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

Rahul Kumar and B.B. Gupta

Abstract Today, the most common way to perform various attacks is to use stepping stone hosts in the attacking path. In stepping stone attacks, attacker creates a long chain of connections via intermediary previously compromised nodes, to execute attack. The only way to break this chain is to detect stepping stones and applying some security constraints on the traffic flowing through them, not to allow malicious traffic through them. In this paper, we present classification and state-of-the-art of existing schemes proposed for stepping stone detection in recent past. Moreover, we compare these techniques based on their merits and demerits, and discuss open issues and challenges that can be used for further research in this domain.

Keywords Stepping stone host · Stepping stone connection chain · Stepping stone connection pair · Stepping stone intrusion path

Introduction

Internet plays a vital role in today's daily life and business. Online availability of various services like banking, shopping, software and hardware services, social networking etc., has number of benefits such as time saving, better customer services and experience, storing and sharing information, etc. Due to these benefits, a mammoth part of population on earth is now getting connected to it. But Internet also has some security holes which creates serious security issue to its user. Lack of security in any of the services can cause a loss of lots of money and most importantly it may cause loss of secret information of any organization, country, or of an individual. Therefore, proper security of these services is required to protect

Rahul Kumar · B.B. Gupta (✉)
National Institute of Technology Kurukshetra, Kurukshetra, India
e-mail: gupta.brij@gmail.com

Rahul Kumar
e-mail: Rahulkumarbit72@gmail.com

N. Afzalpulkar et al. (eds.), *Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing*,
DOI 10.1007/978-81-322-2638-3_59

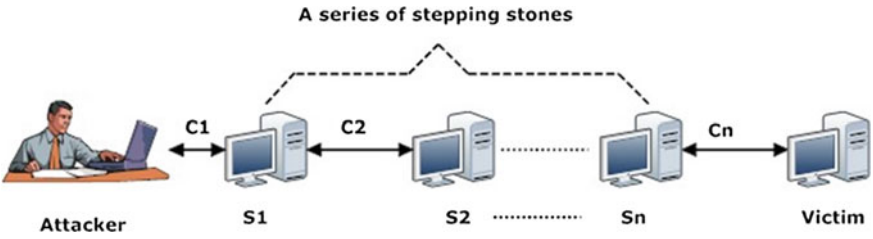

Fig. 1 Stepping stone attacks model

these services from attackers. But attackers are more dominating and they never give up; every time researcher come up with new solution, attackers find new evasion techniques [1, 2].

In most of the cases, attacker uses stepping stone hosts to execute attacking commands instead of attacking from his/her personal computer machine, due to which attacker remains unidentified [3]. Attacker uses various scanners to find out vulnerable computer machines over Internet and then exploit the vulnerabilities found over these machines to compromise it. Moreover, attacker can use these compromised machines to find some more vulnerable machines over Internet to compromise and so on. In this way attacker is able to create chain of connections through compromised hosts/machines, which also known as stepping stone hosts, and use these compromised hosts to execute attacking command [4]. Due to the property of TCP/IP protocol, packets arriving at victim host will contain IP address of last stepping stone host in the chain, therefore last stepping stone host in connection chain appears as an attacker [5].

Stepping stone attacking model is shown in Fig. 1. In this attacking model, there are n stepping stone hosts namely S_1, S_2,…S_n. The series of connections $<C_1, C_2, \ldots C_n>$ is called stepping stone connection chain and any pair of these connections is called stepping stone connection pair. For example, (C_1, C_2) is a stepping stone connection pair [3].

There is flexibility in the way stepping stones can be used to execute attacks due to which different kinds of attacks can be executed such as denial-of-service attacks [6], creating backdoors, dictionary attacks, spreading virus and worms, etc. [4]. Stepping stone attacks are not limited to national boundaries, as in stepping stone attacks, some stepping stone hosts may present in different countries. Presence of stepping stone hosts in different countries can make tracing of attacker more difficult as different countries can have its own cyber laws [4]. In 1995, Staniford-Chen and Heberlein [7] first proposed a stepping stone detection approach based on the content of attacking packets which is also called "thumbprint approach." However, this approach is unable to detect stepping stone hosts when the attack traffic is encrypted.

