International Journal of
Electronic Commerce Studies
Vol.2, No.2, pp.103-126, 2011

A SURVEY OF RESEARCH IN STEPPING-STONE DETECTION

Robert Shullich
John Jay College of Criminal Justice
rshullic@mindspring.com

Jie Chu
Graduate Center
jchu1@gc.cuny.edu

Ping Ji
Graduate Center
pji@jjay.cuny.edu

Weifeng Chen
California University of Pennsylvania
chen@calu.edu

ABSTRACT

Stepping-stone is a method that directs network connections from an attacker to a victim through one or more intermediate compromised systems or devices. The objective of this scheme is to hide the attacker's identity (provide anonymity) and make traceback either difficult or impossible. Evasion techniques that are used to hide this process include encryption, introduction of dummy packets (chaff) into the stream, introducing delay into the timing of the packet stream, using multiple compromised hosts in long connection chains (many hops), and intermixing command and control traffic with multimedia traffic to mask traffic characteristics. This paper provides a survey that focuses on characteristic based, interactive stepping-stone detection and analysis techniques. An overview of the field of research is presented with critique of some of the methods used. We also provide some interesting topics for additional research.

Keywords: Network Security, Intrusion Detection, Stepping-Stone, Connection Chain, Chaff, Watermarking, Information Flow Detection, Flow Correlation, Stepping-Stone Intrusion

1. INTRODUCTION

One of the attack methods used by a computer network attacker is to compromise a device, take over control, and use it as a drone for attack purposes. One of the reasons for this method is to maintain the anonymity of the attacker so the attacker appears not to be directly involved. Another reason is that compromising multiple machines to create an army of "slave devices" makes it easy to launch a distributed denial of service attack (DDoS). The infected victim hosts are called zombies, and newer terminology may reference such victims as bots. Direct forensic analysis of such devices may easily identify and expose the attackers, so to further obfuscate the connection, attackers will use intermediate compromised devices to form a connection chain of many hops between the attacker and victim. Current terminology for these intermediate devices is *stepping-stones*. These are very similar to the concept of application level proxies and operate in almost the same way. There are at least two objectives in the use of stepping-stones. The first is to make it extremely difficult or impossible to trace back through the stepping-stones to the true attacker. The second is to prevent detection of the stepping-stone stream to maintain the attacking chain connectivity for as long as possible. The idea is to prevent the infected stepping-stone nodes from being discovered and taken offline. Techniques for preventing detection include encryption of the traffic, introduction of dummy packets called *chaff* to change the properties of the conversation, and introduction of timing perturbations to change transmission characteristics.

Research in the past decade has focused on how to detect stepping-stones. Content based analysis[1, 2] examines the payloads; however, the use of encryption prevents content-based analysis since the contents are not exposed. Researchers consequently proposed other approaches to detect stepping-stone using methods that analyze packet transmission characteristics, such as timing or counting. This paper will examine and summarize various papers in this field of study and point out areas that may be interesting for future research.

The process of identifying stepping-stones falls into the area of intrusion detection systems (IDS). These come in different flavors such as signature based and characteristic based. Signature based is not examined in this paper as it would require exposed content, which does not work on encrypted traffic. Characteristic based approaches can be further divided into two sub-categories: similarity-based and abnormality-based[3]. Among characteristic based approaches, active and passive methods (such as thumbprints and watermarks) have been explored. Passive methods just examine the data stream, whereas active methods attempt to modify the

transmission stream. Thumbprints[4, 5] are passive methods to develop a signature by matching some attribute of the packet or packet flow. One active method explored in some papers is the process of watermarking[2, 6, 7, 8, 9] : the packet or packet flow is modified to insert a signature that is encoded (inserted) at one point and decoded (recovered) at another point.

The type of information that a method exploits could also be a criteria for classification. Many of the timing methods[9-19] are based on some aspect of packet timing. Inter-packet delay (IPD) is the timing between packet arrival times. Round-trip Time (RTT) represents the time for a transaction to travel round-trip, and is usually the timing between the send and echo commands in these papers. Some additional methods exploit other connections attributes. We will look at these methods in this paper.

The rest of this paper is organized as follows: Section 0 provides some common definitions used in this survey. Section 0 summarizes various papers using the category classifications mentioned above, and in some cases individually critiques the papers. Section 0 surveys other work that does not belong to these category classifications. Section 0 provides interesting topics for future research. Section 0 provides summary and final thoughts.

2. DEFINITIONS

We first present definitions that will be used in this survey.

Stepping-stone refers to the intermediary host of a connection chain which is usually invaded by the attacker.

Causality-constraint means that a packet has to arrive first before it can leave a node.

Chaff is dummy packets that cannot be distinguished from real attack packets.

Delay-constraint is the maximum tolerable delay.

Maximum tolerable delay is a limit on how long a packet can be delayed at a Stepping-stone. It is considered by almost all papers to be bounded and may also be termed delay bound, bounded delay, or delay constraint.

Order-constraint is when packet order is maintained.

Packet-conservation is when no packets are generated or dropped.

Repacketization can either combine two or more closely adjacent packets into a larger packet or split a packet into multiple smaller packets. This may occur at any node, including a router. This may occur due to fragmentation, but there may be other causes.

Think-time is a delay in the processing of a request to reproduce the time that a human would take to read or examine the data displayed from a previous user action.

3. TYPES OF STEPPING-STONE DETECTION TECHNIQUES

Almulhem & Traore[3] previously wrote a survey on connection-chain techniques that covers at least 14 papers of interest in interactive stepping-stone detection analysis. Table 1 summarizes a section of their work. Many of the 14 papers represent foundations in stepping-stone detection and are frequently cited. While Almulhem & Traore[3] ended their survey in 2006, we mainly explore papers that were written in 2006 and beyond to cover more recent work in this field. In some cases, earlier work is still presented.

