows the node after SOM technique has been applied to the arrival time on Host 4. The same epoch value, 100 is also being performed.

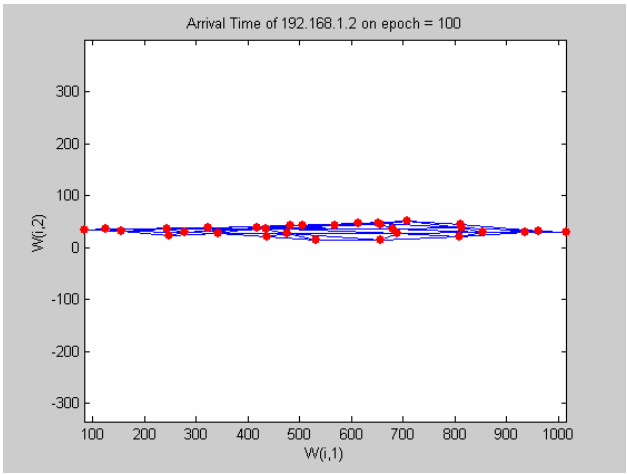

Fig. 10 Node of SOM Training on Host 3

Through Figure 11, the relationship between nodes in Host 4 forms approximately three possible directions. It is suitable by the location of Host 4 that is located at the end of the connection chains. Although the number of arrival packet on each host is vary on each host.

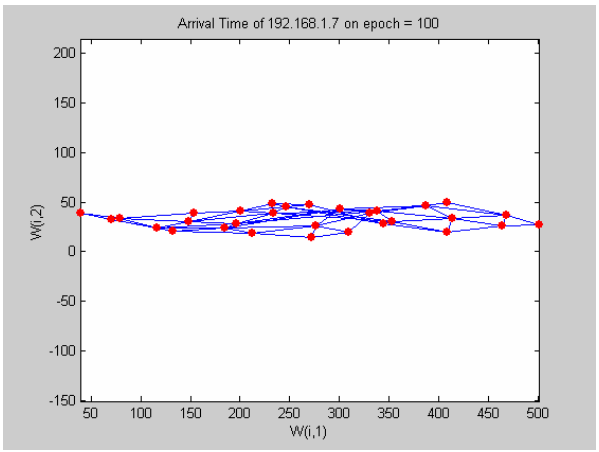

Fig. 11 Node of SOM Training on Host 4

SOM technique successfully determines the number of connection chains that exist in each host. This can be looked at the possible direction that can be form in each SOM training graphs. For example, this can be seen at Figure 8 and Figure 12 that show the existence of three possible directions of node. Two possible directions of node have being formed in Figure 9 and Figure 10 which means that it represents two connection chains that exist in both Host 2 and Host 3.

VIII. CONCLUSION AND FUTURE WORKS

From the study on other AI techniques that have been used on other research fields, this research successfully shows the potential of SOM technique as to count the number of connection chains that involve in network stepping stone detection.

From the executed experiment, it shows and proves that SOM technique can determine the number of connection chain that involved by just looking to the number of possible direction of node created by SOM training. Conventional or statistical technique that need network packet to be captured and compared all of the time in detecting a connection chain is not needed in this proposed technique. Just by capturing network data on certain time, SOM can detect or count the number of connection chains that involved in each host.

In this paper, the usage of SOM technique successfully shows the number of connection chains that involved in each involved host.

For the future work, the studies on SOM techniques will go deeper into precisely determine the number of connection chain that involved. This is because on this paper, the number of involved connection chains is determined by just looking the possible direction that form from the relation of the nodes. One precise formula as to determine the number of connection chains that involved need to be determined.

Besides, future works should involve the effect of active perturbation attack as this issue becomes a focus in current research on stepping stone detection. Active perturbation attack should be on dropped packet problem because the solutions for dropped packet problem still become unsolved problems.

The minimum number of packet that needs to be captured before SOM technique can give the correct result also needs to be studied further. This is to determine the minimum requirement for SOM techniques to produce towards a result with zero percent of false positive and false negative. As the pioneer research that used SOM technique as to detect the connection chain, more researches are expected to come in stepping stone detection research that use SOM techniques.