Different types of techniques have been proposed by various researchers time to time for stepping stone detection. However, every time researchers come up with new solution, attackers try to find some evasion techniques to escape the detection system. Chaff and Timing perturbation, reshuffling, encryption, etc., are commonly used techniques by the attacker to escape detection system. There are certain characteristic of TCP/IP traffic such as packet size, packet timestamp, etc. which can help to detect stepping stone host [2]. In this paper, we present classification and state-of-the-art of existing schemes proposed for stepping stone detection in recent past and compare these techniques based on their merits and demerits. In addition, we discuss current issues and challenges in detection of stepping stone hosts that can be used for further research in this domain.

The rest of this paper is organized as follows. Section "Problem of Stepping Stones" discusses about stepping stone problem. In section "Stepping Stone detection Techniques," we discuss some excellent stepping stone detection techniques, section "Open Issues and Challenges" discusses issues and challenges in the detection of stepping stones. Finally, section "Conclusion and Future Work" concludes the paper and discusses scope for future work.

Problem of Stepping Stones

One major application of computer machines is to store useful and secret information so that they can be accessed at later time. Development of Internet technology allows people to share information easily, access services online, remotely login to other computer, and so on. Attackers generally use Telnet, Open SSH, etc., to remotely login to other computer systems. To remain unidentified attackers execute attacking command through intermediate compromised rather from their personal computer. To make it possible attacker uses remote login tools such rlogin and open SSH to remotely login into vulnerable computers and take over the control and use these systems to remotely login into other and take control of them and so on. In this way attacker creates a long connection chain through intermediate compromised hosts, and launch attacking command from last host in the chain. The intermediate compromised hosts are called stepping stone hosts. Once attack is able to take control over a machine as stepping stone host, he/she can use it to compromise and control other remote computer machines and so on. In stepping stone attacks, origin of attack cannot be detected without detecting stepping stone hosts. The difference between cyber laws of different countries also helps attackers to escape detection processor making detection more difficult, because a country may consider an activity as an offensive while other may not. Due to which attacker may create and use stepping stone in different countries, in that case detecting origin of attack is more difficult.

Stepping Stone Detection Techniques

Various stepping stone detection techniques have been proposed by the researchers. Some of them find out intrusion path created by the attacker while other detects stepping stone by comparing incoming and outgoing connections of host. The efficiency and quality of stepping stone detection techniques can be evaluated by calculating both false negative rate and false positive rate. False positive means a normal host is detected as stepping stone by the algorithm, while false negative means a stepping stone host is left undetected by the algorithm. A high false negative rate is more serious and unacceptable than a high false positive rate. The amount of chaff that an algorithm can handle also measures the effectiveness of the algorithm. Chaff packets are the dummy packets that attackers can insert into connection to escape detection process. Chaff packets need not to be arrived at victim. It may be dropped at an intermediate host. If an algorithm can detect stepping stone host even when attacker has inserted lots of chaff, which is more effective than other approaches that cannot do the same.

Correlation Techniques

Zhang and Paxon [3] proposed an ON/OFF correlation approach. ON/OFF approach correlates incoming and outgoing connections of a host by correlating ON and OFF periods of the connections. A connection is in OFF period if there is no data in connection for more than a specified period of time say T_{ideal}. ON period of a connection begins, when a packet with new payload enters into connection. Suppose C_{in} is an incoming and C_{out} is an outgoing connection on computer host H. C_{in} and C_{out} will be correlated if their OFF periods end at similar time. This will indicate that host H is a stepping stone host and C_{in}, C_{out} is called stepping stone connection pair. This is also true that if two connections form stepping stone pair, they will leave their OFF period at similar times. Using these two definitions, following mathematical expression can be derived:

Suppose $T1$ and $T2$ are the time when two OFF periods OFF1 and OFF2 end, respectively. OFF_n1 and OFF_n2 are OFF periods that occur in C_{in} and C_{out}, respectively, and OFF_{12} is the number of correlated OFF period. Here δ and γ are two control parameters.

- OFF1 and OFF2 are said to be correlated if $T2 - T1 \leq \delta$.
- C_{in} and C_{out} is a pair stepping stone connection, if $\frac{\text{OFF}_{12}}{\min(\text{OFF}_{n1},\text{OFF}_{n2})} \geq \gamma$.

