Evolution of Stepping Stone Detection and Emerging Applications

MOHD NIZAM OMAR[1], ANGELA AMPHAWAN[2], ROSHIDI DIN[3]
InterNetWorks Research Laboratory[1,2], School of Computing[3], UUM College of Arts and Sciences
Universiti Utara Malaysia
UUM Sintok, 06010, Kedah
MALAYSIA
niezam@uum.edu.my[1], angela@uum.edu.my[2], roshidi@uum.edu.my[3]
http://www.internetworks.my[1,2] http://www.soc.uum.edu.my[3]

Abstract: - Stepping Stone Detection (SSD) is conventionally intended for the detection of series of host computers used by attackers to hide their track in a network or host environment. This paper discusses the evolution of SSD and potential applications in other emerging fields. Novel, unique SSD models will be presented for spam, backdoor and proxy detections and expressed mathematically. These preliminary models have promising solutions for addressing current problems in these areas and may be expanded on in the future.

Key-Words: - Stepping stone, stepping stone detection, spam, backdoor, proxy server detection

1 Introduction

Stepping Stone Detection (SSD)-based research was instigated by [1] in 1995. To date, there are more than 50 varieties of SSD [2] focusing on the detection of list of host computers used by the attacker to hide their track. For example, [3] proposed a time-based method in interactive sessions, [4] relates to the delay and chaff problem and the latest research by [5] applied Artificial Intelligent (AI) techniques in detecting the stepping stone.

Although SSD is conventionally used for retrieving the list of host computers in detecting the attacker, recent explorations indicate that it is possible to extend the theory of SSD to other emerging fields. In this paper, we present the evolution of conventional SSD and suggest three important emerging research fields where SSD would be valuable, namely SPAM, backdoor and proxy detection. A novel, unique SSD model will be presented for each emerging field and expressed mathematically. These preliminary models have promising solutions for addressing current problems in these areas and may be expanded on in the future.

The paper proceeds as follows. To understand the SSD models presented later, important terminologies for SSD are first defined in Section II. In Section III, as a basis for the modeling, the general concept of SSD and the current, previous and future landscape for SSD are illustrated. Section IV then presents our three novel SSD models for SPAM, backdoor and proxy detection.

2 Terminology

Important terminologies are presented here to facilitate the understanding of the models presented in later sections. A host is any computer that connected to a computer network. In stepping stone detection-based research, the source refers to the origin host. The destination refers to the destination of the source. A target or victim is usually defined as the last destination of the stepping stone. Another term that needs to be considered is the incoming and outgoing flow. Incoming flow refers to the data that enters a host and outgoing flow indicates the data leave a host. Stepping stone occurs when the host is used for forwarding the data, i.e.by entering and then exiting the host. Stepping stone detection can be defined as the processes of detecting the stepping stoned host. When one host forwards data to another host, this is known as the connection chain. The main goal of stepping stone detection research is to collects the list of hosts. Host-based SSD (HSSD) is SSD focused on solving the stepping stone problem in a host as compared to Network-based SSD (NSSD) which targets SSD problems in a network environment.

3 Past, Current and Future SSD

As an overview to SSD and to place our novel SSD models for emerging fields in context, the general concept of SSD and the research landscape of SSD are presented here.

3.1 General Concept of SSD

The general concept of SSD is depicted in Fig. 1.

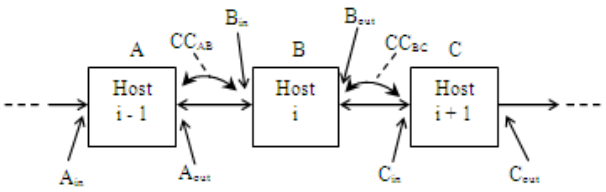

Figure 1: Basic of Stepping Stone Detection

From Fig. 1, 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 outcoming flow (A_{out}). Host B and C also have their corresponding incoming and outgoing flow, denoted B_{in}, C_{in} and B_{out}, C_{out} respectively.

