A Stepping Stone Perspective to Detection of Network Threats: Spam Detection

Mohd Nizam Omar, Ali Yusny Daud, Osman Ghazali

Abstract: ***This paper examines one of the novel applications of the concept of stepping stone detection to address network threats known as spam. Previous research has been identified in several applications such as spam, backdoors, intrusions of proxy servers and denial of service attacks as a possible solution that can be solved by using a stepping stone perspective against network threats. In this paper, an experiment has been conducted as to proof two formulas that generated to solve spam problem. Through the control environment and the development of special prototype to detect spam, the result shows that both formulas in detecting spam attack can be used to detect spam successfully. The successful result of the experiment proofs that one of the identified application really works in the real experiment testbed. By producing another solution to detect spam in this research hopefully can contribute another solution to detect a spam problem.***

I. INTRODUCTION

Computers share their resources as a means of handling data and performing tasks reasonably. Accurate, fast and reliable access to shared data is essential in network architectures. However, shared computer resources and operations proliferate the exposure of these networks to intrusions and threats to security, such as spam, backdoors, intrusions on proxy servers and attacks on Denial of Service (DoS), which may expose the confidentiality, accuracy and availability of shared data.

The stimulated intrusion technique used by attackers to maintain anonymity is through abuse between host computers or stepping stones to initiate attacks on other computers in the network.

To reveal the fraudster 's identity and to avoid further acceleration of uncertain activity, the discovery of the stepping stone is important.

Figure 1 clarifies the stepping stone crossing through the Internet from one host to the next host before the victim is attacked. From the victim's point of view, the attacker comes from the victim's previous stepping stone host. The last host (blameless host) will be recognized as an invader for this purpose.

Revised Manuscript Received on March 08, 2019.

Mohd Nizam Omar, InterNetworks Research Laboratory, School of Computing, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia

Ali Yusny Daud, InterNetworks Research Laboratory, School of Computing, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia

Osman Ghazali, InterNetworks Research Laboratory, School of Computing, Universiti Utara Malaysia, 06010 UUM Sintok, Kedah, Malaysia

The trace back to the source host (the real invader) will be virtually difficult with various internet connections. Research on stepping Stone Detection (SSD) is therefore authoritative. On the basis of traffic flow and appearances, countless algorithms have been established for the detection of stepping stones[1-15]. The SSD's achievement in identifying the right invader lights its novel application in detecting other network intimidations such as spam, backdoors, proxy server intrusions and attacks to Denial of Service (DoS). Current methods for analyzing and detecting spam, backdoors, intrusions of proxy servers and DoS attacks are developed using statistical methods, decision analysis, expert systems, neural networks and fuzzy logic[1-26].

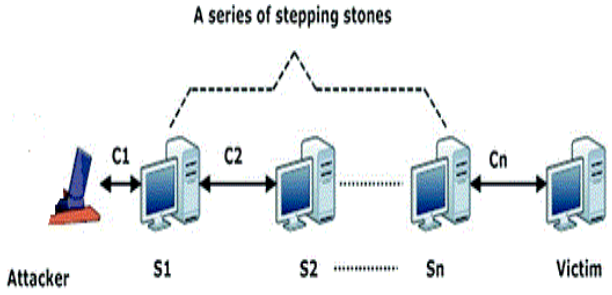

Fig. 1 Stepping Stone in General

To our knowledge, the perception of the SSD has not been extended to detect backdoors, intrusions of proxy servers and attacks on Denial of Service (DoS). As mention previously, SSD can be used to identify the others kind of network attack such as spam, backdoors, proxy server and DOS. After giving a glance information about all kinds of attacks that can be solved by using SSD, this paper will be discussed about one of the network attack known as spam further.

The paper continues as follows. To know the SSD further, essential terminologies for SSD are first distinct in Section 2.In Section 3, the general SSD notations will bediscussed. Section 4 then presents the proposed solution of the spam detection using the SSD approach in details. Section 5 ended this paper by discussing the conclusion of this paper.