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 the authors to pursue PhD study in this field.

REFERENCES

[1] CERT, (2007, February 8). [Online]. Available:http://www.cert.org.
[2] Y. Zhang, and V. Paxson, "Detecting stepping stones", in *Proc. 9th USENIX Security Symposium*, Denver, 2000, pp. 67 – 81.
[3] S. Staniford-Chen, and L. T. Herberlein, "Holding intruders accountable on the Internet", in *Proc. 1995 IEEE Symposium on Security and Privacy*, Oakland, 1995, pp. 39 - 49.
[4] X. Wang, S. Chen, and S. Jajodia, "Network flow watermarking attack on low-latency anonymous communication systems", in *Proceeding of the 2007 IEEE Symposium on Security & Privacy (S & P 2007)*, USA, 2007, pp. 116 – 130.
[5] X. Wang, D. S. Reeves, and S. F. Wu, "Inter-packet delay based correlation for tracing encrypted connection through stepping stone", in *Proc. 7th European Symposium on Research in Computer Security (ESORICS 2002)*, Zurich, 2002, pp. 224 – 263.
[6] X. Wang, D. Reeves, and S. F. Wu, "Tracing based active intrusion response", *Journal of Information Warefare*, vol. 1, Issue 1, pp. 50-61, 2001.
[7] K. Yoda, and H. Etoh, "Finding connection chain for tracing intruders", in *Proc. 6th European Symposium on Research in Computer Security (LNCS 1985)*, France, 2000, pp. 31 – 42.
[8] L. Zhang, A. G. Persaud, A. Johnson, and Y. Guan, "Detection of stepping stone attack under delay and chaff perturbations", in *Proc. 25th IEEE International Performance Computing and Communications Conference (IPCCC 2006)*, USA, 2006, pp. 246 – 256.
[9] 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, 2004.
[10] W. Han-Ching, and S. H. Shou-Hsuan, "Performance of neural networks in stepping-stone intrusion detection", in *Proc. IEEE International Conference on Networking, Sensing and Control 2008 (ICNSC 2008)*, Sanya, 2008, pp. 608 – 613.
[11] Y. Jianhua, and S. H. Shou-Hsuan, "Mining TCP/IP packet to detect stepping-stone intrusion", *Computer & Security*, vol. 26(7-8), pp. 479-484, 2007.
[12] T. Kohonen, "The self-organizing map", In *Proceedings of the IEEE*. USA, 1990, pp. 1464-1480.
[13] M. N. Omar, M. A. Maarof, and A. Zainal, "The Optimization of Stepping Stone Detection: Packet Capture Steps", *Jurnal Teknologi*, 44(D), pp. 1 – 14, 2006.
[14] M. N. Omar, M. A. Maarof, and A. Zainal, "Identification steps for the optimization of stepping stone detection", in *Proc. ECTI Transaction on Electrical / Electronic and Communication (ECTI 2004)*, Thailand, 2004.
[15] M. N. Omar, M. A. Maarof, and A. Zainal, "Comparison Steps for The Optimization of Stepping Stone", in *Proc. Telematics System, Services, and Application 2004 (TSSA 2004)*, Indonesia, 2004.
[16] S. R. Snapp, J. Brentano, G. V. Dias, T. L. Goan, 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", in *Proceeding 14th National Computer Security Conference*, USA, 1991, pp. 161 – 176.
[17] 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", in *Proc. Proceedings of 4th USENIX Security Symposium*, New Orleans, 1997, pp. 69 – 78.
[18] S. Wadell, 1991. Private Communication.
[19] D. Schnackenbert, Dynamic Cooperating Boundary Controllers.
[20] X. Wang, "Tracing Intruders behind Stepping Stones", Ph.D. dissertation, North Carolina State University, 2004.
[21] Telnet Environment Option (2009, February 8). [Online]. Available: http://www.ietf.org/rfc/rfc1572.txt
[22] T. Ylonen, (2009, February 8). [Online]. Available: http://www.ietf.org/internet-drafts/drafft-ietf-secsh-architecture-16.txt.
[23] 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(1), pp. 109-115, 2008.
[24] X. Wang, and D. S. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays", in *Proc. 10th ACM Conference on Computer and Communication Security (CCS 2003)*, USA, 2003, pp. 20 – 29.
