A Stepping Stone Perspective to Detection of Network Threats

Mohd Nizam Omar, Angela Amphawan, and Roshidi Din

***Abstract*—** Current computing trends such as cloud computing, file sharing and social networking promote collaboration and allow greater mobility for users. Nevertheless, these computing trends increase the vulnerability of networks to security threats and challenge network resources. An ingenious technique employed by attackers for retaining anonymity is by exploiting intermediary host computers or stepping stones to instigate attacks on other computers. This paper explores novel application of the stepping stone detection concept in addressing network threats such as spams, backdoors, proxy server intrusions and denial of service attacks. Preliminary stepping stone detection models for each security threat will be constructed and the potential detection process is delineated. These preliminary concepts and models may prove useful for further optimization of network security in conjunction with other conventional detection techniques.

***Keywords*—** Stepping stone, stepping stone detection, spam, backdoor, proxy server detection, DoS, DDoS.

I. Introduction

Current computing trends such as file sharing, video streaming, cloud computing, interactive games and social networking promotes collaboration and allow greater mobility for users. Computers share their resources as a means to inexpensively handle data and perform tasks. In these network architectures, accurate, rapid and reliable access to shared data is essential. Nevertheless, the shared computational resources and operations increase the vulnerability of these networks to intrusions and security threats such as spams, backdoors, proxy server intrusions and denial of service (DoS) attacks, which may jeopardize the confidentiality, accuracy and accessibility of shared data.

For retaining anonymity, an ingenious intrusion technique employed by attackers is by exploiting intermediary host computers or stepping stones to instigate attacks on other computers within the network.

Mohd Nizam Omar is with InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia, Sintok 06010 MALAYSIA (corresponding author provide phone: 6017-538-7991; fax: 604-928-4753 ; e-mail: niezam@uum.edu.my).

Angela Amphawan is with InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia, Sintok 06010 MALAYSIA (e-mail: angela@uum.edu.my).

Roshidi Din is with School of Computing, Universiti Utara Malaysia, Sintok 06010 MALAYSIA (e-mail: roshidi@uum.edu.my).

The discovery of the stepping stone is important for revealing the identity of the fraudster and for preventing further escalation of dubious activity.

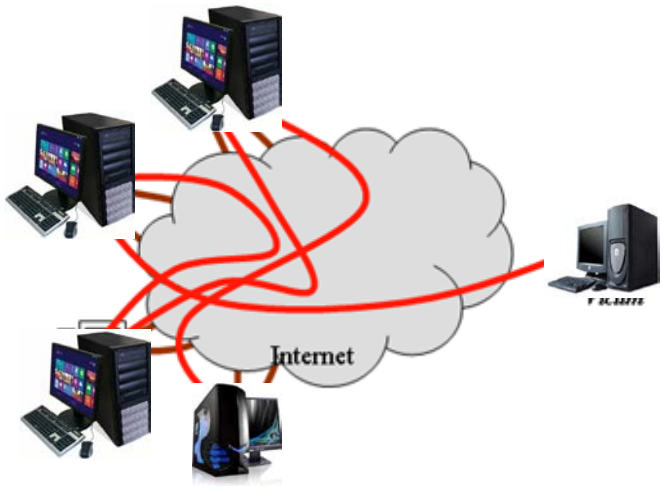

Figure 1: Stepping stone.

Figure 1 illustrates the traversal of the stepping stone from one host to next through the Internet before the real the victim is attacked. From the victim's perspective, the attacker originates from the last stepping stone host connected to the victim. For this reason, the last host (innocent host) will be identified as the attacker. With multifarious inter-connections on the Internet, the trace-back to the origin host (the real attacker) will be almost impossible. Therefore, research on stepping stone detection (SSD) is imperative.

Various algorithms for stepping stone detection have been developed based on traffic flow and characteristics [1-15]. The success SSD in detecting the true attacker brings to light its novel application in the detection of other network threats such as spams, backdoors, proxy server intrusions and denial of service (DoS) attacks. Existing methods for analyzing and detecting spams, backdoors, proxy server intrusions and DoS attacks are based on statistical methods, decision analysis, expert systems, neural networks, and fuzzy logic[1-26].