II. TERMINOLOGY

Some important terminologies are presented here to facilitate the understanding of the basic SSD that this paper will discuss further. First, a host is a computer connected to a network of computers.

Assume that Host A is a target in Figure 2, Host B is the source and Host C is the endpoint. Since Host B is the source, the data flow from Host B to Host A is recognized as an incoming flow for Host A. However, the outgoing flow from Host A to Host C is measured.

The source denotes the originating host in stepping stone detection-based research and the destination denotes the source destination. An endpoint or victim is typically defined as the stepping stone's end destination. The incoming and outgoing flow is another term that is essential to consider. Alternatively, the incoming flow refers to the data entering a host and the outgoing flow indicates that the data leave a host. When the host is used to forward the data, i.e. by entering and then leaving the host, the stepping stone takes place.

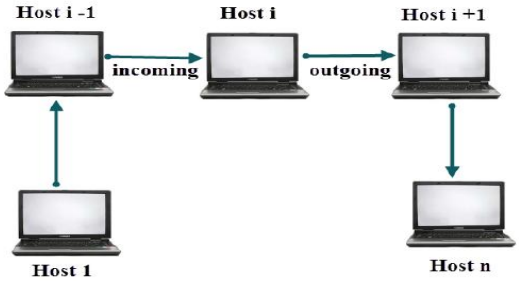

Fig. 2 General Stepping Stone Terminology

SSD can be defined as the stepping stoned host identification practices. This is known as the connection chain when one host transfers data to another host. SSD research's main objective is to compile the list of hosts. Host-based SSD (HSSD) is SSD concentrated on unraveling the stepping stone problem in a host, as compared to Network-based SSD (NSSD) which goals SSD problems in a network environment.

HSSD detects the stepping stone by using a host. It is done through the traffic of incoming and outgoing data. Host B obtains incoming data traffic from Host A in this case and the data traffic flows out to Host C as outgoing data. The relationship between incoming and outgoing data traffic on the same host occurs for HSSD. In other words, HSSD includes only a host for the detection of the stepping stone.

NSSD includes the detection of stepping stone from one or more hosts on the other hands. The detection also includes each host's incoming and outgoing data. Each connection between the host must be recognized in this case either as stepping stones or not as stepping stones. The NSSD is generated by the list. It is clear from the description that NSSD includes a different number of hosts.

III. GENERAL SSD

Figure 3 shows that there are three hosts labeled as A, B and C. Host B (Host i) exists before Host A (Host *i - 1*) and Host C (Host *i + 1*) exists after Host B. *i* represents the current stepping stoned host, *i - 1* represents the host before the *i* host and *i + 1* represents the host after the *i*-th host. Each host has its own incoming and outgoing flow. Host A has one incoming (A_{in}) and one outgoing flow (A_{out}). Host B and C also have their corresponding incoming and outgoing flow, symbolized B_{in}, C_{in} and B_{out}, C_{out} respectively.

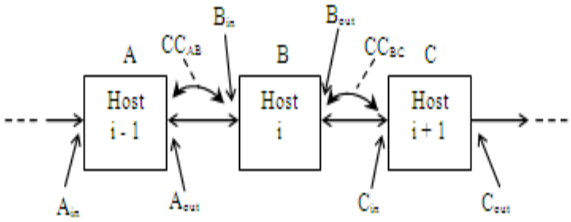

Fig. 3 Basic Notation of SSD

If the incoming flow is comparable to the outgoing flow, any host can be distinct as a stepping stoned host. If n_{in} and n_{out} are an incoming and outgoing flow to the host n,

$$SSD = \begin{cases} 1, & if\ n_{in} = n_{out} \\ 0, & if\ n_{in} \neq n_{out} \end{cases} \qquad (1)$$

From Eq. (1), if the incoming and outgoing flow is the same or $n_{in}=n_{out}$, the host is a stepping stone host or nss. The host is not considered a step stone otherwise. The connection chain between host A and host B, symbolized by CC_{AB} (can also be called CC_{BA} because $CC_{AB}= CC_{BA}$), occurs when host A and B are nss. A or B indicates the stepping stone's source or destination. In Eq. (1), two connection chains are available, CC_{AB} and CC_{BC}.