Refinement of this approach is done using the concept of *casuality*, if two connections C_{in} and C_{out} are part of connection chain and OFF period of C_{in} end before C_{out}, then OFF period of C_{in} will always end first among the OFF periods of C_{in} and C_{out}. This observation helps to remove unwanted connection pair. Another

improvement in this approach can be done by considering consecutive coincidences, because consecutive coincidence occur more for stepping stone connection pair as compared to normal connection pair. Using this definition, C_{in} and C_{out} form a stepping stone pair if $\text{OFF}^{*}_{1,2} \geq \min_{\text{CSC}}$ and $\frac{\text{OFF}^{*}_{1,2}}{\min(\text{OFF}_{n1},\text{OFF}_{n2})} \geq \gamma'.\text{OFF}^{*}_{1,2}$ is the number of consecutive coincidence. $\min_{\text{CSC}}$ is the minimum number of consecutive coincidence and γ' is new control parameter.

Stepping Stone Detection with Encrypted Attacking Traffic

Ting He et al. [8] proposed a stepping stone detection technique which can detect stepping stones in the presence of encrypted attacking traffic. While performing stepping stone attacks, attacker has to face two constraints. First is the *Bounded memory constraints*, which state that there is limit on the amount of memory that attacker can use on stepping stone host. Second is *Bounded delay constraints*, which state that attacker cannot delay a packet more than specified amount of time.

To detect stepping stone with bounded memory, consider that attacker is allowed to use memory for only M packet. Suppose S_{in} and S_{out} are incoming and outgoing streams of host H, if maximum variation between these streams is less than or equal to M then S_{in} and S_{out} is called stepping stone connection pair with bounded memory M otherwise (S_{in}, S_{out}) is called normal pair.

Suppose $N_{\text{in}}(w)$ and $N_{\text{out}}(w)$ represent the number of packets observed in S_{in} and S_{out}, respectively, where w is the total number of packets observed. Suppose diff(w) represents the packet difference between S_{in} and S_{out}, then diff(w) can be calculated as given below: diff$(w) = N_{\text{in}}(w) - N_{\text{out}}(w)$. Similarly, maximum variation can be defined as $\text{var}(w) = \max_{1 \leq i \leq w} \text{diff}(i) - \min_{1 \leq i \leq w} \text{diff}(i)$. Using definition of maximum variation stepping stone detector with bounded memory is given by

$$\delta_{\text{DMV}}(S_{\text{in}}, S_{\text{out}}, \text{M}, \text{n}) = \begin{cases} 1 \text{ if } \text{var}(n) \leq M \\ 0 \quad \text{otherwise} \end{cases} \tag{1}$$

Here 1 (one) indicates that S_{in}, S_{out} is a stepping stone pair, and 0 (zero) indicates that S_{in}, S_{out} is a normal pair.

Delay constraints approach assumes a value Δ which represents the maximum value of delay in the delivery of a packet. A pair of stream (S_{in}, S_{out}) is a stepping stone pair with bounded delay Δ, if for each packet in S_{in} there exist a corresponding packet in S_{out} subjected to maximum bounded delay Δ.

To detect stepping stone pair with bounded delay a *detect match (DM)* algorithm was given by in [8]. *DM* algorithm searches valid pair for each packet p in S_{in}. Pair (p, q) is valid pair where $p \in S_{\text{in}}$ and $q \in S_{\text{out}}$. Time stamp difference between p and q is $t(q) - t(p) \leq \Delta$, where $t(q)$ and $t(p)$ are the timestamps of q and p, respectively. If *DM* algorithm found a valid pair for each incoming packet then *DM* indicates that host is

stepping stone otherwise it is a normal host, *DM* performs same operation for all incoming connection. *detect match* has exponential time complexity but it searches for order preserving mapping, an order preserving mapping reduces miss detection.

To detect stepping stone with bounded memory in the presence of chaff, they proposed *detect bounded memory chaff (DBMC)* algorithm. *DBMC* algorithm uses a counter *C*, to count number of times the memory goes *underflow* and *overflow*. If $C/n < (1/(M + 1))$ then *DBMC* returns attack otherwise normal. Here n is the number of packet observed. *DBMC* can handle $1/(M + 1)$ amount of *chaff*, over which detection quality will decrease. *DBDC* algorithm counts the number of *chaff* packet using counter *C*, and if $C/n < (1/(1 + \lambda\Delta))$ algorithm returns attack otherwise returns normal. *DBDC* can handle $1/(1 + \lambda\Delta)$ amount of *chaff*. λ is decision parameter and Δ is the maximum tolerable delay.