Table 1. Prior work previously outlined in survey[3]

Scheme	Authors	Year	Reference
Thumbprint	S. Staniford-Chen and L.T. Herberlein	1995	[1]
ON/OFF	Y. Zhang, V. Paxson	2000	[19]
Deviation	K. Yoda and H. Etoh	2000	[18]
IPD	X. Wang, et al.	2002	[15]
Multiscale	D. L. Donoho, et al.	2002	[11]
Send-ack/Send-echo	K. H. Yung	2002	[20]
Watermark	X. Wang, D. Reeves	2003	[9]
State-Space	T. Strayer, et al.	2003	[21]
Detect-Attacks	A. Blum, et al.	2004	[10]
Real-time Analysis	J. Yang and S.-H. Huang	2004	[22],[23]
Delay + Chaff	L. Zhang, et al.	2006	[14]
Signal Processing	T. He and L. Tong	2006	[24]
Watermark Secrecy	P. Peng, et al.	2006	[6]

This section begins with papers that explore packet transmission characteristics such as timing (Section 3.1) or counting (Section 3.2). This is followed by approaches using passive methods such as thumbprints (Section 3.3) and active methods such as watermarking (Section 3.4). Abnormality-based approaches are then surveyed (Section 3.5).

3.1 General Timing-based

J. Yang and S.S. Huang[17] provided a packet matching method to detect long connection chains by using a step function and analyzing RTT for send and echo messages. The concept is to use Deviation-Based Clustering to find the RTT. Two algorithms were described: a conservative one and a greedy one. When the attacker extends the connection chain by one hop, the observed RTT at the monitor node will increase by approximately the latency of that hop. Therefore, we can observe a step function between RTT and length of the connection chain. Through changes in RTT, one can detect the length of the chain and determine how many nodes the connection traverses. The authors assume that the connection is interactive, since it will have *think-time*, and that chains are not longer than three or four nodes. A drawback is that this algorithm has to monitor the entire connection session. Yang *et al.*[25] researched how to compute reliable and sufficient RTT, extending their work in[17] using data clustering and partitioning to find RTT. The process examines packets in groups and uses a Max-min distance clustering algorithm, assuming only one connection between hosts in the chain. The process computes and matches differences of send and echo times. However, it still has the same drawback as in[17] stated above.

Zhang *et al.*[26] provided algorithms that detect stepping-stone traffic with bounded delay perturbation and/or chaff. This paper aims to provide an effective algorithm under timing perturbations, especially when delay and chaff perturbations exist simultaneously. The authors assumed time synchronization between hosts, bounded delay, chaff perturbation independent of the original flow, and no packet loss. They proved that this strategy has an exponentially decaying false alarm probability for independent Poisson streams. However, stepping-stone traffic can be detected if chaff is only inserted in the departing stream. The authors leave for future work the task of relaxing the assumptions, addressing packet retransmission and dropping, and merging and splitting flows. A weakness in their approach is that a significant number of packets need to be collected in order to keep false positive and negative rates low.

He *et al.*[27] stated that there are still fundamental limits to stepping-stone attacks, even with chaff insertion, timing perturbations, padding, and encryption,. They defined a mapping process between the packet arrival and packet departure times at a specific host called *bijection,* which also allows permutation of packets during the relay. There are constraints such as packet-conservation and causality. *Maximum tolerable delay* is defined and sets a limit on how long a packet can be delayed at a stepping-stone. It is suggested that one hop stepping-stones are difficult to detect because the attackers can make the traffic look like anything they want. A theorem is presented which shows that if an attacker wants to hide the connection chain path, the amount of *chaff* packets increases exponentially with path length. With fundamental limits set, the results also require the attacker to mimic Poisson processes. The analysis is that stepping-stone attacks can be impeded through randomizing packet transmissions that allow flows to be traced through unique timing characteristics. However, this process could have negative impacts on time sensitive traffic such as multimedia. The authors do not go into detail about how packet scheduling is performed. It can be assumed that some modification of the host network stack may be required to control packet flows. Such active processing would require host access or control in order to make such modifications. The paper presented by T. He and L. Tong[28] is an extension of this packet scheduling paper[27]. Their objective is to build a general form detector that is specifically designed to provide guaranteed detection in the presence of chaff noise. This will be dependent on their prior research; as the connection chain gets longer, it becomes more difficult to hide an information flow. The resultant bounds set on various variables provide limits that can be used to construct a relatively precise detector. Analysis for building the detector is based on independent Poisson processes, but with adjustment it can be used for other types of traffic.

Strayer *et al.*[29] indicated that prior research only focused on one aspect or attribute such as similar content, similar packet arrival time, or similar burst times. In addition, the computation costs of pair-wise comparison were high, on the order of N^2. The authors worked on building a new flow correlation algorithm to track more than one flow characteristic at a time. Flow characteristics should be dynamic, expressed as a time series, and measure something about the flow. They proposed ten characteristics based on these standards, and used principal component analysis (PCA) to select the proper characteristics which contribute the most variance for all flows and the least variance for correlated flows. Each flow has a vector in n-space if there are n characteristics, and the correlation is determined by calculating a Euclidean distance within n-space.

Strayer *et al.*[30] discussed a system for performing IP Traceback and integration with a stepping-stone detector. Instead of using just one algorithm, they organized what they called a "master function" that created a composite score over four selected stepping-stone detection algorithms: On/Off, CLT, State Space, and Thumbprints. Some algorithms worked better than others, as each performed differently based on the stream being presented. Use of multiple algorithms allowed the authors to exploit the advantages and mitigate the disadvantages of the selected algorithms.