IV. THE NEW APPLICATION OF SSD

As discussed in previous sections, SSD-based research was usually limited to the detection of stepping stones without looking at the full capacity of SSD in other research fields. Potential SSD applications in other fields are listed here by studying other related research fields in security. These include spam detection, detection of proxy servers, backdoor detection and detection of DoS attacks.

SPAM Detection

Spam is the manipulation of electronic messaging systems by sending unsolicited messages indiscriminately in bulk [27]. Although spam can be used in many types of media, such as instant messaging, the USENET newsgroup and the web search engine [28], the potential use of SSD can be found in e-mail spam.

For e-mail-based spam such as[29],[30] and[31], a range of spam detection techniques have been explored. In the case of[29], the disclosure is performed manually by removing the spammed e-mail directly from the mailbox of the user e-mail, as the end user usually. On the other hand, spam detection is proposed in[30] using filtering. However, both techniques categorize a message by simply specifying the keyword, phrase and address. The results of spam detection will be in a high percentage of false positive signals using this method. The Artificial Intelligent (AI) techniques were recommended to overcome the problem [31]. However, the use of AI in spam detection, such as data mining, tends to take time. The spam e-mail appears to be a user's advertisement. Phishing and fraud are other purposes of spamming.

From the SSD point of view, a spam can be found from a host's incoming and outgoing e-mail port. Instead of finding

many port selections to be examined, the spam can be detected from the incoming port and the outgoing e-mail port. Instead of all ports used by other applications, the SSD approach can be more focused on the detection of anexact port. For example, port numbers 25, 143 and 110 are used for e-mail applications using the Simple Mail Transfer Protocol (SMTP), the Internet Message Access Protocol (IMAP) and Post Office Protocol Version 3 (POP3). Essentially, these are the ports that must be examined in the SSD approach.

The difference between the SSD concept practices in the detection of spam is that the number of incoming and outgoing traffic is not equal. In fact, the incoming spam is usually sent to one recipient and the same e-mail is sent to many other recipients. Then, e-mail spam detection can be written as

$$SPAM_{SSD} = \begin{cases} 1, if\ n_{in} < n_{out} \\ 0, if\ n_{in} \neq n_{out} \end{cases} \quad (2)$$

From Eq. (2), the incoming e-mail should be less than the outgoing e-mail in a host.

Proxy Server Detection

A proxy server is a server that assembles a client application, such as a web browser and a real server[32]. It is clear that the proxy server acts as an intermediary for requests to connect to the web server from hosts or clients. The proxy server will therefore temporarily store any data transmitted between hosts. It is vital to detect the proxy server because it prevents the user from remaining anonymous in the network.

A various methods for the detection of proxy servers have been explored. The network administrator's conventional approach is to use specific monitoring software such as Wireshark[34] for the detection of proxy servers. This approach, however, is not reliable. Another approach is to use a more reliable Intrusion Detection System (IDS) than the conventional approach, although it may take time. This latency may be caused by the use of data mining techniques in IDS[34-36].

This research recommended a simple SSD-based approach to improve latency in the detection of proxy servers. Figure 4 shows a model of basic SSD-based proxy server communication.