To our knowledge, the SSD concept has not, to date, been extended to the detection of backdoors, proxy server intrusions and denial of service (DoS) attacks.

This paper explores the novel application of stepping stone detection in addressing a diverse range of network threats such as spams, backdoors, proxy server intrusions and denial of service (DoS) attacks. Preliminary stepping stone detection models for each security threat will be constructed mathematically and the potential detection process for each is delineated. These preliminary concepts and models may prove useful for further optimization of network security in conjunction with other conventional detection techniques.

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 four novel preliminary SSD models for SPAM, backdoor, proxy and DoS attack detection.

II. Terminology

Some important terminologies are presented here as to facilitate the understanding of the models presented in next sections. First, a host is any computer that connected to a computer network.

Figure 2 shows the related terminology discussed previously.

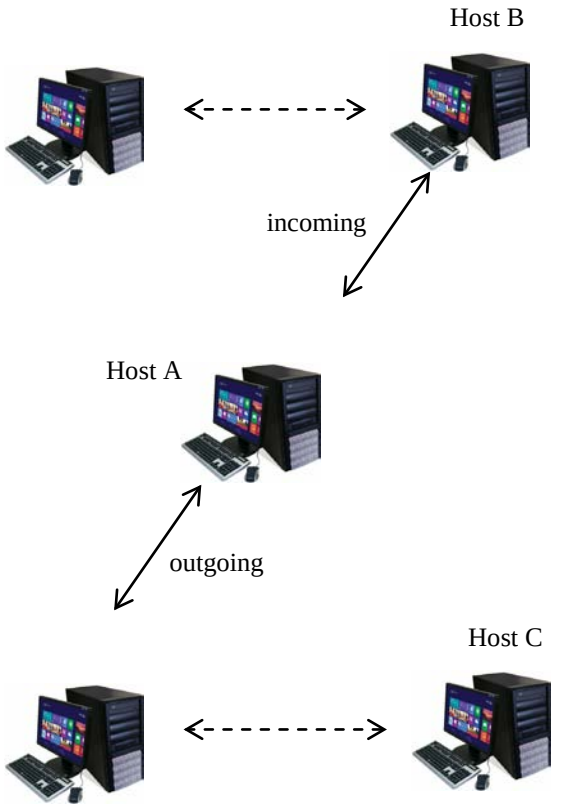

Figure 2: General Stepping Stone

In Figure 2, assume Host A is a victim, Host B is the source and Host C is the destination. Since Host B is the source, the data flow from Host B to Host A is identified as an incoming flow for Host A. However, the data flow between Host A and Host C is considered an outgoing flow from Host A to Host C.

In stepping stone detection-based research, the source refers to the origin host and the destination refers to the destination of the source. A target or victim is usually defined as the last destination of the stepping stone. Then, another term that needs to be considered is the incoming and outgoing flow. Incoming flow on the other hand 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. Figure 3 shows the HSSD and Figure 4 shows the NSSD.

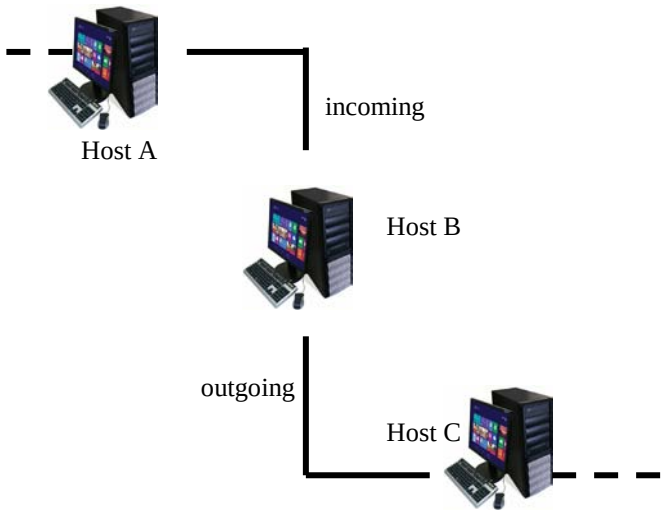

Figure 3: HSSD

From Figure 3, it is clear that HSSD employs a host to detect the stepping stone. It is done by compare incoming and outgoing data traffic. In this case Host B receive incoming data traffic from Host A and the data traffic data flows out as outgoing data to Host C. For HSSD, the comparison occurs between incoming and outgoing of the data traffic on the same host. On the other words, HSSD only involves a host as to detect the stepping stone.

