

# Evaluation of passive stepping stone detection techniques under chaff and delay

Michael Shell  
Georgia Institute of Technology  
someemail@somedomain.com

Homer Simpson  
Twentieth Century Fox  
homer@thesimpsons.com

James Kirk  
and Montgomery Scott  
Starfleet Academy  
someemail@somedomain.com

**Abstract—Bla**

## I. INTRODUCTION

Network attackers frequently use a chain of compromised intermediate nodes to attack a target machine and maintain anonymity. This chain of nodes between the attacker and the target is called a stepping stone chain.

Malicious actors on the Internet frequently use chains of compromised hosts to relay their attack, in order to obtain access to restricted resources and to reduce the chance of being detected. These hosts, called **stepping-stones**, are used by the attacker as relay machines, to which they log in using tools such as SSH or telnet [reference](#).

Accessing a server via multiple relayed TCP connections can make it harder to tell the intruder's geographical location, and enables attackers to hide behind a long interactive stepping-stone chain. Furthermore, it is often required to relay an attack via privileged hosts in a network that have access to restricted resources.

However, detecting that a host is used in a stepping stone chain is a clear indication of malicious behaviour. If a stepping-stone intrusion can be detected during the attack stage, the connection can be terminated to interrupt the attack. Stepping-stone detection primarily looks at network traffic, with most approaches aiming to identify potential correlation between two connections going from or to a particular host.

There are a number of approaches to detect stepping-stones, with the earliest one having been proposed by Zhang and Paxson in 2000 [citation needed?](#). However, no

### A. Explain normal setup

The remaining of this paper is organized as follows. In Section 2, we present some significant host-based approaches for SSID. In Section 3, we summarize some typically known network-based (connection-chain based) approaches for SSID. In Section 4, we propose several open problems in this area. Finally, we conclude our paper in Section 5 and provide the funding information of this research work in the declarations section.

### B. Related work

Xin et al. [11] developed a standard test bed for stepping-stone detection, called *SST*. The main objectives in the development of SST was to provide (1) reproducible evaluation of stepping stone chain detection algorithms, (2) scalability and expandability, and (3) easy configuration, management and operation. The tool generates scripts to mimic an interactive SSH and TelNet connections.

To insert time delays and chaff perturbations, the authors modified the OpenSSH proto. They demonstrated this with an experiment comparing two stepping-stone detection approaches,

It does not require special hardware, yet provides features like interactive session generation with

Wang et al. [10] recently conducted an extensive survey of stepping stone intrusion detection. The authors group methods according to the respective methodology into

- content-thumbprint,
- time-thumbprint,
- packet counting,
- random-walk-based,
- cross-over packet-based,
- watermarking,
- network-based,
- and software-defined-networking-based,

but do not cover [graph-based methods](#) such as [4] or anomaly-based methods such as [2], which are increasing in popularity recently. The authors then proceed to explain the different methods and highlight their benefits and shortcomings. The authors discuss open problems, but do not provide a comparison of detection rates.

Shullich et al. [8] in 2011 also conducted a survey on stepping stone intrusion detection. The authors perform a similar grouping of methods, but also discuss related work in evasion tactics and test frameworks. The authors furthermore give an outlook on areas for future research, such as hacker motivation, the cardinality problem when correlating connection pairs, the difficulty of tracing back chains through firewalls, the lack of real-world data examples, or detection in covert channels or protocols such as UDP.

[1]

Stepping Stone Detection Techniques: Classification and State-of-the-Art, bad though

Metrics: A Study on the Performance Metrics for Evaluating Stepping Stone Detection (SSD) Stepping Stone Detection: Measuring the SSD Capability

## II. DATASET CREATION

### A. Simulating stepping stones with SSH-tunnels and Docker

insert figure with one and with three stepping stones

SSH tunnel on respective port on the starting point of the chain, tunnels to port on the next point in the chain. Finally, Refer to figure.