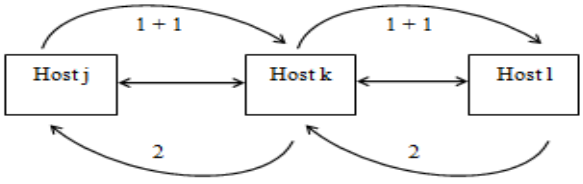

Fig. 4 Proxy Server Communication

From Figure 4, Host *j* as a proxy server sends a request to Host *l* through Host k. Therefore, by using the definitions given in Eq. (1) and Eq. (2), $CC_{j,k} = CC_{k,l}$ and $CC_{l,k} = CC_{k,j}$. Each host involved in proxy server detection through SSD.

$$Proxy_{SSD} = \begin{cases} 1, if\ n_{in} = n_{out} \\ 0, if\ n_{in} \neq n_{out} \end{cases} \quad (3)$$

From Eq. (3) to detect the proxy server; it is necessary to capture the incoming and outgoing traffic on the selected host.

Back Door Detection

The backdoor can be defined as an unseen approach to avoid regular computer authentication systems[37]. The backdoor program can be aggregate or exists for system processes. Backdoor programs are also implanted from numerous worms such as Sobig and Mydoom. Sophos Labs has exposed the W32 / Induc-A virus that has infected Delphi 's program compiler. The virus presents a new program with its own code that contaminates many systems that the programmer does not know about. This backdoor attack worked intangibly in the background until exposed a year later. Until exposed a year later, this backdoor attack worked intangibly in the background. This shows the danger of attacks on the backdoor.

Antivirus solutions can most often prevent backdoor intrusion [38]. However, this involves the correct signature implanted into the antivirus and can only be recognized in the host environment. For this reason, this research recommends a simpler solution to detect the backdoor using stepping stone detection concepts.

If the incoming and outgoing flow through the host is equal, a host can be defined as a stepping stoned host. In this situation, the detection occurs when the anexact port is connected many times. It usually occurs when the affected host directs the data abruptly to the external network using the same port number and within the same time period. Eq.(4) the detection of the back door was formulated using SSD.

$$Back\ Door_{SSD} = \begin{cases} 1, \ if\ n_{in} = n_{out} \\ 0, \ if\ n_{in} \neq n_{out} \end{cases} \quad (4)$$

From Eq. (4), host can be defined as the “back door” when incoming and outgoing flow are equal and not if not.

DoS Attack Detection

A Denial of Service(DoS) or Distributed Denial of Service(DDoS) attack is a type of attack that attempts to prevent network resources [39]. DoS or DDoS attacks usually occur on websites hosting high- profile web servers such as banks or payment gateways for credit cards.

There are many forms of attacks on DoS/DDoS. SYN flooding is a type of DoS attack where many SYN packets are directed and certainly not recognized, which suspends other users from the server and in severe cases endangers users when the server is completely shut down [40]. By using firewall settings [41], DoS attacks can be resolved. On the other hand, research by [42] addresses DDoS using a special Intrusion Detection System (IDS). In [43], the neural network is used to unravel DDoS. To address DoS attacks, a variety of AI can be used.

If numerous attackers are assumed to send many requests instantly to the target, the authentic user is therefore unable to send any request to the victim. This is the basic nature of the DoS/DDoS problem. These can be transmitted from the SSD perception as:

$$DoS_{SSD} = \begin{cases} 1, \ if\ n_{in} < n_{out}\ for\ \forall n \\ 0, \ if\ n_{in} = n_{out}\ for\ \forall n \end{cases} \quad (5)$$

Referring to Eq. (5), the stepping stone is identified if the number of incoming flows is less than the number of outgoing flows for the host by discovering the victim as the stepping stone host. Each host should balance the number of incoming and outgoing flows.