Detecting Intrusion Path Using Data Mining Techniques

Data mining approach was proposed by Yang et al. [9], which focuses on detection of origin of attack. They gave a *clustering–partitioning (C-P) algorithm*, which uses *maximum–minimum distance clustering algorithm*. clustering–partitioning algorithm matches *send* and *echo* packet globally which results in correct value of RTT and estimates the length of intrusion path correctly. C-P algorithm captures all send and echo packets for certain period of time. C-P algorithm one by one computes timestamp difference among a send packet p and all echo packets that arrive after p. This ensures that the correct RTT for P is one of these differences. RTT of TCP/IP send packet is given by $\mathrm{RTT}(t) = \mathrm{RTT}_0 + \Delta\mathrm{RTT}(t)$, where RTT_0 represents fixed delay and $\Delta\mathrm{RTT}(t)$ represents variable delay. The idea behind this approach is that RTT's of connection chain with same length will form a single cluster.

To prepare RTT dataset, C-P algorithm captures send and eco packet in certain period of time, and computes timestamp difference between each send packet P and all echo packets after P. In first step, C-P algorithm applies *maximum–minimum distance clustering algorithm* on RTT data set. In second step, C-P algorithm removes all duplicate elements from each cluster. Third step of C-P algorithm is to measure the likelihood of each cluster to check whether it can represent a RTT level or not. In Steps 4 and 5 *C-P* algorithm searches for set of clusters where each cluster represents an RTT level, for that algorithm searches for clusters having higher ratio than other. Ratio of cluster *R* is considered to be higher if $2\sigma > \mu$, where μ and σ are mean and standard deviation of *R*, respectively. If all send packets can be partitioned into a set of clusters with no send packet command between them, such set of clusters is called true cluster set. A set of cluster satisfying these two properties represent the true RTT levels, number of cluster is equal to the number of connection in the chain.

Packet Context Approach for Stepping Stone Detection

Yang et al. [10] proposed a packet context approach to detect stepping stones. In this approach they correlate packet context of TCP/IP packets using *Perason product moment correlation coefficent* to find which incoming connection is correlated to which outgoing connection. Packet context based approaches compute context distance between packet context of each packet P in incoming connection and packet context of all packets in outgoing connection. A host is a stepping stone if any two connections of it are relayed. Packet context approach computes context distance between packet context of each packet P in incoming connection and packet context of all packets in outgoing connection. This computation results into a context distance *set* $D = \{d_1, d_2, d_{3\ldots} d_m\}$. Packet corresponding to outlier d_i $(|d_i - \mu| < 2\delta)$ represents matched packet for packet P, where δ and μ are the standard deviation and mean of D, respectively. Context distance between two contexts X and Y is given by $d_i = 1 - P_{X,Y}$, where $P_{X,Y}$ is Pearson product moment correlation coefficient between packet context X and Y and is given by

$$P_{X,Y} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}} \tag{2}$$

To find relayed connections, authors have used packet context approach to compute a relay degree set $R = \{r_1, r_2, r_3, \ldots r_l\}$ for each incoming connection C_{in}, where r_i is relay degree between an incoming and an outgoing connection, C_{in} and $C_{i,}$ respectively. The connection corresponding to outlier r_i $(|r_i - \mu| > 2\delta)$ of set R represents the relayed connection of incoming connection C_{in}. The same process can be applied for each incoming connection to find its relay.

Neural Network Approach for Stepping Stone Detection

A neural network approach was given by Wu et al. [11] to detect length of intrusion path, which is based on RTT's of TCP/IP send packet. This approach is called *RTT group approach* because RTT dataset is divided into groups and each group is applied as input to the input layer of neural network. A monitoring and capturing system is placed on the host next to the attacker. Packets are captured from the time when there is one connection in the chain to the time when complete stepping stone chain is established. RTT of each send packet is computed by timestamp difference between send and echo packets. Training of neural network is needed before they engaged for function, same is here and then incoming packet can be used as testing data, due to which we do not require to observe connection chain continuously. This RTT group scheme consists of three preprocessing steps, first step is to capture

send and echo packet only, second step is to run a matching algorithm to compute RTT of send packets, third step is to build neural network using RTT dataset created in previous step. To detect number of host in the connection, one can simply plot output of neural networks, where X-coordinate represents RTT group number and *Y*-coordinate represents number of host. The number of steps in the graph represents the number of connection in the chain.