Any host may be defined as a stepping stoned host when the incoming flow is similar to outgoing flow.If n_{in} and n_{out} represent incoming and outgoing flow on host n,

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

From (1), if the incoming and outgoing flow is equal, $n_{in} = n_{out}$ this means the host is a stepping stoned host or n_{ss}. Otherwise, the host is not a stepping stone.

The connection chain from Host A to Host B, denoted by CC_{AB} (may also can be denoted as CC_{BA} because $CC_{AB} = CC_{BA}$) occurs when Host A and B are n_{ss}. A or B represents the source or destination of the stepping stone. In Fig. 1, there are two connection chains, CC_{AB} and CC_{BC}.

In SSD, the series of connection chain that exists along the network that we monitor may be expressed as:

$$SSD = \{CC_{s_n d_m},\ CC_{s_{n+1} d_{m+1}}, CC_{s_{n+2} d_{m+2}} \ldots, CC_{s_{n-k} d_{m-k}},\ CC_{s_{n+k} d_{m+k}}\} \quad (2)$$

wheres is the source, d is the destination. From (2), it is clear that SSD has a collection of CC

from $CC_{s_n\, d_m}$ to $CC_{s_{n+k}\, d_{m+k}}$.

However, it is only to be true when the number of CC is more than one, |SSD|> 1.

3.2 Past SSD

As the pioneer in SSD research, [1] proposed the concept of 'thumbprint' that summarized a packet's content by providing it with a unique identity which differentiated it from other packets. However, the thumbprint solution was not suitable for encrypted connections. Consequently, [6] and [3] proposed on/off and deviation methods respectively. Unfortunately, these two methods were prone to high false positive and active perturbation problems. [7] proposed the "reply-echo" method to reduce the false positive problem and [8] proposed overcoming the perturbation problem using APA. APA is a technique created by the intruder to influence the SSD process. At the same time, [9] applied the Inter-packet delay (IPD) method to solve the stepping stone problem by proposing a new use of data that is more effective in detecting stepping stones.

After [7] first introduced a new technique, Round Trip Time (RTT), which is for reducing the false positive rate, past SSD researches began conducting experiments related to [7]'s research. Research by [10] introduced the "Step-Function" and "Conservative & Heuristic" [11] which were methods enhanced from [7] methods. Meanwhile, research by [12] was the only research that focused on the wireless environment in detecting stepping stones. In conclusion, it seems that past SSD research focused on the right data type to be used in the SSD approach. The differences lie only in different types of data (e.g. data, time, inter-packet delay) and their concentration on RTT at the end of the past SSD period.

3.3 Current SSD

[13], [17], [15], [16] and [18] have shown that SSD researchers have changed their focus from enhancing the SSD approach to something that can make SSD more robust against perturbation. This can be seen in research by [13] that re-directed SSD research towards achieving less false positives and false negative rates. [19] created a method to influence SSD and [15] provided a testbed through which the SSD approach can be examined. [14], on the other hand, provided SSD taxonomy to expose those outside the field to SSD.

Research on the present SSD have become more widespread with the introduction of Artificial Intelligence (AI) techniques. Research which

applies AI techniques are referred to as RTT-based research [20], [18], [21], [22]. This effort was started by [21] who proposed the data mining technique to mine for TCP/IP packets in the effort of finding RTT. The application of AI was continued by [18] who introduced the Neural Network technique that focuses on finding RTT. From the discussion on AI techniques that have been used, it seems that their technique had the potential of solving SSD problems.

The present SSD research not only focuses on issues beyond those of the past SSD research, but also introduces new discoveries to the SSD research world. The introduction of different AI techniques used to detect RTT and later to detect stepping stones, shows that the present SSD is evolving. The present SSD also shows that the extensive buffering method used as perturb to the present SSD approach exists [19]. There are also studies which focus on confidence bound [13], false positive and false negative rates [18]. Attached testbed which is much needed in SSD research has also been proposed by [15].

3.4 Future SSD

Future SSD would focus on the development of SSD testbeds. The standard testbed is necessary to the SSD-based research to execute the standard experiment or testing. In the testbed, the requirements, the tools and the topology that will be used are well defined. So far, SSD research has only depended on the testbed developed by [15]. Unfortunately, this testbed has still to be made known to the public. Moreover, from the readings it was found that a standard SSD testbed does not exist to date and most researchers use their own testbeds. Because of the use of AI techniques in the SSD environment, future SSD should focus on the development of testbeds that support AI SSD.

