Evaluation of passiv 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 targetmachine and maintain anonymity. This chain of nodes between the attacker and the target is calleda 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

Stepping-stone intruders can make a connection chain-shown as in Fig.1 using telnet/rlogin/ssh to launch theirattacks. In Fig.1, we assume that Host 0 is used by an in-truder to launch an attack to Host N via compromisedhosts Host 1, Host 2,...,Hosti1, Hosti,Hosti+1,...,and HostN1. SSID can occur at one of thestepping-stones. It is assumed that the detection programresides in Hostiwhich is called a (detecting) sensor. SSIDis to determine whether the sensor Hostiis used as astepping-stone. The connection from Hosti1toHostiis called an incoming connection to Hosti, and the con-nection from Hostito Hosti+ 1 is called an outgoingconnection from Hosti. A necessary condition that Hostiis used as a stepping-stone is that there is at least onerelayed pair between all the incoming connections and allthe outgoing connections.One type of approach to detect stepping-stone intru-sion is to compare all the incoming connections with allthe outgoing connections of the same host to see if thereexists a relayed pair. This type of approach is calledhos-t-basedSSID. We will discuss all the significant researchwork for host-based SSID in Section2. The primaryissue of this type of approach is that high false-positiveerrors can be easily introduced since some legal applica-tions may use stepping-stones to access remote servers.Another type of approach to overcome the issues ofhost-based detection is to estimate the number of connections from Host 0 to HostN(as shown in Fig.1),which is referred to as the length of the connectionchain. If there are more than three connections involvedin a connection chain, it indicates that the user obvi-ously tries to access HostNvia more than three com-puter hosts. Clearly, the more hosts involved in aninteractive session to access a server, the slower the net-work communication, unless there are something hidden; otherwise, it does not make sense to access aremote sever via more than three hosts. The number"three"is used because it was found that most legal ap-plications rarely used more than three stepping-stonesto access a remote server. This type of approach is callednetwork-based(orconnection-chain based) SSID.Estimating the length of the connection chain fromHost 0 to Hostias shown in Fig.1 is called upstreamdetection. Similarly, estimating the length of the connec-tion chain form Hostito HostNis called downstreamdetection. Accurate estimation of the length of the wholeconnection chain requires both downstream and up-stream detections. Unfortunately, performing upstreamdetection is extremely challenging. Such a problem is still open and remains unsolved. Therefore, it is ex-tremely hard to estimate the length of the whole connec-tion chain. So most researchers in SSID primarilyfocused on using the length of downstream connectionto decide whether there is a stepping-stone intrusion.Compared to using the length of the whole connectionchain, this simplified method may introduce false nega-tive errors, but it performs much better than host-baseddetection as well as largely reduces false positive errors.

The remaining of this paper is organized as follows. In-Section2, we present some significant host-based ap-proaches for SSID. In Section3, we summarize sometypically known network-based (connection-chain based)approaches for SSID. In Section4, we propose severalopen problems in this area.

Network and Distributed Systems Security (NDSS) Symposium 2020
23-26 February 2020, San Diego, CA, USA
ISBN 1-891562-61-4
https://dx.doi.org/10.14722/ndss.2020.23xxx
www.ndss-symposium.org

Finally, we conclude ourpaper in Section5 and provide the funding informationof this research work in the declarations section.

B. Related work

Wang et al. 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 [**?**] or [**?**], 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.

[**?**]

[**?**]

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

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

II. 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. [**?**], [**?**] and Huang et al. [**?**], [**?**], [**?**] 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 [**?**]:* The model combines packet-counting methods and RTT mining methods to improve detection results from [**?**].

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 insertation 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 [**?**]:* This method improves the detection methods proposed by Ding et al. [**?**]. 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 connectin 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

Crescenzo et al. [**?**]

D. Detecting Anomalies in Active Insider Stepping Stone Attacks

Huang et al. [**?**].

E. Neural networks

Performance of neural networks in stepping-stone intrusion detection 2008 but no good results, better in Neural networks-based detection of stepping-stone intrusion

III. 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 μ, variance σ^2, and correlation ρ_1. We furthermore apply correlated packet loss and corruption drawn from a binomial distribution with probability p and correlation ρ_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 artifical 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 δ_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

[**?**]

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. Apruzzese, F. Pierazzi, M. Colajanni, and M. Marchetti, "Detection and threat prioritization of pivoting attacks in large networks," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[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