Applying Stepping Stone Approach for Network Threat Detection

Omar et al. [12] use stepping stone perspective to detect various network threats like spam, proxy attacks, DoS attacks, backdoor attack. Stepping stone approach for spam detection is based on monitoring of incoming and outgoing email ports. There are three main ports *SMTP port 25, IMAP port 143, and POP3 port 110* required to monitor for spam detection. In spam attacks a host is used as medium to send email to multiple receiving host, thus an email is sent to host which in turn forwarded to multiple receivers. Therefore, number of emails that a host receives is always lesser than it sends to other, thus this can be mathemetically formulated as follows:

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

Stepping stone approach to detect proxy server compares incoming connection with outgoing connections of a host, if any incoming connection is equal to any outgoing connection and vice versa, this indicates that host is acting as a proxy server. Mathematical expression of proxy detector is given by

$$\mathrm{PROXY}_{\mathrm{SSD}} = \begin{cases} 1 \text{ if } n_{\mathrm{in}} = n_{\mathrm{out}} \\ 0 \text{ if } n_{\mathrm{in}} \neq n_{\mathrm{out}} \end{cases} \tag{4}$$

Backdoor creates an unauthenticated user access to any normal computer machine. Backdoor programs work in background without the knowledge of actual user of computer system. They may be an installed program or may be associated with some virus or worms. This approach detects backdoor without using their signatures. This approach find out whether a host is sending data to outside world using same port for same period of time again and again or not, if yes means backdoor exists in system otherwise not.

Stepping stone approach for DoS detection involves comparing the number of incoming and outgoing connections. If the number of incoming connections is less than number of outgoing connections in a host, then it indicates that the host is a victim of DoS attacks. Using this definition DoS detector can be define as

$$\mathrm{DoS}_{\mathrm{SSD}} = \begin{cases} 1 \text{ if } n_{\mathrm{in}} < n_{\mathrm{out}} \text{ for all } n \\ 0 \text{ if } n_{\mathrm{in}} \neq n_{\mathrm{out}} \text{ for all } n \end{cases} \quad (5)$$

Hybrid Stepping Stone Detection

Omar et al. [13] proposed a hybrid approach for stepping stone detection. Hybrid approach is a combination of two different types of approaches, the host-based approach and network-based approach. Intrusion detection system is an integral part of architecture of hybrid stepping stone detection system, which detects intrusion whenever occurs and raises alarm. Network-based stepping stone detection system starts working and captures network traffic within its boundary. After that it identifies a unique feature from packets captured. The unique feature is then used to detect stepping stone hosts. Network-based stepping stone detection results in a list of stepping stone hosts.

After that host-based stepping stone detection system comes into picture and uses list produced by network-based stepping stone detection system. Each host in list runs its own host-based stepping stone detection. Moreover, each successful host-based detection is listed in a host-based detection list. Host-based detection list is then compared with the list produced by network-based stepping stone detection, if both lists contain same hosts then this indicates that stepping stone host exists in the network.

Open Issues and Challenges

Researcher has proposed many stepping stone detection techniques, but still there are some open issues which researchers can exploit for further development of efficient stepping stone detection techniques. Hybrid approaches are the combination of network-based and host-based stepping stone detection approaches. It combines the advantages of both types of approaches and removes their problem which makes it more efficient but it is more complex in nature. As shown in Table 1, most of the host-based stepping stone detection approaches have some limitations. Scheme proposed in [10] is packet context approach and works correctly only with large number of incoming and outgoing connections. Researcher has used stepping stone perspective for detecting network threats but not supporting it by implementation. As hybrid approach have high detection rate and there are very few hybrid approaches, thus there is need and scope for the development of hybrid approaches. The stepping stone detection approaches which find out the length of intrusion path assumes sensor to be placed on next host to attacker. However, it is difficult to detect a host which is placed next to the attacker.

Table 1 Comparison between various stepping stone detection techniques

Approach	Strength	Weaknesses
ON/OFF (Y. Zhang et al. 2000) [3]	−Detect stepping stones with encrypted attacking traffic −Resistance to evasion	−Chaff packet, Timing perturbation, high false positive rate −Failed to anticipate legitimate stepping stones
Encrypted stepping stone detection (T. He et al. 2007) [8]	−Low false alarm probability −No miss detection using detect match algorithm	−Can handle limited amount of chaff only
Data mining (J. Yang et al. 2007) [9]	−Higher matching rate results in correct RTT dataset. −High quality matching due to global matching	−Higher time complexity −Require continuous monitoring of connection chain
Neural network (H. Wu et al. 2008) [11]	−Do not require continuous monitoring of the connections −Fast regeneration of neural network	−Neural network has to be regenerated for different datasets
Correlating TC/IP packet context (J. Yang et al. 2011) [10]	−No false detection in case of chaff perturbation	−High false negative rate −Require large number of connection
Threat detection using stepping stone perspective (Omar et al. 2013) [12]	−Spam and proxy detection is faster and do not require any signature of backdoor to detect it	−Incomplete and nonreal-time approach
Hybrid approach (Omar et al. 2008) [13]	−Low false negative, low false positive rate, high accuracy	−Undefined behavior when both network- and host-based list are totally different

Therefore, there is a need to design and develop an efficient scheme which can find host next to attacker which can make the talk easier to detect origin of attack.