B. Coskun and N. Memon [31] discussed relay nodes and classified them as either store-and-forward or delay-constraint. Stepping-stones belong to the delay-constraint category. Prior work is identified that solves a relay flow problem correlating flows to search for flow pairs. This is a harder problem than relay node identification and requires an order of complexity of quadratic time for each node. The complexities of identifying a relay node with their research can be solved in linear time, and this could be used to first filter out the stepping-stones then to provide a smaller subset to perform correlation against. This partitions the overall problem and limits further analysis to identified relay nodes rather than an entire network. The basic idea works as follows: time slots are allocated; based on the assumption of *maximum tolerable delay*, if a node is a relay, then in any time slot one node is transmitting while another node is receiving; when two nodes appear often in a time slot, one of them is probably relaying to the other. Some traffic, such as TCP, is bidirectional and could be picked up automatically. This is compensated for by checking the reverse direction (IP addresses and port numbers swapped) and eliminating the return flow. Restrictions are imposed; in order for the algorithm to be successful, all nodes need to be *delay-constrained*, and traffic needs to be sparse so there are empty time slots. If the nodes are not delayed-constrained, the *maximum tolerable delay* may be violated. If the traffic is not sparse, it will match with all other flows and be tagged as a relay because it would always be active and appear in all time slots. In his thesis paper, Padhye *et al.*[32] develops an attack kit called SNEAK which provides a sender-side and receiver-side pair of algorithms that control packet loss as needed. This SNEAK generates a stream that would not be detected by this method, as it is not sparse and violates the assumption of *think-time*. Future work is required to improve on the false positive and negative error rates.

H-C. Wu and S-H. S. Huang[33] presented a detector using neural networks (NN). They assumed that legitimate use of stepping-stones would not go through more than two hops, so any connection chain longer than two hops would be suspicious. Three preprocessing steps were used to prepare the data for testing and training. In the first step, ten variables were

extracted from the TCP packets that were captured, including packet size, source port, destination port, source IP, and destination IP. The second step computed the RTT using a NN. The authors trained the neural network with known stepping-stone connection data, and some additional sample streams were used to test if the NN learned to detect stepping-stones. Performance varied depending on the number of connection paths, and the process depends on having representative data to conduct NN training. A key argument for this process is that an entire session does not need to be monitored; only a short interval of monitoring is required.

The authors extended this work in[34]. They proposed another scheme of neural networks for stepping-stone detection. The input layer accepts *n* inputs which are *n* consecutive RTTs in a flow. This new scheme could exploit the effect of RTT fluctuation of packets. The neural network was trained with the data they collected, and the model was evaluated in a similar way as previous work, with slight changes in the experiment settings.

Ahmad Almulhem and Issa Traore[35] proposed a host-based stepping-stone detection method employing a data mining technique called association analysis. Their implementation relies on Java Virtual Machine and a standard library to capture the packet; these are available with most operating systems. The result has less OS-dependence than other host-based methods. They used a public data set which contains traffic to a FTP server; the FTP session is interactive and used to mine the pattern helpful in finding interactive stepping-stone attacks. They also simulated connection-chain traffic in order to improve the true positive rate.

3.2 Packet Counting

T. He and L.Tong[12, 13] presented an algorithm that does not depend on timing or require traffic manipulation. It assumed that the memory in the host and timing delays are bounded. Packet orders are maintained, and *order-constraint* is required so that the analysis complexity can be kept linear rather than exponential. The paper introduced a counting based algorithm, DETECT-MAXIMUM-VARIATION (DMV). Three variations of this algorithm were presented, one for bounded memory and two to handle the case where timing perturbations and chaff insertion are simultaneously used.

Huang *et al.*[36] developed a connection-chain detection procedure used as a stepping-stone detection tool to analyze correlations between the frequencies of the cumulative numbers of packets sent in incoming and outgoing connections, supplementing the research in[10]. This process

correlates the connections based on some factor of the number of packets. Their objective was to develop a technique that could work with chaff and timing-jitter insertion, with reasonable error rates. Traffic is monitored in both directions and analysis is performed on the send/echo streams. The concept is that if the frequency of the send stream is linearly related to the frequency of the echo stream, then the stepping-stone is identified. There is an assumption that the conversation is interactive. The method works well in simulation when multiple connection streams pass through the same stepping-stone node, and user operations performed are similar. Their experiments did not yield all the required results, and as a result, their paper is incomplete.

3.3 Thumbprinting

J. Yang and S-H. S. Huang[4] introduced the concept of a temporal thumbprint (T-thumbprint). The T-thumbprint is based on the sequence of time gaps between adjacent TCP packets. These thumbprints are compared to determine which streams are connection pairs. The authors stated that there are several countermeasures an attacker can take to avoid detection, and within limits, the T-thumbprint can still be effective. It appears there are still methods of evading this detection that an attacker can take. As the technique can only handle up to 35% additional characters, throwing the timing off or adding a lot of chaff may be sufficient to avoid detection. The method does not require synchronized clocks and is passive, as the packet stream is not modified as part of the detection process.

The paper presented by J. Yang and S-H. S. Huang[5] is a follow-up of their initial paper on temporal thumbprints[4]. In this method, they thumbprint the RTT, which is computed as the round-trip-time between a pair of sending and echoing packets. Two different algorithms were presented: an exhaustive approach and a heuristic approach. The heuristic approach showed similar performance to the exhaustive approach but was more efficient. The RTT is sensitive to network fluctuation and will differ between local traffic and traffic that traverses the WAN. An issue that makes the process difficult is the matching of echo packets with the send packets. Many actions can throw the process off, including dropped and retransmitted packets. An advantage of RTT Thumbprints is that they can avoid random delay and chaff insertion.

3.4 Watermarking

The method presented by Wang *et al.*[2] is an active detection method that uses watermarking. It was termed sleepy because no overhead is

introduced for normal traffic unless intrusion is detected. It assumes interactive and bidirectional flows, and the same application, the *Watermark-enabled application*, is used along the chain because of the application dependent watermark. It is invisible to end users and can even track the connection when the intruder is silent. It also assumes that the message contents do not vary from hop to hop and therefore cannot be encrypted. Routers used as guardian gateways are assumed trusted.