From Figure 4, NSSD involves the detection of stepping stone from one or more hosts. The detection also involves the incoming and outgoing data for each host. In this case, each connection between the host need to be identified either it is stepping stoned or not stepping stoned. The list of these list create the NSSD. From the explanation, it is clear that NSSD involves different number of host to be examined.

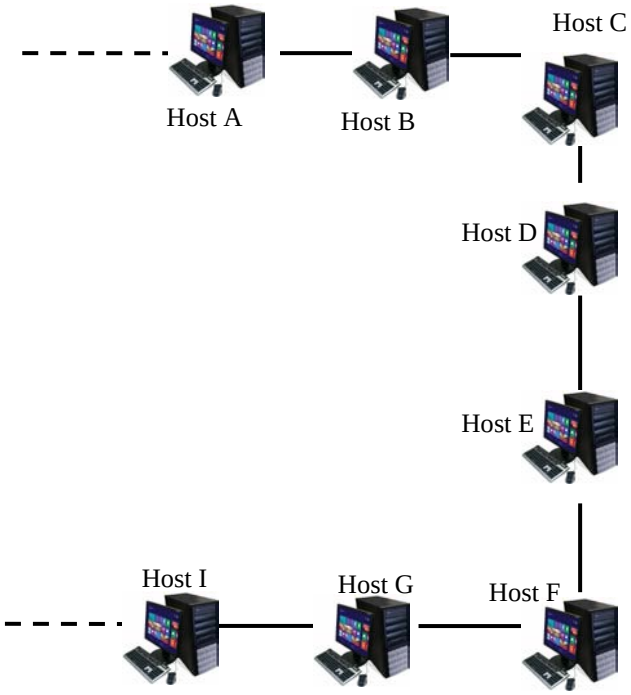

Figure 4: NSSD

III. SSD in General

The general concept of SSD is depicted in Fig. 5.

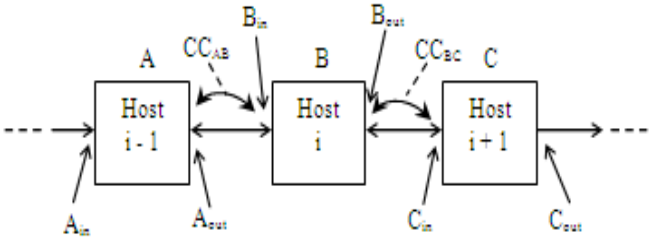

Figure 5: Basic of Stepping Stone Detection

From Fig. 5, 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 stone 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:

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

wheres is the source, d is the destination. From (2), it is clear that SSD has a collection of CC from CC_{s_n} to $CC_{s_{n+k}\, d_m}$. However, it is only to be true when the number of CC is more than one, |SSD|> 1.

IV. History of 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.

Generally, we can divide the overall history of SSD into three different parts; past SSD, current SSD and future SSD. Past SSD refers to SSD approaches used before what we have currently. Current SSD on the other hand includes SSD approaches that are used presently while future SSD brings novel ideas related to the applications of SSD in the future. Figure 6 summarizes the past, current and future SSD.

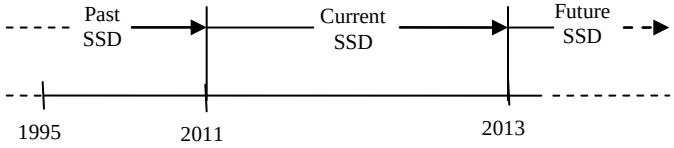

Figure 6: Past, Current and Future SSD

From Figure 6, it is shown that SSD-based research started from 1995 and has evolved to what we have today in 2013. Current SSD-based research can be considered SSD approaches in between 2011 and 2013. Details about each part of the SSD will be discussed in subsequent subsections.

A. 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 Active Penetration Attack (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. The "Step-Function" method decreases the false positive and false negative errors and successfully worked in the Local Area Network (LAN) environment. The "Conservative & Heuristic" method, on the other hand, was enhanced to be used in the Internet environment.