There are various factors which are required to be considered while evaluating a stepping stone detection approach. These factors include *false negative rate, false positive rate of algorithm,* and amount of *time delay, chaff packets* that algorithm can handle. A high false negative rate is undesirable because it lowers the detection rate while false positive rate is tolerable to some extent. Stepping stone detection algorithm must be capable of handling high amount of chaff. It must also be capable of handling time delays.

Conclusion and Future Work

In this paper, we discussed various stepping stone detection techniques proposed in recent past. Some of these approaches compare incoming and outgoing connections on a host to test whether that host is stepping stone host or not. Some techniques

estimate the length of intrusion path created by attacker by which they can detect all stepping stone hosts in single attempt. Most of the comparison-based schemes are vulnerable to time delays, chaff perturbation, and have high false positive rate. Stepping stone detection schemes estimate that length of intrusion path, mostly estimate downstream length from sensor to victim and do not consider distance between attackers to sensor due to which they find incorrect length of stepping stone path. In addition, we have also discussed a proposed hybrid approach which has high accuracy but there is some undefined behavior in this approach. Therefore, there is need of a real-time stepping stone detection approach. In future, we will work to design and develop a stepping stone intrusion path detection algorithm which can detect correct length of intrusion path.

References

1. Srivastava, A., Gupta, B.B., Tyagi, A., Sharma, A., Mishra, A.: A recent survey on DDoS attacks and defense mechanisms. In: Book on Advances in Parallel Distributed Computing, pp. 570-580. Springer (2011)
2. He, T., et al.: Packet scheduling against stepping-stone attacks with chaff. In: the proceeding of 25th IEEE Military Communications Conference (MILCOM), pp. 1–7 (2006)
3. Zhang, Y., Paxson, V.: Detecting stepping-stones. In: Proceedings of the 9th USENIX Security Symposium, pp. 67–81. Denver, CO, Aug 2000
4. Hsiao, H., Fan, W.-C.: Detecting step stone with network traffic mining approach. In: the proceeding of 4th IEEE Conference on Innovative Computing, Information and Control (ICICIC), pp. 1176—1179 (2009)
5. He, T., Tong, L.: Signal processing perspective to stepping stone detection. In: proceeding of 40th IEEE Annual Conference on Information Sciences and Systems, (CISS), pp. 687–692 (2006)
6. Gupta, B.B., Joshi, R.C., Misra, M.: Defending against distributed denial of service attacks: issues and challenges. Inf. Secur. J. Global Perspect. **18**(5), 224–247 (2009)
7. Staniford-Chen, S., Heberlein, L.T.: Holding intruders accountable on the internet. In: Proceedings of IEEE Symposium on Security and Privacy, pp. 39-49. Oakland, CA (1995)
8. He, T., Tong, L.: Detecting encrypted stepping-stone connections. IEEE Trans. signal process. **55**(5), 1612–1623 (2007)
9. Yang, J., Huang, S.-H.S.: Mining TCP/IP packets to detect stepping stone intrusion. Comput. Secur. **26**(7–8), 479–484 (2007)
10. yang, J., Woolbright, D.: Correlating TCP/IP packet context to detect stepping stone intrusion. Comput. Secur. **30**(4), 538–546 (2011)
11. Wu, H., Stephen Huang, S.-H.: Stepping stone intrusion detection using neural networks approach. J. Expert Syst. Appl. **37**(2), 431–1437, Mar 2010
12. Omar, M.N., et al.: A stepping stone perspective to detection of network threats. Int. J. Appl. Math. Inf. **7**(3), 97–106 (2013)
13. Omar, M.N. et al.: Hybrid stepping stone detection method. In: the Proceeding of 1st IEEE Conference on Distributed Framework and Applications (DFmA—2008), pp. 134–138 (2008)