Peng *et al.*[7] presented an active timing-based algorithm that will work with both chaff and limited perturbations. It requires interactive sessions and no packet loss, as well as *packet-conservation*, *order-constraint*, and *delay-constraint*, combining watermarking with packet matching. Their scheme injects watermarks into the upstream flow and tries to detect them in the downstream flow; therefore, chaff can only appear in the downstream flow. The attacker can detect the watermark and take counter action against detection because this is an active intervention that exposes the detection process.

Wang *et al.*[16] used a method of probabilistic watermarking where the inter-packet delay (IPD) is modified to represent either a one bit or zero bit. It also assumes that *order-constraint*, *packet-conservation* and *delay-constraint* are satisfied. The Hamming distance was used instead of a direct match to detect the changes and provide a better matching rate. The process also depends on a shared secret between the encoder and decoder. A weakness of IPD watermarking is that precise packet synchronization is required in order to decode the watermark.

Pyun *et al.*[37] presented another watermarking technique, which they defined as interval-based watermarking. Their work is an extension of the work in[9], addressing the *repacketization* problem. The algorithm divides the duration of each flow into short, fixed-length intervals and does not require precise clock synchronization between the watermark encoder and decoder, as "the intervals are self-synchronized during decoding." The process only requires a few hundred packets to make a decision, but the implementation cost includes timing modifications to add the watermark and deployment of monitors to capture and analyze traffic.

3.5 Anomaly Detection

Kampasi *et al.*[38] provided methods to improve stepping-stone detection when jitter and/or chaff are introduced into a packet stream. They claimed to have used real world traces and achieved 99% accuracy. Three separate methods were introduced: a response-time based method which detects jitter anomalies, an edit-distance method to detect chaff

abnormalities in packet traffic, and a causality based method to detect chaff anomalies in packet traffic. These three algorithms produce a framework that is difficult to evade when used to supplement existing timing-based correlation algorithms. The algorithms can be used for stream or collected data. The main premise of the design is that if the attacker adds jitter or chaff then the traffic will appear anomalous, initializing the three specialized algorithms. This all still depends on interactive stepping-stones and is also based on the ON/OFF correlation method. The algorithms are based on what the authors call magic numbers, which are required for the algorithm to be successful; however, the authors did not explain the nature of these magic numbers, their computation, or the complications involved in getting the right numbers. Dynamic adjustment of these parameters is left for future work, which leaves practical questions on how these success-critical magic numbers can be used without any information on how to tune them.

4. OTHER RELATED WORK

In addition to the papers surveyed in the previous section, some papers that use other techniques to detect stepping-stones are summarized in this section.

4.1 Optimization

In[39], the authors suggested that stepping-stone detection is a three-stage process: Packet Capture, Identification and Comparison. They also described optimization strategies for the three stages. By using packet filters, i.e., capture only the packets' headers, the elapsed time of the packet-capture process can be reduced, leading to a shorter detection plus response time. The time for the identification process can be reduced by examining only the header of the packet, ignoring the data portion. The comparison step can also be optimized by only processing one characteristic of connections, but this may have weaknesses due to the fact that multiple characteristics may help increase test accuracy.

In[40] and [41], Omar *et al.* supplemented the above research. In[40] the authors conducted an analysis of different buffer sizes for the packet capture step to determine how to improve and speed up packet capture. Simulations were developed with kernel/user buffer sizes set to 4MB, 64MB, and 1024MB. Their experiments showed that the packet capture process performed faster and captured more packets when the kernel/user buffer size was 4MB. In[41] the authors showed the impact of packet loss was significant and that packet loss would occur over a wide area network.

Most of the other papers presented assume no packet loss. However, this study was built upon simulations with a 25% packet loss rate. SNEAK could be used to evade most existing approaches which assume no packet loss.

X. Wang[42, 43] formulated the core of the stepping-stone detection problem: connection correlation and back tracing by using *set theory*. It shows that, regardless of correlation approach, all current research only solves part of the problem, even with perfect correlation. Stepping-stones and the connection pairs can be identified; however, the connection path, including direction and order, is not observed. All it takes to throw off current analyses is for the attacker to loop the same connection chain through the same stepping-stone more than once. This is what the authors refer to as the "loop fallacy". The process of reconstructing the connection chain requires a form of serialization, especially when loops are created. Through the use of set theoretic approach, the author can reconstruct the entire chain in the proper order and also determine the direction of the chain from attacker to victim. Construction of the entire chain and path relies on connection correlation, which is not yet well solved.

Tang *et al.*[44] addressed the monitor placement problem for stepping-stone detection. The authors formulated the problem into a graph problem. They proved its NP-Completeness by reducing the Graph Partitioning problem to this problem. Recursive greedy algorithms with different complexities for different goals were proposed. In their metric definition, they define *Average entropy* and *Worst-case entropy*. Average entropy is defined by the scenario where the attacker has no knowledge of network topology and monitor placement; Worst-case entropy assumes that the attacker has some knowledge that could be used to adjust traffic to avoid monitors. In this paper, the authors limited the process to cases where each node has an equal probability of being the origin of an attack. Future work is recommended to investigate the impact of unequal probabilities.

Y.J. Pyun and D. S. Reeves[8] also investigated the placement of network monitors. The goal of this paper is to determine the minimal number of close-to-optimal monitor placements for the desired level of accuracy. This optimization problem is also translated into a graph partitioning problem. Determination of this value is an intractable problem, so the authors used heuristics to yield acceptable non-optimal results. Monitor placement would be static, and the authors leave for future research the use of dynamic approaches.

4.2 Evasion Techniques

M. Venkateshaiah and M. Wright[45,46] proposed a buffering technique to avoid detection. By buffering the packet stream, the stepping-stone can transmit at an almost constant rate and remove the inter-packet delays. This is capable of breaking current watermarking methods. The packet stream shows constant rate and no variance in matching timing statistics. Most of the detection algorithms discussed here are geared towards interactive stepping-stones which depend on the existence of "*think-time*"; constant rate traffic no longer has this property.