Meanwhile, research by [12] was the only research that focused on the wireless environment in detecting stepping stones. In their research, state-space algorithm was proposed. However, after that, there has been no such research on SSD conducted in the wireless environment.

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.

B. Current SSD

Present SSD research begins with [13], which concentrates on the confidence bound problem, false positive rate with and without chaff perturbation based on computational and random walks theories. [14] later divided SSD methods into three categories: host, network and system.

However, in this research study, SSD methods are divided into only two categories; host and network SSDs. This is supported by [9] who also divided SSD methods into two categories. For this research, a system (from [14] can be either host or network-based SSD) and should not become a separate category of the SSD methods. Studied in [15] has proposed a special testbed for SSD attack testing.

According to their research, the testbed can be used dynamically on different types of SSD approaches by using different types of attack. After that, research on SSD continued with the introduction of the buffering method used to perturb the SSD approach. This research was conducted by [16]. Through a simulation of the watermark-based SSD, it was shown that by only buffering the packet and arranging it according to user-defined patterns, the watermark-based SSDs can be perturbed easily.

Several studied in [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. One of studied by [19] was created a method to influence SSD and studied in [15] also was provided a testbed through which the SSD approach can be examined. [14], on the other hand, provided Stepping Stone Detection 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. [18] once more continued their research on Neural Network was used in SSD but this time it focused on improving the performance of Neural Network techniques used in their previous research. [19] [20], on the other hand, proposed a clustering-partitioning algorithm to find TCP packet's RTT.

In [18] was also involved in a research that focused on SSD perturbation. In their work, they proposed the packet fluctuation approach by generating two algorithms and test them when chaff perturbation exists and studied in [23] was proposed the association rules technique to solve SSD in a host-based environment. In their research, they used association rules for detecting connection chains in a host-based environment. Clearly, 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].

C. Future SSD

SSD research that would become the focus of researchers in the future is inferred from the literature of the past and present SSD researches. With regards to the type of data that would be used in future SSD research, it is predicted that timing-based data (and RTT) would still be chosen. This fact actually comes from the past and present SSD which used timing-based data as the data needed to be captured in detecting stepping stones. Research by [1] changed to timing-based data after researchers

realized that the type of data using payload could not be used in encrypted connections [3] and [6]. Research using RTT as the type of applied data were [7], [24], [25], who could also be categorized as researchers using timing-based data type because RTT actually uses time as its main data type.

The difference is that the RTT-based research relies on the echoes of the packets. In conclusion, timing-based data would become a data type to be used in future SSD research. Secondly, research on SSD would focus on the applications of AI techniques. Whether the AI technique is used to obtain RTT or more will be discussed later.

Recently, [23] proposed the associative rules technique to solve the problems of SSD by taking arrival time as the data input. In their research, connection chains were traced by just using the packet's arrival and departure times.

Future SSD would also 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.

V. The New 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. These consist of spam detection, proxy server detection, backdoor detection and Dos attack detection. Figure 7 shows the classification of new applications based on SSD. Each new application that inspired by the SSD will be discussed later on the next discussion.

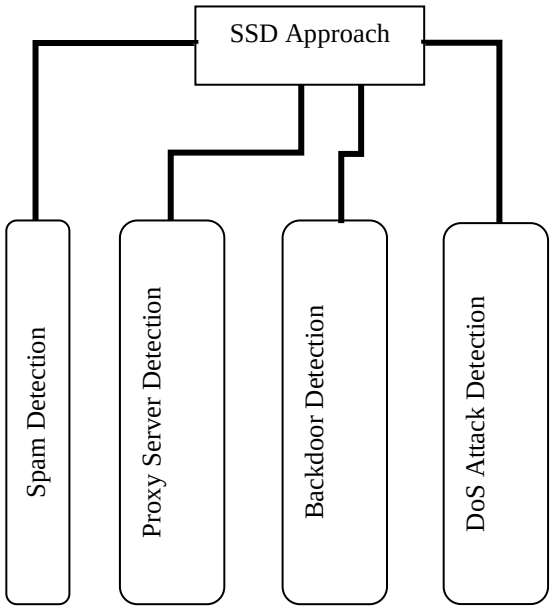

Figure 7: New Application for SSD

A. 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 by deleting the spammed e-mail directly from user e-mail's mailbox as we. In [30] on the other hand, 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.