[25] M. Venkateshaiah, "Evading Existing Stepping Stone Detection Methods", *Master Thesis*, University of Texas at Arlington, 2006.
[26] M. N. Omar, L. Siregar, and R. Budiarto, "Hybrid stepping stone detection method", In *Proc. The 1st International Conference on Distributed Frameworks and Application (DFmA 08)*, Malaysia, 2008, pp. 134 – 138.
[27] M. N. Omar, and R. Budiarto, "Intelligent host-based stepping stone detection approach", *2009 World Congress on Computer Science and Information Engineering (CSIE 2009),* to be published.
[28] N. Michael, *Artificial Intelligence A Guide to Intelligent Systems*. Addison-Wesley. England, 2001.
[29] F. L. George, *Artificial Intelligence Structures and Strategies for Complex Problem Solving*, Addison-Wesley. England, 4th Edition, 2002.
[30] Wikipedia. (2009, February 8). Artificial Neural Network. [Online]. Available: http://en.wikipedia.org/wiki/Artificial_neural_network.
[31] S. Jian-Hua, J. Hai, C. Hao, and H. Zong-Fen, "MA-IDS: A Distributed Intrusion Detection System Based on Data Mining", *Wuhan University Journal of Natural Sciences (WUJNS)*, vol. 10(1), pp. 111-114, 2005.
[32] I. Yoo, and U. Ultes-Nitsche, "Intelligent firewall: packet-based recognition against internet-scale virus attacks", in *Proc. of Conference on Communications and Computer Networks (CCN 2002),* USA, 2002.
[33] P. Lichodzijewski, A. Z-H. Nur, and M. I. Heywood, "Host-based intrusion detection using self-organizing maps", in *Proc. of the 2002 International Joint Conference on Neural Network (IJCNN 02)*, USA, 2002, pp. 1714 – 1719.
[34] J. H. Albert, and S. S. Antti, "A computer host-based user anomaly detection system using the self-organizing map", in *Proc. of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00)*, Italy, 2000, pp. 411 – 416.
[35] H. Y. Kwong, "Detecting Long Connection Chains of Interactive Terminal Session" in *Proc. RAID 2002*. Switzerland, 2002, pp. 1 – 6.
[36] Y. Jianhua, and S. H. Shou-Hsuan, "A real-time algorithm to detect long connection chains of interactive terminal session", in *Proc. of the 3rd International Conference on Information Security*, China, 2004, pp. 198 – 203.
[37] Y. Jianhua, and S. H. Shou-Hsuan, and D. W. Ming, "A Clustering-Partitioning Algorithm to Find TCP Packet Round-Trip Time for Intrusion Detection" in *Proc. of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)*, Austria, 2006, pp. 231 – 236.
[38] Wireshark (2009, February 8). [Online]. Available: http://www.wireshark.org.
[39] Wareseeker (2008, February 8). [Online]. Available: http://wareseeker.com/freeware/telnet-scripting-tool-1.0/19344/TST10.zip.
[40] H. Duane, and L. Bruce, *Mastering MATLAB A Comprehensive Tutorial and Reference,* Prentice-Hall. New Jersey, 1996.

Mohd Nizam Omar became a Member (M) of IEEE in 2009. Received the B.S(Hons) and MSc in Computer Science from Malaysia University of Technology in 2002 and 2005, respectively. During 2000 – 2003, he stayed in Artifical Intelligent Lab and Group of Artifical Intrusion Network (GAINS) as Research Assistance. Currently, he is working at College Arts and Sciences, Information Technogy Building, Malaysia University of North, Malaysia. Currently, he is a Ph.D candidate at School of Computer Sciences, Malaysia University of Science, Penang, Malaysia. His research interest includes network security, tracing intruder, stepping stone detection, and Artificial Intelligent.

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 Universiti Sains Malaysia. His research interest includes IPv6, network security and Intelligent Systems. He was chairman of APAN Security Working Group. He is a member of IEEE Computer Society.