4.3 Test Framework

Xin *et al.*[47] developed a standard test bed for stepping-stone detection. It does not require special hardware, yet provides features like interactive session generation with chaff permutations and time delay. The tool generates scripts to mimic an interactive connection which supports some of the most common commands. Plugins can be implemented for different stepping-stone detection methods. They demonstrated this with an experiment comparing two stepping-stone detection approaches, as described in [15] and [11]. Because the data provided by this system is not captured from the real world, it may not reflect all the characteristics of stepping-stone attacks. Most research studies prefers to collect test data and set up experiments on their own, with the exception of [14], which uses this test bed as a secondary experiment to the real trace data.

5. AREAS FOR FUTURE RESEARCH

This section describes some interesting topics for future research.

5.1 Hacker Motivation

When unauthorized activities are performed by a hacker, they may be either malicious or somewhat harmless. There are different motivations for hacking[48]: Self-Education; Financial Gain; Revenge; External Pressure; Terrorist, Political, and Issue Motivation Groups; and The Wannabe ("Walter Mitty"). This may be an important attribute to analyze, as the objectives and targets of the hacker may affect the tools and resultant data streams generated. As a quick example, if research is explored using UNIX environments, but the final attack target is a corporation and the attacker objective is financial gain through espionage, then the scenario may

possibly change, as most corporations use a Windows environment or mainframes.

5.2 The Cardinality Problem

A basic assumption in most existing papers is the correlation of stream connection pairs. This requires analysis of an incoming to an outgoing stream, assuming a one-to-one mapping (one input to one output connection). Some papers mention or address the issue of multiple sessions traversing through the same stepping-stone but it is still one attacker vs. one victim.

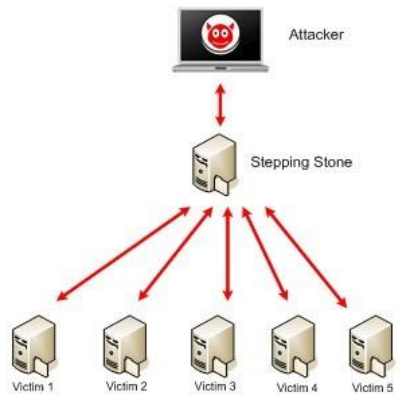

Figure 1. One to many attack scenario

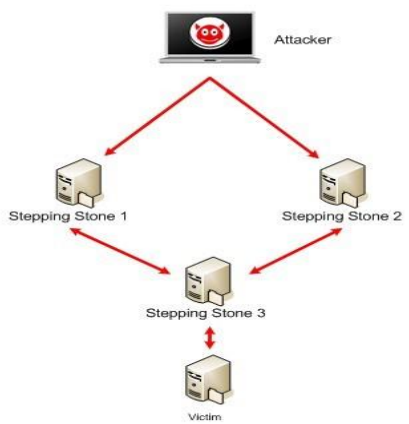

Figure 2. Many to one attack scenario

Suppose one attacker is interactively attacking five victims (See Figure 1). To effectively correlate traffic in this type of scenario, the traffic between the attacker and the stepping-stone may need correlation with the aggregate traffic over all five victims. In Figure 1, there are six connections: one connecting the attacker with the stepping-stone and five connections connecting the stepping-stone with the victims. It is no longer a one-to-one "connection pair". Another variation is the many-to-one scenario. As seen in Figure 2, we have stepping-stone 3 with two input streams and one output stream, with a single attacker conducting the attack.

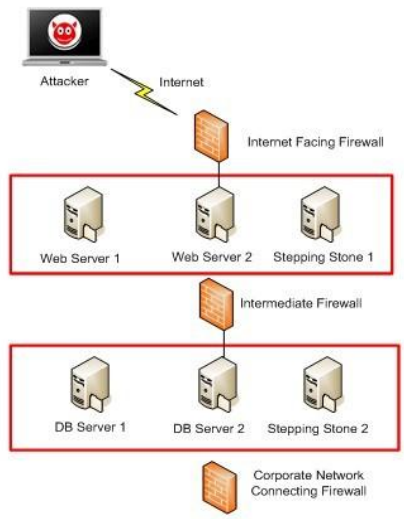

Figure 3. Corporate Multi-Tier DMZ Configuration

5.3 Corporate Enterprises and Firewalls

An attack on a larger LAN, such as a corporate enterprise network, may encounter various firewall configurations. Figure 3 shows an example of a multi-tiered DMZ (demilitarized zone) configuration.

In Figure 3, we have a connection from a Web farm to a database farm, and both farms are separated from each other by a firewall. Another firewall separates the Internet from the web farm, and the last firewall separates the database servers from the corporate network. The database servers are the point of interest, as they are usually where the crown jewels reside. This multi-tiered structure is a specially crafted infrastructure, and when configured correctly, it provides effective hurdles that slow down attackers. An example of good configuration principles is to allow HTTP/HTTPS through the gateway firewall but not the intermediate firewalls. This forces

the use of different protocols between the attacker and stepping-stone 1 (in the web farm) and between stepping-stone 1 and stepping-stone 2 (in the DB farm). However, existing research mainly assumes that the tools or traffic patterns are the same all along the chain. Research may be needed to address the change in traffic due to the change in tools and protocols required as each firewall is traversed. None of the existing work seen addresses the interference provided by firewalls; firewalls are not addressed or mentioned at all. Figure 3 provides good principles to block any inbound Telnet or SSH sessions at the gateway firewall, as well as blocking any outbound traffic for any protocol or port that is initiated from within the DMZ. This should be done at the protocol level, not the port level, so the firewall is protocol aware in case an attacker attempts to run SSH on port 80 or 443.