Another possibility that could become the research focus for future SSD is the concept of hybrid SSD. More often than not, the past and present SSD research has only depended on network-based SSD (NSSD) [2], [1], [6]. Although these studies did not explicitly define their SSD approach as NSSD, the use of network packets as the main source of stepping stone detection process shows that it is NSSD. Studies by [14] and [26] have divided the SSD approach into network-based and host-based SSD (HSSD).

From the discussion on past, current and future SSD, it is concluded that all of the researcher focus to the main usage of SSD; to detect stepping stone either in host or network-based environment. No such a research that realized the other usage of the SSD in other fields of research. This is what we will provide in this paper, to provide a new direction of the stepping stone detection based research.

4 Emerging Fields for Application of SSD

As discussed in previous sections, stepping stone detection-based research was mostly limited to the detection of stepping stones without looking to the full capabilities of stepping stone detection in other fields of research. Suggestions for potential applications of SSD in other fields are listed here.

4.1 Spam Detection

Spam is the abuse of electronic messaging systems by indiscriminate sending of unsolicited messages in bulk [27]. Although several types of media such as instant messaging, USENET newsgroup and web search engine fall prey to spam [28], the potential use for SSD may find its way in e-mail-based spam.

A variety of spam detection techniques have been investigated for e-mail-based spam such as [29], [30] and [31]. In the case of [29], the detection is carried out manually and [30] filtering is proposed for spam detection. However, both techniques classify a message by simply identifying keyword, phrase and sending address. This results in a high percentage of false positive signals. To overcome the problem [31] recommended suggests the Artificial Intelligent (AI) techniques. However, frequently, the application of AI in spam detection, such as data mining, tends to be time consuming.

From the SSD perspective, a spam can be detected from the incoming and outgoing e-mail port from a host. Instead of detection on many choices of port that need to be monitored, detection of the spam can be made from the incoming port and the outgoing port of the e-mail. This allows the SSD approach to be more focused on the detection of a specific port, rather than all ports used by other applications. For instance, port number 25, 143 and 110 are used for Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP) and Post Office Protocol Version 3 (POP3) applications of the e-mail, respectively. These are actually the ports that need to be monitored in SSD approach. In fact, the total number of ports used by the application range up to 65535 ports.

The different point between the usages of SSD concept in the spam detection is the number of incoming and outgoing traffic definitely not in an equal numbers. In fact, the incoming spam usually addressed to one receiver, and then the same e-mail

will be used to be sent to many other receivers. Therefore, e-mail spam detection can be written as

$$\text{SPAM}_{\text{SSD}} = \begin{cases} 1, & if\ n_{in} < n_{out} \\ 0, & if\ n_{in} \neq n_{out} \end{cases} \quad (3)$$

From (3), it is shows that the incoming e-mail in a host should be less than the outgoing of the e-mail. If there are n hosts involved in the spam stepping stone detection,

$$\text{SPAM}_{\text{SSD}} \text{ for } n_1 = \text{SPAM}_{\text{SSD}} \text{ for } n_2 = \text{SPAM}_{\text{SSD}} \text{ for } n_3 \ldots \text{SPAM}_{\text{SSD}} \text{ for } n_{k-1} = \text{SPAM}_{\text{SSD}} \text{ for } n_k \quad (4)$$

where *k* is the number of host.

From (4), we can collect all of the host that involved as the spammed host as

$$\text{SPAM}_{\text{SSD's}}\ CC = \{n_1, n_2, n_3, \ldots, n_{k-1}, n_k\} \quad (5)$$

In (5), the spam SSD actually collects the connection chain between one host to another host. If (5) has been applied to different mail servers, the origin of the spam may possibly be identified easily. In other word, the list of hosts that are involved in the spam is actually the connection chain that exists between one host to another host.