1) *Adding network congestion*: Docker communication takes place over virtual bridge networks, so the throughput is far higher and more reliable than in real-world networks. To retard the quality of the Docker network to realistic levels, we rely on the emulation tools Netem. Netem [add reference](#) is a Linux command line tool that allows users to artificially simulate network conditions such as high latency, low bandwidth or packet corruption in a flexible manner. We apply Netem commands to the network interface of each container, which adds correlated delays to incoming and outgoing packets that are drawn from a normal distribution with mean  $\mu$ , variance  $\sigma^2$ , and correlation  $\rho_1$ . We furthermore apply correlated packet loss and corruption drawn from a binomial distribution with probability  $p$  and correlation  $\rho_2$ .

We set the network settings for the starting point and the end point container individually and draw each of the given parameters from a suitable distribution ([should I specify which one for each? Seems a bit much...](#)) before each [run](#) to allow for a good amount of variation in the generated data.

2) *Adding delays and chaff*: To add artificial delays to forwarded packets on a stepping stone host for detection evasion, we can again use NetEm. We draw delays for departing packets from a uniform distribution, as suggested by [add reference](#), covering the interval  $[0, \delta_d]$ , with no packet correlation.

To add chaff packets to the relayed connection, we forward two additional ports through the SSH-tunnel of a stepping stone host. We then use NetCat [add reference](#) to send data to both ports from either direction and collect it at the other side. Figure ... depicts this setup for an individual tunnel. The data sent through tunnel  $i$  consists of strings with random size  $x$  drawn from a Cauchy-distribution with mean  $xx_i$ , and is sent in intervals of random length  $\delta_c$  drawn from an exponential distribution with mean  $yy_i$ . By adjusting  $yy_i$ , we can control the amount of chaff sent through a tunnel.

### B. Simulating interactive SSH-traffic

In order to generate enough data instances representing interactive stepping stone behaviour, we automatised the communication between the start point and the end point of the stepping stone chain. To do so, we generate a script with SSH-commands at the start of each [execution](#) that is passed and run by the [starting point](#) of the chain. The generated script consists of a sequence of ordinary SSH-commands [list them](#)

[here?](#), which are drawn randomly from a command catalogue and are each separated by *sleep*-commands for a time  $t$  that is drawn each time from a Cauchy-distribution. The average sleep-time is around [insert](#). The length of the script is reached when the *end*-command is drawn from the catalogue. [Insert example](#)

### C. HTTP-interactions

In order to provide an additional, different type of interaction between the [starting point](#) and [end point](#), we directed HTTP traffic over the stepping stone chain. Here, the starting point hosts Scrapy, a web crawling service [insert citation](#), that surfs the 1 million most popular website by clicking links on them. The requests are sent over the stepping stone chain to the web.

This type of traffic is not meant to necessarily represent realistic stepping stone behaviour, but to provide an additional source of interactive traffic that differs substantially from SSH in order to test detection methods from another angle.

SSH 1 node	no pert.	var. delays	var. chaff	delay&chaff
HTTP 1 node	no pert.	var. delays	var. chaff	delay&chaff
SSH 3 node	no pert.	var. delays	var. chaff	delay&chaff
HTTP 1 node	no pert.	var. delays	var. chaff	delay&chaff

[9]

## III. SELECTED APPROACHES

### A. Packet-correlation-based approaches

Efficient multi-dimensional flow correlation

Detecting Connection-Chains: A Data Mining Approach

Correlating TCP/IP Packet contexts to detectstepping-stone intrusion 2011

### B. RTT-based approaches

Another prominent approach to detect stepping stones is based on *Round-trip/times* (RTTs). The RTT of a connection is the time it takes for a packet to be sent to the receiver plus the time it takes for an acknowledgement of that packet to be received. For a normal connection, the measured RTTs should be centered closely around one value. However, since information is relayed over one or more hosts in a stepping stone chain, [the assumption for RTT-based detection](#) is that the responses from different hosts within the chain generate multiple RTTs. Observing multiple RTTs is therefore a clear indication of relaying behaviour.