5.4 The WAN Environment

Most of the research surveyed has been conducted in a lab environment, such as simulations run on a local area network (LAN) which presents ideal situations. However, within a wide area network (WAN), queuing delays and packet loss may occur. At least one paper addressed this issue[41]. However, research needs to be expanded to address timing delays introduced by the network, packet loss and other factors beyond the control of the attacker. Other anomalies introduced by a WAN also need study. The paper on packet loss[41] only proved packet loss as an issue, but there is still need for algorithms that detect stepping-stones with the packet loss assumption. This may require device performance analysis and overall tolerance for packet loss at the monitor.

5.5 Operating System and Tools

The surveyed papers base their analysis on either Telnet[49] and/or SSH[50, 51] protocols and transmission characteristics. This makes sense, as most of the work has been university research in environments where UNIX or UNIX variants such as Linux are available and widely used. However, research has not addressed much in the areas where Windows operating systems are used as stepping-stones or attack targets.

Since Windows operating systems are heavily used both at home and in corporate environments, it makes sense to examine Windows use. This may have additional significance, as services such as Telnet and SSH are not available in Windows by default. This means that if an attacker was running a stepping-stone chain through a Windows system, the attacker would have to provide the appropriate servers and clients as part of the

infection compromising the system into a stepping-stone. In general, this is not a foreign concept since many of the current worm infections create their own SMTP server to transmit SPAM e-mail. There are other tools that are used for communications, such as Netcat, Cryptcat, Stunnel, Symantec pcAnywhere and Dameware. VNC is another remote access tool used on Linux and Windows that provides full screen capabilities. A professional version is also available that provides encryption. Each of these remote access control products may have different flow characteristics as they work in full screen mode, which is different from character or line modes.

5.6 Other Protocol Issues Including Covert Channels

Most of the analysis in the research papers addresses connection pairs. This assumes that a connection can be found and analyzed. Telnet and SSH are TCP protocols under the TCP/IP network stack. How does one address stepping-stone control using a protocol such as UDP? Although machine automated rather than interactive, the storm worm uses the Overnet/eDonkey protocol which rides on the UDP protocol. There are no sessions in UDP since UDP is connectionless. Research is missing on the use of UDP as a stepping-stone attack carrier. Even ICMP traffic could be used, with attack commands imbedded in the ICMP header[52] to create a covert channel. Analysis of Botnets[53,54] has been done with analysis of using UDP and ICMP as command and control tools for controlling Bots. Analysis is different for interactive stepping-stones because the traffic characteristics provide more hints about the packets and may be unique to Bots and their detection. Peer-to-peer (P2P) networks, which use UDP and ICMP, should also be feasible for interactive control of a stepping-stone and require further research.

When the researchers focus on TCP protocols, the volume of network traffic is too great to completely store and analyze. People use filters to get rid of the irrelevant packets. But similar to the evasion technique developed, an attacker could use other applications based on TCP for commands and control, for example, IRC. Other application-layer protocols could also concede the command path. A connection chain could be seen as a combination of several chains when multiple protocols are used along the chain. It may have only one chain on the IP layer because all transmissions are done in an IP network. However, if the attacker uses both SSH and IRC as the command and control channels, we will see different portions of the connection chain when we focus on one protocol. Some work[55] has been done to detect botnets which use IRC as their command and control channel. It would be helpful if the prior work in botnet detection could be combined with the current stepping-stone detection methods.

5.7 Wireless Network

Nowadays, many Wi-Fi hotspots have been widely deployed due to the price drop of off-shelf Wi-Fi access points. The combination of 3G and 4G networks with more powerful mobile devices is making it easier for people to access the Internet whenever they want. Mobile networks do and will contribute a huge portion of Internet traffic. In wireless/mobile networks, the transmission media is switched to air from all kinds of wires or fibers. The transmission error rate is also higher in a wireless network, which means more packets are corrupted and need to be retransmitted. However, most research done for stepping-stone detection usually tries to ignore the case of packet drop or retransmission. Another fact is that people are no longer staying in the same place when they surf the Internet, and network conditions are more likely to change within a connection. This mobility could lead to different characteristics of the network connection. Wireless sensor networks have also been widely deployed. Those sensors are usually customized systems with limited resources that may be more difficult for attackers to invade. It could still be a potential target in the future. Despite the increasing importance of the wireless network, few researches studies have been done for this different network environment. Potential work could be to verify the characteristics of wireless network traffic and how much they affect stepping-stone detection.

5.8 Theoretical Analysis and Experimental Study

Stepping-stone detection is a hard problem. But how hard is it? At this time, the stepping-stone detection problem is not well defined. X. Wang [42,43] formulated connection correlation as a relation over a set, denoted as CORR. Under the assumption of perfect correlation, she investigated the serialization problem. However, the author skipped most works that have been proposed to compute this CORR relation. In[10], the authors proposed a counting packet method to match two streams and "analyzed provable (polynomial) upper bounds on the number of packets needed to detect and identify stepping-stone streams". However, this method has difficulties when attackers insert chaff into the traffic. It becomes even more complicated when attackers apply more evasion techniques. Investigating the stepping-stone detection problem from a theoretical perspective could be another potential direction.

Some of the research like[14] compared their scheme with previous works. Most research work did experiment with their settings, but experimental research evaluation across different methods under the same

environment settings, especially recent approaches, is worthy of further study.

6. CONCLUSIONS

Stepping-stone detection is a complex problem. With the use of encryption to hide packet payloads, solving this problem requires techniques that are similar to black box testing or black box analysis. In black box operations the internal structure is unknown; in stepping-stone connections the payloads are unknown. Analysis requires inspection of the stream behaviors in order to predict what is going on inside. The papers surveyed here and in the previous survey[3] have applied mathematical methods to study and predict behavior using set theories, Poisson analysis, correlation, other statistical methods, queuing theories, anomaly detection, neural networks, and genetic algorithms. These have been applied to attributes such as packet counts, IPD, thumbprints, watermarks and RTT. To round off this survey, papers were shown that provided suggestions for optimization, monitor placement, and constraint issues.