V. SPAM DETECTION USING SSD TECHNIQUE

Through study that has been conducted in SPAM, based on Eq. (6), SPAM can only be detected if $n_{in} \geq n_{out}$. Although in Eq. (3) earlier, SPAM can be detected if $n_{in} \neq n_{out}$, this is actually right because n_{in} is not equal with n_{out} or more accurately, n_{in} need to be greater or equal to n_{out}. This is shown in Eq. (6). Based on Eq. (2) also, to achieve accurate detection, this proposed research has also come out with Eq. (7) which is examine time of the packet out and in to the host.Based on the previous explanation about the spam, the problem formulation of Spam Stepping Stone Detection is based on this formula.

$$SPAM_{SSD} = \begin{Bmatrix} 1, if\ n_{in} < n_{out} \\ 0, if\ n_{in} \geq n_{out} \end{Bmatrix} \quad (6)$$

$$HOST_{spam} = \begin{Bmatrix} 1, if\ n_{out}time < n_{in}time \\ 0, if\ n_{out}time \geq n_{in}time \end{Bmatrix} \quad (7)$$

The $SPAM_{SSD}$ is the spam detection formula using the basic stepping stone. Eq. (6) is valid or true when inbound message to a host is less than outbound message from the host and it is false when inbound message is greater than or equal to outbound message. The inbound and outbound message is categorized based on the email header such as subject or body. When the similar email directed to a host and then the host redirected out the same email to all of his contacts, Eq. (6) will be triggered.

$HOST_{spam}$ on the other hands is the host that sending spam message through a network. Eq. (7) is true when the outgoing message time is less than the incoming message time and false when the time for outgoing message is greater than or equal to incoming message time. Eq. (7) is needed for detecting the real spammer in a network.

Experiment

To proof the Eq. (6) and Eq. (7), a set of experiment testbed has been set up. The problem explanation for Spam using SSD is based on testbed oriented where the sample is organized, numbers of host are fixed and the spam sample is directed manually.To send spam message, e-mail account needs to be set up. For this reason, a free e-mail account (gmail) has been set up.

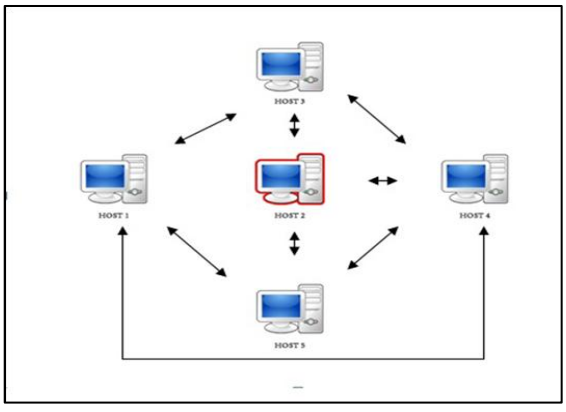

Fig. 5 Experimental Testbed

As shown in Figure 5, the inbound and outbound message flows are shown through the use of the arrow that pass through between the host and the details of email is logged to excel file. By referring to Table 1, each host does not have every host address.Figure 5 also shows that host two is highlighted in red. This is to show that the spammer is the host two which will be sending spam email throughout the network.

Table. 1 Host Connection Details

Host	Email	Stepping Stone Chain
Host 1	spamhost.one@gmail.com	H_3, H_4, H_5
Host 2	spamhost.two@gmail.com	H_3, H_4, H_5
Host 3	spamhost.three@gmail.com	H_1, H_2, H_4
Host 4	spamhost.four@gmail.com	H_1, H_2, H_3, H_5
Host 5	spamhost.five@gmail.com	H_1, H_2, H_4

In Table 1, host one has host three, host four, and host five addresses. Host one email address is spamhost.one@gmail.com.

For host two, three contacts are available in his list of contacts. The host two contacts are host three, host four and host five. The email address is spamhost.two@gmail.com.

Host three address is spamhost.three@gmail.com and has the host one, host two and host four address in his contact list. Meanwhile, host four has contact addresses of host one, host two, host three and host five with an email address which is spamhost.four@gmail.com. Lastly, the host five contacts are host one, host two and host four. The email address is spamhost.five@gmail.com. This kind of design is to mimic real life situations where a host might not have all contact address.