Yang et al. [13], [12] and Huang et al. [6], [3], [7] both have proposed multiple approaches for estimating and employing RTTs for stepping stone detection. We have selected two papers that depict the [state-of-the-art...](#)

1) *RTT-based Random Walk Approach to Detect Stepping-Stone Intrusion* [13]: The model combines packet-counting methods and RTT mining methods to improve detection results from [12].

A widely-used approach is to compare the number of incoming packets in one connection with the number of outgoing packets in another connection to determine if the

pair represents a stepping stone relay. However, the insertion of chaff can **separate** these numbers substantially. To resist intruders evasion, the authors propose to use the number of round-trips in a connection to determine if the connection is being relayed. Packet pairs representing a round-trip for each connection are estimated using a combination of packet matching and clustering, and counted as  $N_{in}$  and  $N_{out}$ . The authors then claim that the value of  $N_{in} - N_{out}$  is only bounded if the two connections are relayed.

2) *Detecting Stepping-Stone Intruders by Identifying Crossover Packets in SSH Connections* [6]: This method improves the detection methods proposed by Ding et al. [3]. The authors target specifically relayed interactive SSH communication at the end of a connection chain. They build their detection model on the fact that in a long connection chain, the round-trip time of a packet may be longer than the intervals between two consecutive keystrokes. Normally after sending a request packet, a client will wait for the server response before sending another request. However, TCP/IP allows a client to send a limited number of packets to the server without having to wait for the response. In a long connection chain, this will result in cross-overs between request and response, which causes the curve of sorted Upstream RTTs to rise more steeply than in a regular connection. A stepping stone is detected if the maximum increase in the curve exceeds a threshold. The authors do not state a universal threshold value and instead suggest a method to estimate the appropriate value for a given setting.

### C. Anomaly-based approaches

Since the insertion of time delays and chaff perturbations is so far relatively successful at evading detection, two authors have proposed algorithms to detect these two activities in a connection as deviations from typical TCP-behaviour to indicate of suspicious behaviour.

1) *Detecting Anomalies in Active Insider Stepping Stone Attacks*: Crescenzo et al. [2] have proposed a **assembly** of three methods to detect time delays and chaff perturbations in a selected connection.

- 1) The response-time based method is targeted at detecting packet time-delays. It estimates the RTT of a connection, and then flags acknowledgement packets if their response-time exceeds  $(RTT + \delta_{RT})$ . The authors claim that an attacker would have to impose delays on more than 70% of all packets to evade this method.
- 2) The edit-distance based method is targeted at detecting

2) *Detecting Chaff Perturbation on Stepping-Stone Connection*: Huang et al. [5] proposed a different approach to detect chaff perturbations in stepping-stone connections. The underlying hypothesis is that interarrival times in benign connections follow distributions like the Pareto or Lognormal, which is not the case for chaffed connections. To identify if a connection is subject to chaff perturbations, the authors fit a distribution using maximum likelihood estimation. Afterwards, two statistical tests assess how well the connection follows the fitted distribution and produce a disagreement score. The produced score is then compared to a threshold estimated from

a set of scores from benign connections. The authors are able to detect chaff without false positives with 95% for a chaff ratio of 50%, and with 100% at a chaff ratio of 100% or higher. For lower chaff ratios, the detection scores decrease rapidly.

### D. Neural networks

Similar to other areas in intrusion detection, researchers have recently begun to train artificial neural networks to identify stepping stones. A notable initial example came from **citation**, who ... Performance of neural networks in stepping-stone intrusion detection 2008 however, the authors concluded that the achieved results did not improve existing detection rates, but no good results, better in Neural networks-based detection of stepping-stone intrusion

A more recent example comes from **citaton**, who train a convolutional neural network to identify correlation between two simultaneous connections from the upstream and downstream interarrival times and packet sizes in each connection. The trained network is large, with over 200 input filters and consists of three convolutional and three feed-forward layers. The trained model was initially applied to a dataset of Tor-connections as well, where the authors achieved strong results. The authors achieve a 90% detection rate with 0.02% false positives.