7. ACKNOWLEDGEMENT

This work is supported in part by National Science Foundation grant CNS-0904901 and National Science Foundation grant DUE-0830840.

8. REFERENCES

[1] S. Staniford-Chen, and L. T. Heberlein, Holding intruders accountable on the internet. *IEEE Symposium on Security and Privacy*, Washington, DC, USA, May 8-10, 1995.

[2] X. Wang, D. S. Reeves, S. F. Wu, and J. Yuill, Sleepy watermark tracing: an active network-based intrusion response framework. *The 16th international conference on Information security: Trusted information*, Norwell, MA, USA, June 11-13, 2001.

[3] A. Almulhem, and I. Traore, A survey of connection-chains detection techniques. 2007 *IEEE PacRim Conference on Communications, Computers and Signal Processing*, Victoria, B.C., Canada, August 22-24, 2007.

[4] J. Yang, AND S.-H. S. Huang, Correlating temporal thumbprint for tracing intruders. *International Conference on Computing, Communications and Control Technologies*, Austin, Texas, USA, July 24-27, 2005.

[5] J. Yang, AND S.-H. S. Huang, Improved thumbprint and its application for intrusion detection. *Proceedings of International*

Conference Networking and Mobile Computing, Zhangjiajie, China, August 2-4, 2005.

[6] P. Peng, P. Ning, and D. S. Reeves, On the secrecy of timing-based active watermarking trace-back techniques. *IEEE Symposium on Security and Privacy*, Washington, DC, USA, May 21-24, 2006.

[7] P. Peng, P. Ning, D. S. Reeves, and X. Wang, Active timing-based correlation of perturbed traffic flows with chaff packets. *International Workshop on Security in Distributed Computing Systems*, Columbus, Ohio, USA, June 6-10, 2005.

[8] Y. Pyun, and D. S. Reeves, Strategic deployment of network monitors for attack attribution. *International Conference on Broadband Communications, Networks and Systems*, Raleigh, North Carolina, USA, 10-14 September 2007.

[9] X. Wang, and D. S. Reeves, Robust correlation of encrypted attack traffic through stepping stones by manipulation of interpacket delays. *ACM conference on Computer and communications security*, New York, NY, USA, October 27-30, 2003.

[10] A. Blum, D. Song, and S. Venkataraman, Detection of interactive stepping stones: Algorithms and confidence bounds. *Seventh International Symposium on Recent Advances in Intrusion Detection*, Sophia Antipolis, French Riviera, France, September 15–17, 2004.

[11] 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. *Fifth International Symposium on. Recent Advances in Intrusion Detection*, Zurich, Switzerland, October 16-18, 2002.

[12] T. He, and L. Tong, Detecting encrypted interactive stepping-stone connections. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, May 14-19, 2006.

[13] T. He, and L. Tong, Detecting encrypted stepping-stone connections, *Technical Report ACSPTR-01-06-02, Cornell University, 2006.*

[14] Linfeng Zhang, A. J. Anthony Persaud, and Y. Guan, Stepping-stone attack attribution in non-cooperative ip networks. *IEEE International Performance Computing and Conference*, Phoenix, Arizona, USA, April 10-12, 2006.

[15] X. Wang, D. S. Reeves, and S. Felix, Inter-packet delay based correlation for tracing encrypted connections through stepping stones. *European Symposium on Research in Computer Security*, Zurich, Switzerland, October 14-16, 2002.

[16] X. Wang, D. S. Reeves, P. Ning, and F. Feng, Robust network-based attack attribution through probabilistic watermarking of packet flows. *Technical Report TR-2005-10*, NC State University, 2005.

[17] J. Yang, and S.-H. S. Huang, Matching TCP/IP packets to detect stepping-stone intrusion. *International Journal of Computer Science and Network Security*, 6(4), p269-276, 2006.
[18] K. Yoda, and H. Etoh, Finding a connection chain for tracing intruders. *Proceedings of European Symposium on Research in Computer Security*, Toulouse, France, October 4-6, 2000.
[19] Y. Zhang, and V. Paxson, Detecting stepping stones. *USENIX Security Symposium*, Denver, CO, USA, Aug. 14-17, 2000.
[20] K. H. Yung, Detecting long connection chains of interactive terminal sessions. *Fifth International Symposium on. Recent Advances in Intrusion Detection*, Zurich, Switzerland, October 16-18, 2002.
[21] W.T. Strayer, C.E. Jones, I. Castineyra, J.B. Levin, and R.R. Hain: An integrated architecture for attack attribution. *BBN REPORT- 8384*, BBN Technologies, 2003.
[22] J. Yang, and S.-H. S. Huang, A real-time algorithm to detect long connection chains of interactive terminal sessions. *International conference on Information security*, Shanghai, China, November 14-16, 2004.
[23] J. Yang, and S.-H. S. Huang, Matching TCP packets and its application to the detection of long connection chains on the internet. *International Conference on Advanced Information Networking and Applications*, Taipei, Taiwan, March 28-30, 2005.
[24] T. He, and L. Tong, A signal processing perspective to stepping-stone detection. *Conference on Information Sciences and Systems*, Princeton, NJ, USA, March 22-24, 2006.
[25] J. Yang, S.-H. S. Huang, and M. D. Wan, A clustering-partitioning algorithm to find TCP packet round-trip time for intrusion detection. *International Conference on Advanced Information Networking and Applications*, Vienna, Austria, April 18-20, 2006.
[26] L. Zhang, A. Persaud, A. Johnson, and Y. Guan, Detection of stepping stone attack under delay and chaff perturbations. *Performance, Computing, and Communications Conference*, Phoenix, Arizona, USA, April 10-12, 2006.
[27] T. He, P. Venkitasubramaniam, and L. Tong, Packet scheduling against stepping-stone attacks with chaff. *IEEE Military Communications Conference*, Washington, DC, USA, September 25-28, 2006.
[28] T. He, and L. Tong, Detecting information flows: Improving chaff tolerance by joint detection. *Conference on Information Sciences and Systems*, Baltimore, MD, USA, March 14-16, 2007.
[29] W. T. Strayer, C. Jones, B. Schwartz, S. Edwards, W. Milliken, and A. Jackson, Efficient multi-dimensional flow correlation. *IEEE*

Conference on Local Computer Networks, Clontarf Castle, Dublin, Ireland, October 15-18, 2007.