Figure 8 shows a snapshot of a typical spam received in a mail box. The spam e-mail appears as an advertisement to the user. Other purposes for spamming include phishing and fraud. 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.

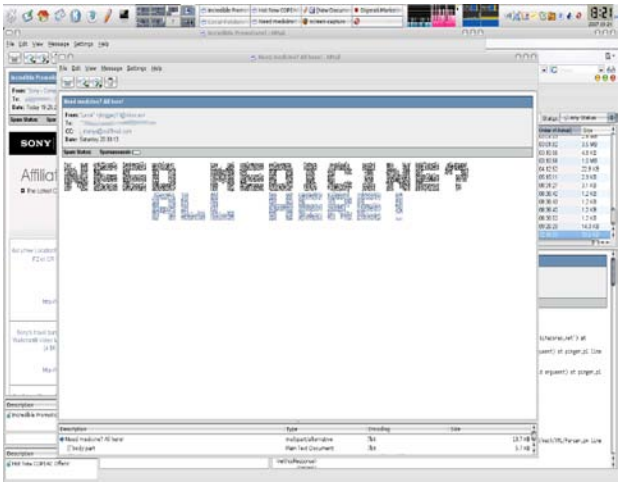

Figure 8: Example of Spam

The spam e-mail appears as an advertisement to the user. Other purposes for spamming include phishing and fraud.

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

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

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.

B. 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]. Figure 9 shows the general setup of a proxy server.

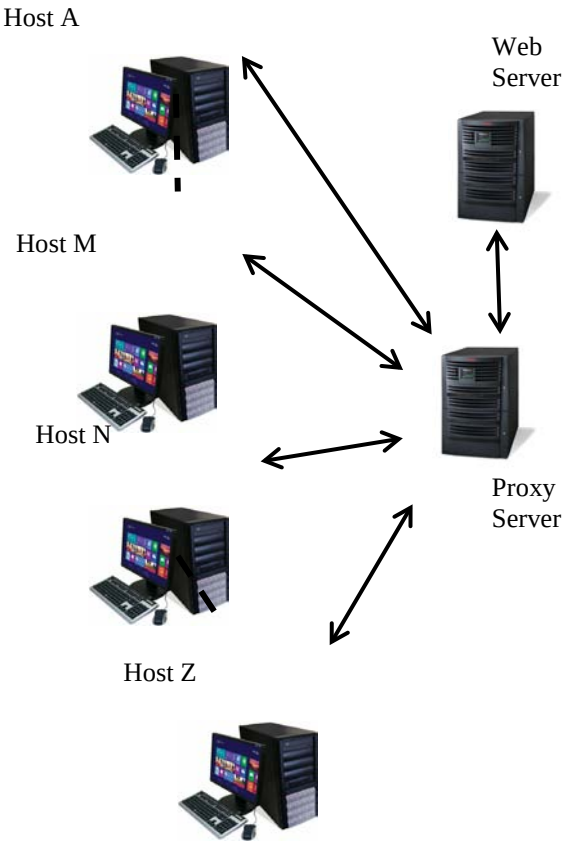

Figure 9: General Setup of Proxy Server.

From Figure 9, it is clear that proxy server will act as an intermediary for connection requests from the hosts or clients to the web server. Therefore, the proxy server will temporarily store any data transmitted between the hosts. Detecting the proxy server is important because it will prevent the user from remaining anonymous in the network.

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. 10.

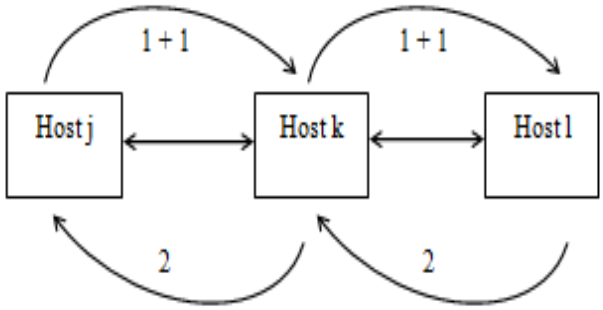

Figure 10: Proxy Server Communication

Form Fig. 10, 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:

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

where

k is the last number of host. By assuming

Hosts, *d* also including Hostd,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.