For the development of the application, the Visual Basic.Net based application is used. In the experiment, the application assembles information such as sender address, receiver address, subject, body, and timestamp. Then, all of this information transformed into an excel file as to create a log book of inbound and outbound message. This information is critical to detect a spammer using the SSD technique because it only way to regulate the spammer by inbound and outbound message as stated in Eq. (6).

Figure 6 displays the main interface of the SSD e-mail application that is being used for the experiment. It has two key functions which are to send email or view SSD Log.. Host needs to fill in all fields in order to send the e-mail. Then, the information is kept in an excel file.

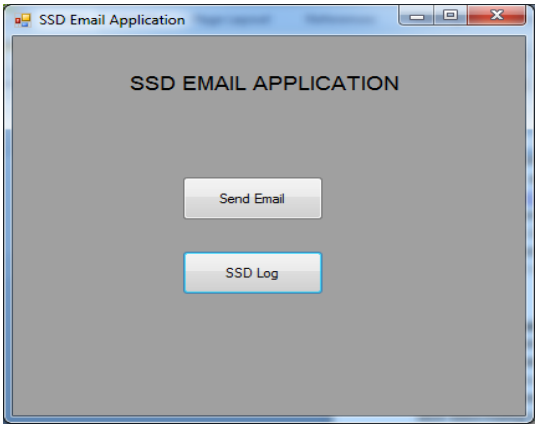

Fig. 6 SSD Email Application Interface

The complete information is keptin an excel file, including the body of an email. Because it is difficult to detect spammer with all the information, the application has also come out with the SSD Log that only to be used to display the information needed for the purpose of detecting spammer using SSD approach. In the SSD Log, the sender, receiver, subject and timestamp is presented so that the analysis can be done to detect the source of the spam. In Eq. (7), timestamp will be required as to detect the real perpetrator that distributes spam in the network.

The development of the application is significant as to show the competences of the proposed approach. Through the development of the application also, the detection of spam can be more effective compared to the usage of mathematical equation solely. The discussion of the result will be discussed later in the section (Section 5.2).

VI. RESULT

The result of the experiment in Table 2. Although the table is quite alike compared to Table 1, it is in the same time disclosures that the detection of spam using SSD successfully interpreted to the result. On the other hand, by considering to these two tablesarerevealed that the overall experiment shows the result clearly.

Table. 2 Experiment Result

Host	Detection
Host 1	H_3, H_4, H_5
Host 2	H_3, H_4, H_5
Host 3	H_1, H_2, H_4
Host 4	H_1, H_2, H_3, H_5
Host 5	H_1, H_2, H_4

From Table 2, it is exposed that each host effectivelyidentify all connected spam hosts. For example, Host 1 successfully identifies Host 3, Host 4 and Host 5 as the spammer. This is also applied to Host 2, Host 3, Host 4 and Host 5. Every host that used stepping stone detection approach identify all spammers without fail. On the other words, it is 100% of detection occurred here.

Through the experiment that has been performed, the connection between the hosts that involved in the experiment can be discovered. Not only each host has the ability of detecting spam (using a stepping stone detection technique) by itself, but the connection from differences host also possible to be recorded. The detection of a list of hosts that involved in spam solves many problems of spam detection.

VII. CONCLUSION

SSD has untapped potential in many emerging research fields for the discovery of host series by attackers, namely in spam, backdoor, proxy and DoS attack detection. In order to demonstrate the potential of SSD to address current spam, backdoor and proxy detection problems, four possible new SSD models are presented.

In this research, one of potentially several emerging research has been discussed further, it is known as spam. Extensive experiments has been executed as to prove the theory that has been discussed before.

The general result in this paper shown that the elementaryimpression on the detection of the SSD can be used for other purposed; in this case spam detection. This not only validate the idea through of our previous research paper, this paper also demonstrates the accomplishment of the research through the usage of real-time based templates.