[30] W. T. Strayer, C. E. Jones, and B. I. Schwartz, Architecture for multi-stage network attack traceback. *IEEE Conference on Local Computer Networks,* Sydney, Australia, November 15-17, 2005.

[31] B. Coskun, and N. Memon, Efficient detection of delay-constrained relay nodes. *Annual Computer Security Applications Conference*, Miami Beach, Florida, USA, December 10-14, 2007.

[32] J. D. Padhye, and M. Wright, Stepping-stone network attack kit (sneak) for evading timing-based detection methods under the cloak of constant rate multimedia streams. *Doctoral dissertation*, University of Texas at Arlington, 2008

[33] H.-C. Wu, and S.-H. Huang, Performance of neural networks in stepping-stone intrusion detection. *IEEE International Conference on Networking, Sensing and Control*, Hainan, China, April 6-8, 2008.

[34] H.-C. Wu, and S.-H. S. Huang, Neural networks-based detection of stepping-stone intrusion. *Expert Systems with Applications,* 37(2), p1431-1437, 2010.

[35] A. Almulhem, and I. Traore, Detecting connection-chains: A data mining approach. *International Journal of Network Security*, 10(1), p62-74, 2008.

[36] S.-H. Huang, R. Lychev, and J. Yang, Stepping-Stone detection via Request-Response traffic analysis. *Leture Notes in Computer Science*. 276-285, 4610, 2007

[37] Y. J. Pyun, Y. H. Park, X. Wang, D. S. Reeves, and P. Ning, Tracing traffic through intermediate hosts that repacketize flows. *IEEE International Conference on Computer Communications,* Anchorage, Alaska, USA, May 6-12 , 2007.

[38] A. Kampasi, Y. Zhang, G. Di Crescenzo, A. Ghosh, and R.Talpade, Improving Stepping-Stone Detection Algorithms using Anomaly Detection Techniques. *Report TR-07-28 (regular report),* The University of Texas at Austin, 2007.

[39] M. N. Omar, M. A. Maarof, and A, Zainal, Solving time gap problems through the optimization of detecting stepping stone algorithm. *International Conference on Computer and Information Technology,* Wuhan, China, September 14-16, 2004.

[40] M. N. Omar, M. A. M., and Zainal, A., The optimization of stepping stone detection: Packet capture steps. *Jurnal Teknologi,* 44(D), 1-14, 2006.

[41] M. N. Omar, L. Siregar, and R. Budiarto, Dropped packet problems in stepping-stone detection. *International Journal of Computer Science and Network Security*, 8(2), p109-115, 2008.

[42] X. Wang, The loop fallacy and serialization in tracing intrusion connections through stepping stones. *ACM symposium on Applied computing*, Nicosia, Cyprus, March 14-17, 2004.

[43] X. Wang, The loop fallacy and deterministic serialization in tracing intrusion connections through stepping stones. *International Journal of Security and Networks*, 1(3/4), p184-197, 2006.

[44] Y. Tang, Y. Liverpool, and T. E. Daniels, Monitor placement for stepping stone analysis. *International Performance Computing and Communications Conference*, Phoenix, Arizona, USA, April 10-12, 2006.

[45] M. Venkateshaiah, and M. Wright, Evading stepping-stone detection under the cloak of streaming media. *Technical Report*, University of Texas at Arlington, 2007.

[46] M. Venkateshaiah, and M. Wright, Evading existing stepping-stone detection methods using buffering. *Doctoral dissertation*, University of Texas at Arlington, 2007

[47] J. Xin, L. Zhang, B. Aswegan, J. Dickerson, T. Daniels, and Y. Guan, A testbed for evaluation and analysis of stepping stone attack attribution techniques. *International Conference on Testbeds & Research Infrastructures for the Development of Networks & Communities*, Barcelona, Spain, March 1-3, 2006.

[48] A. Chantler, and R. Broadhurst, Social engineering and crime prevention in cyberspace. *Technical report*, Justice, Queensland University of Technology, 2006.

[49] J. Postel, and J. K. Reynolds, Telnet Protocol specification. *RFC 854*, May, 1983.

[50] S. Kent, and R. Atkinson, Security architecture for the Internet Protocol. *RFC 2401*, 1998.

[51] T. Ylonen, The Secure Shell (SSH) Protocol Architecture. *RFC 4251*, 2006.

[52] Z. Trabelsi, W. El-Hajj, and S. Hamdy, Implementation of an ICMP-based covert channel for file and message transfer, *International Conference on Electronics, Circuits, and Systems*, St. Julians, Malta, August 31 – September 3, 2008.

[53] G. Gu, J. Zhang, and W. Lee, BotSniffer: Detecting botnet command and control channels in network traffic. *Annual Network and Distributed System Security Symposium*, San Diego, California, USA, February 10 - 13, 2008.

[54] G. Gu, R. Perdisci, J. Zhang, and W. Lee, BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. *USENIX Security Symposium*, San Jose, CA, USA July 28-August 1, 2008.

[55] W. T. Strayer, R. Walsh, C. Livadas, and D. Lapsley, Detecting botnets with tight command and control. *Annual IEEE Conference on Local Computer Networks,* Tampa, Florida, USA, November 14-16, 2006.