4.2 Proxy Server Detection

A proxy server is a server that sits between a client application, such as a web browser and a real server [32]. A diverse range of approaches for proxy server detection have been investigated. The conventional approach is for the network administrator to use specialized monitoring software such as Wireshark [34] for proxy server detection. However, this approach is not infallible. Another approach is to use Intrusion Detection System (IDS) which is more fail-safe than the conventional approach, although it can be time-consuming. The use of data mining technique in IDS possibly is the cause of this latency [34][35][36].

To alleviate latency in proxy server detection, we propose a simple SSD-based approach. A preliminary model of basic proxy server communication based on SSD is shown in Fig. 2.

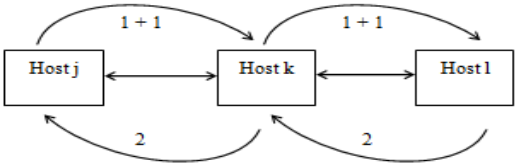

Figure 2: Basic Proxy Server Communication

Form Fig. 2, Host j sends a request to Host l through Host *k* as the proxy server. Therefore, by using the definitions given in (1) and (2), $CC_{j,k} = CC_{k,l}$ and $CC_{l,k} = CC_{k,j}$.

For the proxy server detection through SSD, each host involved:

$$\text{Proxy}_{\text{SSD}} = \begin{cases} 1, & if\ n_{in} = n_{out} \\ 0, & if\ n_{in} \neq n_{out} \end{cases} \quad (6)$$

For network-based proxy server detection or the list of connection chains involved in the proxy server, it based on:

$$\text{Host\_Proxy}_{\text{SSD}} \text{ for } \text{Host}_{s,d} = \text{Host\_Proxy}_{\text{SSD}} \text{ for } \text{Host}_{s+1,d+1} = \text{Host\_Proxy}_{\text{SSD}} \text{ for } \text{Host}_{s+1,\,d+1} = \ldots = \text{Host\_Proxy}_{\text{SSD}} \text{ for } = \text{Host\_ProxySSD for } \text{Host}_{s+1-k,\ d+1-k} = \text{Host\_Proxy}_{\text{SSD}} \text{ for } \text{Host}_{s+k,d+k}$$

where *k* is the last number of host. By assuming

$\text{Host}_{s,d}$ also including $\text{Host}_{d,s}$ for each host, we can write a full network-based in the form of

$$\text{Network\_Proxy}_{\text{SSD}} = \{\text{Host}_{s,d}, \text{Host}_{s+1,\,d+1}, \ldots, \text{Host}_{s+1-k,\ d+1-k}, \text{Host}_{s+1-k,\,d+1-k}\} \quad (7)$$

From (7), it is clear that to detect the proxy server; we simply need to find the incoming and outgoing traffic on the chosen host.

4.3 Backdoor Detection

Backdoor can be defined as a hidden approach for bypassing normal computer authentication systems [37]. Most of the time, antivirus utilities play an important role in overcoming the backdoor problem [38]. However, this requires the right signature embedded into the antivirus and the detection can only be executed in host-based environment. For this reason, we propose a simpler solution for detecting backdoor by using concepts from stepping stone detection based-research.

Backdoor detection using the SSD concept is directed to the host-based level. However, it is can be extended to the network-based level or detection on the chain of the backdoor so as to find the origin of the backdoor as discussed in (2).

Referring to (1), a host can be defined as a stepping stone host when the incoming and outgoing

flow through the host is the same. In the backdoor situation, the detection occurs when a connection occurs for a many times for a specific port. It usually happens when the affected host suddenly sends a data to the outside network using the same port number and at the same period of time. If the backdoor affects a number of hosts (used as stepping stone), we can use (2) to overcome the problem. Open research questions include the number of occurrences that need to be counted and the port affected by backdoors.

5 Conclusion

Conventionally for the detection of series of host computers by attackers, SSD also has untapped potential in several emerging research fields, namely in SPAM, backdoor and proxy detection. We present three novel SSD mathematical models to demonstrate the potential of SSD in addressing current issues in SPAM, backdoor and proxy detection. For future work, extensive simulations on real data for each emerging area will be undertaken.

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 ans 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, Washinton 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," Electro/Information Technology, 2008. EIT 2008. IEEE International Conference on , 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.