VIII. ACKNOWLEDGEMENT

This work was made possible by the Fundamental Research Grant Scheme (FRGS) (S/O 12896) by the Ministry of Education of Malaysia.

REFERENCES

1. 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.
2. S. Robert, C. Jie, J. Ping and C. Weifeng, “A Survey of Research in Stepping Stone Detection”, International Journal of Electronic Commerce Studies”, Vol. 2, No. 2, pp. 103 – 126, 2001.
3. Y. Zhang and V. Paxson, “Detecting Stepping Stones”, Proc. 9th USENIX Security Symposium, 2000, pp. 67-81.
4. 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.
5. 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.
6. 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.
7. 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.
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. 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.
10. Y. Jianhua, 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.

11. S. Jianhua, J. Hai, C. Hao and H. Zong-Fen, MA-IDS: A Distributed Intrusion Detection System Based on Data Mining, Wuhan University Jornal of Natural Science (WUJNS), 10(1), pp. 111-114.
12. W. T. Strayer, C. E. Jones, I. Castineyra, J. B Levin and R. R Hain, “An Integrated architecture for attack attribution”, BBN Technologies, Technical Report. BBN REPORT-8384, 2003.
13. 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.
14. A. Almulhem and I. Traore, “A Survey of Connection-chains Detection Technique”, 2007IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, B. C, Canada, 22 – 24 August 2007, pp. 219 – 222.
15. X. Jianqiang, Z Lingeng, B. Aswegan, D. Daniels, J. T. Y. Guan,. (2006) A Testbed for Evaluation and Analysis of Stepping Stone Attack Attribution Techniques. Proc. 2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM 2006), 1-3 March 2006, Barcelona, Spain, pp. 369-379.
16. M. Venkateshaiah, “Evading Existing Stepping Stone Detection Methods”, Master Thesis, University of Texas at Arlington, December 2006.
17. A. Almulhem, Detection and Analysis of Connection Chains in Network Forensics, Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Victoria, Canada.
18. H. Wu, and S., S. Huang, Stepping Stone Intrusion Detection Using Neural Network Approach, Novel Algorithm and Techniques in Telecommunications, Automation and Industrial Electronics, pp. 358-363.
19. M. Venkateshaiah, and M. Wright, Evading Stepping Stone Detection Under the Cloak of Streaming Media, Technical Report, Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019, 2007.
20. J. Yang, and S. S. Huang and D. W. Ming. A Clustering-Partitioning Algorithm to Find TCP Packet Round-Trip Time for Intrusion Detection, Proceeding of 20th International Conference on Advanced Information Networking and Applications (AINA 2009), Bradford, UK, pp. 231-236.
21. J. Yang, and S. S. Huang, S. S. Mining TCP/IP packet to detect stepping-stone intrusion. Computer & Security, 26(7-8), pp.479-484.
22. H. Wu, and S. S. Huang. Neural Network-based Detection of Stepping Stone Intrusion. Expert Systems with Applications, 32(2), pp.1431-1437.
23. A. Almulhem and I. Traore. Detecting Connection-Chains: A Data Mining Approach, International Journal of Network Security, 10(1), pp.62–74.
24. J. Yang and S. S. Huang. A Real-Time Algorithm to Detect Long Connection Chains of Interactive Terminal Session. Proceeding of The 3rd International Conference on Information Security (InfoSecu04). 14-16 November 2004, Shanghai, China, pp. 198-203.
25. J. Yang and S. S. Huang. Matching TCP Packets and Its Application to the Detection of Long Connection Chains on the Internet. The 19th International Conference on Advanced Information Networking and Application (AINA 05), 28-30 March 2005, Taipei, Taiwan, pp.1005-1010.