C. Backdoor Detection

Backdoor can be defined as a hidden approach for bypassing normal computer authentication systems [37]. The backdoor program can be an installed program or exist from the system processes. Backdoor programs are also embedded from various worms such as Sobig and Mydoom.

In August 2009, Sophos Labs discovered the W32/Induc-A virus which infected a program complier for Delphi [49]. The virus introduces its own code to a new program that infects many systems unknown by the programmer. This backdoor attack worked elusively in the background until discovered a year later. This demonstrates the danger of backdoor attacks.

Most of the time, antivirus solutions are able to thwart backdoor intrusion [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).

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

Referring to (1),

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

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.

D. DoS Attack Detection

A DoS or Distributed Denial of Service (DDoS) attack is a kind of attack that attempts to make network resources unavailable [39]. Usually, DDoS attacks occur to websites hosted on high-profile web servers such as banks or credit card payment gateways.

There are several types of DoS/DDoS attacks. SYN flooding is a type of DoS attack where many SYN packets are sent and never acknowledged, delaying other users from accessing the server and in severe cases jeopardizing users when the server shuts down completely [40].

DoS attacks may be solved by manipulating firewall settings [41]. Research by [42] on the other hand addresses DDoS by using a special Intrusion Detection System (IDS). In [43], DDoS is solved by using neural network. A variety of artificial intelligence may be used for addressing DoS attacks.

The basic of the DoS attack illustrated in Figure 11.

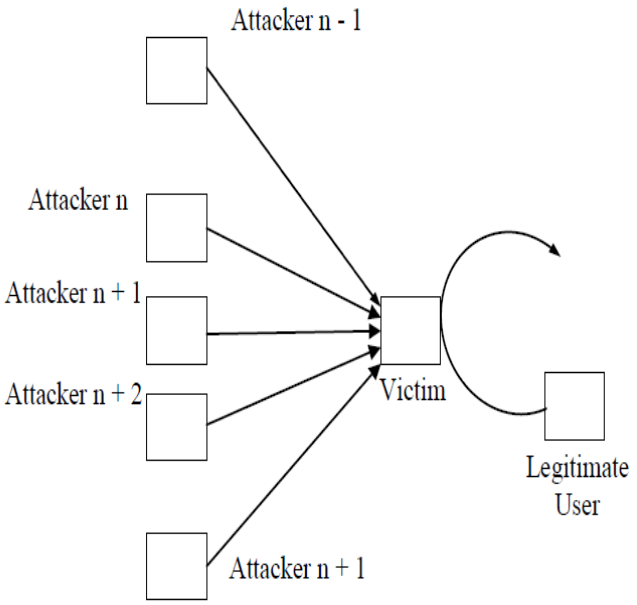

Figure 11: DoS Attack System

From Figure Assuming there are several attackers, sending many requests to the victim simultaneously, therefore causes the legitimate user to be unable to send any request to the victim. This is the basic nature of the DoS problem.

From the SSD perspective, these can be expressed as:

$$\text{DoS}_{\text{SSD}} = \begin{cases} 1, & if\ n_{in} < n_{out}\ for\ \forall n \\ 0, & if\ n_{in} = n_{out}\ for\ \forall n \end{cases} \quad (8)$$

Referring to Eq. (8), by locating the victim as the stepping stoned host, the stepping stone is detected if the number of the incoming flow is less than the number of outgoing flow for the host. The number of incoming and outgoing flows should be balanced for each host. If *k* is the last of host n and based on Eq. (2), it is possible to find the origin of the DoS attack.

VI. PRELIMINARY RESULT

Section 5.1 to Section 5.4 illustrate that each type of security threat may be characterized by equations given in Table 1 to solve its respective problem.