### E. Behaviour-based approaches

A different direction in the detection of stepping stones focuses more on the general communication behaviour of selected hosts rather than individual connections. Features include the timely correlation of connecting IP-address on a selected host or unusual paths of simultaneously existing connections within a computer network.

Emulating these features realistically in network traffic datasets is difficult, and there exist little research on how strongly actual attack behaviour influences these features. Since our dataset focuses only on individual connections and corresponding evasion, **we are not able to include behaviour-based approaches in our evaluation.**

### F. Evaluation methods

All of the above presented methods classify individual connections or pairs of connections as malicious or benign. True stepping stone connections are rare compared to benign ones, making their detection an imbalanced classification problem. As discussed by **insert citation**, an appropriate measure to evaluate SSD methods are false positive and false negative rates as well as the *Area-under-ROC-curve* (AUC) for threshold-based methods.

The false positive rate is defined as

$$\frac{\text{number of benign connections classified as malicious}}{\text{overall number of benign connections}},$$

while the false negative rate is defined as

$$\frac{\text{number of malicious connections classified as benign}}{\text{overall number of malicious connections}}.$$

## IV. RESULTS

### A. Unperturbed data

#### REFERENCES

- [1] A. Almulhem and I. Traore, "A survey of connection-chains detection techniques," in *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. IEEE, 2007, pp. 219–222.
- [2] G. Di Crescenzo, A. Ghosh, A. Kampasi, R. Talpade, and Y. Zhang, "Detecting anomalies in active insider stepping stone attacks." *JoWUA*, vol. 2, no. 1, pp. 103–120, 2011.
- [3] W. Ding, M. J. Hausknecht, S.-H. S. Huang, and Z. Riggle, "Detecting stepping-stone intruders with long connection chains," in *2009 Fifth International Conference on Information Assurance and Security*, vol. 2. IEEE, 2009, pp. 665–669.
- [4] M. Gamarra, S. Shetty, D. M. Nicol, O. Gonazlez, C. A. Kamhoua, and L. Njilla, "Analysis of stepping stone attacks in dynamic vulnerability graphs," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–7.
- [5] S.-H. S. Huang and Y.-W. Kuo, "Detecting chaff perturbation on stepping-stone connection," in *2011 IEEE 17th International Conference on Parallel and Distributed Systems*. IEEE, 2011, pp. 660–667.
- [6] S.-H. S. Huang, H. Zhang, and M. Phay, "Detecting stepping-stone intruders by identifying crossover packets in ssh connections," in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*. IEEE, 2016, pp. 1043–1050.
- [7] S.-H. S. Huang, R. Lychev, and J. Yang, "Stepping-stone detection via request-response traffic analysis," in *International Conference on Autonomic and Trusted Computing*. Springer, 2007, pp. 276–285.
- [8] R. Shullich, J. Chu, P. Ji, and W. Chen, "A survey of research in stepping-stone detection," " *International Journal of Electronic Commerce Studies*", vol. 2, no. 2, pp. 103–126, 2011.
- [9] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *2010 IEEE Symposium on Security and Privacy*, May 2010, pp. 305–316.
- [10] L. Wang and J. Yang, "A research survey in stepping-stone intrusion detection," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 276, 2018.
- [11] 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," in *2nd International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006*. IEEE, 2006, pp. 9–pp.
- [12] J. Yang and S.-H. S. Huang, "Mining tcp/ip packets to detect stepping-stone intrusion," *computers & security*, vol. 26, no. 7-8, pp. 479–484, 2007.
- [13] J. Yang and Y. Zhang, "Rtt-based random walk approach to detect stepping-stone intrusion," in *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*. IEEE, 2015, pp. 558–563.

#### APPENDIX