26. X. Wang and D. S. Reeves. Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays, The 10th ACM Conference on Computer and Communication Security (CCS 2003), 27-30 October 2003, Washington D.C., USA, pp. 20-29.
27. B. Whitworth and E. Whitworth, "Spam and the social-technical gap," Computer, vol. 37, pp. 38-45, 2004.
28. M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail," in Learning for Text Categorization: Papers from the 1998 Workshop, 1998.
29. D. D'Ambra, "Killer spam: clawing at your door", Inf. Prof. 4, vol. 28, no. 4, 2007.
30. Z. Le, Z. Jing and Y. Tianshun, "An Evaluation of Statistical Spam Filtering Techinques", ACM Transactions on Asian Language Information Processing (TALIP) vol. 3, 2004, pp. 243-269.
31. M.N. Marsono, M. Watheq, and F. Gebali, "Binary LNS-based naïve Bayes inference engine for spam control: noise analysis and FPGA implementation", IET Comput. Digit. Tech, vol. 56, no. 2, 2008.
32. O. O. Abiona, T. Anjali, L. O. Kehinde, "Simulation of a cyclic multicast proxy server," IEEE International Conference on Electro/Information Technology, 2008. EIT 2008., vol., no., pp.102-107, 18-20 May 2008
33. O. Angela, R. Gibert, B. Jay and W. Joshua. Wireshark & Ethereal Network Protocol Analyzer Toolkit (Jay Beale's Open Source Security), Syngress Publishing, Inc., 800 Hingham Street, Rockland, MA 02370.
34. R. Chetan and D. V. Ashoka, "Data mining based network intrusion detection system: A database centric approach," Computer Communication and Informatics (ICCCI), 2012 International Conference on , vol., no., pp.1-6, 10-12 Jan. 2012
35. F. Desheng, Z. Shu and G. Ping, "The Design and Implementation of a Distributed Network Intrusion Detection System Based on Data Mining," Software Engineering, 2009. WCSE '09. WRI, World Congress on,Vol. 3, no., pp. 446-450, 19-21 May 2009
36. L. Lei, Y. De-Zhang and S. Fang-Cheng, "A novel rule-based Intrusion Detection System using data mining," Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on,Vol. 6, no., pp. 169-172, 9-11 July 2010
37. H. Agrawal, J. Alberi, L. Bahler, W. Conner, J. Micallef, A. Virodov, S. R. Snyder, "Preventing insider malware threats using program analysis techniques," MILITARY COMMUNICATIONS CONFERENCE, 2010 - MILCOM 2010, vol., no., pp. 936-941, Oct. 31 2010-Nov. 3 2010
38. S. Rahul, “Effectiveness of Antivirus in Detecting Web Application Backdoors”, retrieved from http://www.chmag.in/article/feb2011/effectiveness-antivirus-detecting-web-application-backdoors, July 30, 2012.
39. Fang-Yie Leu; Zhi-Yang Li; "Detecting DoS and DDoS Attacks by Using an Intrusion Detection and Remote Prevention System," Information Assurance and Security, 2009. IAS '09. Fifth International Conference on,Vol. 2, no., pp. 251-254, 18-20 Aug. 2009
40. Mehdi Ebady Manna, Angela Amphawan; “Review of SYN-flooding attack detection mechanism”, International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.1, pp. 99-117, January 2012
41. Salah, K.; Sattar, K.; Sqalli, M.; Al-Shaer, E.; , "A probing technique for discovering last-matching rules of a network firewall," Innovations in Information Technology, 2008. IIT 2008. International Conference on, vol., no., pp. 578-582, 16-18 Dec. 2008
42. Bose, S.; Kannan, A. , "Detecting Denial of Service Attacks using Cross Layer based Intrusion Detection System in Wireless Ad Hoc Networks," International Conference on Signal Processing, Communications and Networking, 2008. ICSCN '08., vol., no., pp.182-188, 4-6 Jan. 2008.
43. Jin Li; Yong Liu; Lin Gu; , "DDoS attack detection based on neural network," Aware Computing (ISAC), 2010 2nd International Symposium on , vol., no., pp.196-199, 1-4 Nov. 2010.