Table 1: Application and Formula

No.	Application	Formula
1	Spam Detection	$\text{SPAM}_{\text{SSD}} = \begin{cases} 1, & if\ n_{in} < n_{out} \\ 0, & if\ n_{in} \neq n_{out} \end{cases}$
2	Proxy Server	$\text{Proxy}_{\text{SSD}} = \begin{cases} 1, & if\ n_{in} = n_{out} \\ 0, & if\ n_{in} \neq n_{out} \end{cases}$
3	Backdoor	$\text{Backdoor}_{\text{SSD}} = \begin{cases} 1, & if\ n_{in} = n_{out} \\ 0, & if\ n_{in} \neq n_{out} \end{cases}$
4	DoS Attack	$\text{DoS}_{\text{SSD}} = \begin{cases} 1, & if\ n_{in} < n_{out}\ for\ \forall n \\ 0, & if\ n_{in} = n_{out}\ for\ \forall n \end{cases}$

Table 1 shows that each network threat requires a very simple formula for detection. SSD provides a simple solution for addressing the network threat at hand, thereby eliminating the need for more computationally expensive methods for detection. SSD may also be used in conjunction with existing detection systems forming hybrid techniques for analyzing and detecting these specific network threats.

Further SSD-based applications are conceivable in the future and the applications listed are not exhaustive. However, the preliminary models in this paper are intended to demonstrate that several online security threats may be unraveled from the SSD dimension, in conjunction with conventional traffic analysis techniques.

VII. CONCLUSION

For the detection of series of host computers by attackers, SSD has untapped potential in several emerging research fields, namely in spam, backdoor, proxy and DoS attack detection. Four novel SSD models are presented to demonstrate the potential of SSD in addressing current issues in spam, backdoor and proxy detection.

For future work, extensive SSD simulations and verification on real data such as wireless network [44, 45], and mobile wireless network [46] using CI methods [47, 48] for each emerging domain 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," 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.

[44] L. A. Peting, Introduction of a new network reliability model to evaluate the performance of sensor networks", *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 5(3), pp. 577 – 585, 2011.
[45] Nurhayati, S. H. Choi, and K. O. Lee, "A Cluster Based Energy Efficient Location Routing Protocol in Wireless Sensor Networks", *International Journal of Computers and Communications*, vol. 5(2), pp. 67 – 74, 2011.
[46] D. Ofrim, D. Sacaleanu, R. Stoian, and V. Lazarescu, "A 3-dimensional Localization Algorithm for Mobile Wireless Multimedia Sensor Networks", *International Journal of Communications*, vol. 5(4), pp. 149 – 156, 2011.
[47] R. Din and A. Samsudin, "Digital Steganalysis: Computational Intelligence Approach" *International Journal of Computers*, vol. 3(1), pp. 161 – 170, 2009. ISSN: 1998-4308.
[48] A. Popov, "Genetic Algorithms for Optimization – Application in Controller Design Problems", pp. 1 – 21, 2005. *Retrieved from http://p0p0v.com/science/downloads/Popov05a.pdf*
[49] M. Erez, "Managed Code Rootkits: Hooking into Runtime Envoronments", Syngress, 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA, 2011.

Mohd Nizam Omar is a Senior Lecturer at the School of Computing (SoC), UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia (UUM). He received his Bachelor of Computer Science with Honors and Master Science of Computer Science from Universiti Teknologi Malaysia, Skudai, Johor in 2000 and 2005, respectively. He recently received his Ph.D in Computer Science from Universiti Sains Malaysia, Penang, Malaysia in 2011. Dr. Mohd Nizam is engaged in several professional societies such as in IEEE and IAENG and welcomes participation in other professional societies after this.

Angela Amphawan is currently a Senior Lecturer at the School of Computing, UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia. She received her Bachelor of Engineering (Hons) and Master in Engineering Science from Multimedia University, Selangor in 2001 and 2003 respectively. She later completed her Ph.D. from University of Oxford, United Kingdom in 2009. She has published prolifically in reputable journals in mathematics, physics, computer science and engineering.

Roshidi Din is a Senior Lecturer at the School of Computing (SoC), UUM College of Arts and Sciences (CAS), Universiti Utara Malaysia (UUM). He received his B.I.T and M.Sc.IT degrees from Universiti Utara Malaysia in 1996 and 1999, respectively. Since working over 14 years in UUM, he has published his work in more than 50 papers in conferences and international journals publication. His current research interests lie in information security, steganology and steganalysis.