This paper provides the brief summary of the concepts that are important to understand Stepping-Stone intrusion detection methods. Summary of intrusion detection systems helps us to understand Host based intrusion detection and Network based detection methods and different tools available to sniff the packets for detection purposes. The summary of different approaches to detect stepping-stone provides an idea on how the research is going on and what the areas that needs more work. This paper is divided in five sections covering the intrusion detection systems, different network sniffing tools, stepping-stone intrusion detection, and round trip time calculation algorithm to detect stepping-stone and intrusion detection evasion techniques.

Section 1: Host and network Intrusion detection

Even though intrusion prevention is the ideal methodology, it is not always possible to prevent the intrusions due to either the complexities involved to prevent or prevention is much costlier. So in situations where prevention is not possible, detection is helpful. With detection, systems can be strengthened to close the gaps and minimize intrusions. Also it helps to take some legal actions on the intruders to recover the financial and other loses etc. Detection is done by monitoring the different sensors like network packets, log files, system calls etc. and analyzing and auditing those sensors for anomalies either at each computer level (Host based) or at the network level (Network based) [1].

Host Based detection

In host based detections, a particular host or group of hosts which provides the services like hosting a web application or providing FTP services etc. are monitored, audited and analyzed for intrusions. These hosts are monitored and audited either by installing a special software on the host for capturing all the activities going on within the system or the using the existing system logs, application logs, and others logs on user activity. Special software creates the additional overhead but provides the results in the needed format. Native (existing) logs do not create any installation or processing overhead for capturing activities, but may need output to be translated into required format for further analysis [1]. The captured data is analyzed by using combination of the Anomaly detection and signature detection methods for intrusion detection.

Network Based detection

In network-based detections, traffic is monitored at select points on a network either real time or close to real time. Network, transport and application level protocol activity is monitored by deploying the sensors either in inline mode or passive mode. Inline sensors can be combined with other hard devices like firewalls to reduce the hardware overhead or installed as standalone device. Inline is primarily used to block the attacks when detected. Passive sensors
analyze the copy of the network traffic, so a physical tap is connected to transmit the data to sensor that is travelling on the medium. Passive sensors do not cause the packet delay [14].

Section 2: Network traffic sniffing tools

Network sniffing tools are computer program that intercepts and captures the packets of the traffic on a network [3]. On a positive side this sniffing is useful to analyze the network traffic for bandwidth utilization, network utilization, network statistics, decrypting packets for legitimate reasons, detecting intrusions etc [4]. On the flipside, these are used for capturing sensitive information like user-id's, passwords, credit card numbers, assess open ports etc for illegal purposes [4]. There are lots of free/community and commercial network sniffing tools available and can be used either on Windows based OS or UNIX based OS to sniff either the wired or wireless networks [3]. These tools are strategically plugged to the computer /device to sniff either single system or whole subnet. In the following paragraphs I am going to summarize couple of tools, but the Wikipedia link http://en.wikipedia.org/wiki/Comparison_of_packet_analyzers provides good list of tools with other details.

Sniffers working on the data link layer of the TCP/IP protocol are called packet sniffers. These are plugged to network interface card and translate the binary data to human readable format. Administrators use this data to monitor rogue data. Attacker has to be on same network to use these sniffers [6]. Usually Hackers sneak into the network using existing weakness/vulnerabilities to plant Trojans and run in promiscuous mode [6].

Wireshark

Wireshark is a client based network protocol analyzer which analyzes both live network traffic and captured data. It works on both unix based and windows based Operating systems, has user interface and provides features like language filter and TCP session reconstructions. It is has both free and licensed versions [5].

Cain and Abel

Surprisingly Cain and Abel also use the network sniffing methods to recover or crack the windows passwords. Cain and Abel tool has to be configured to the network card or MAC that needs to be sniffed either for legitimate password recovery or black hat activities. It is a freeware [8].

Kismet

Kismet is powerful open-source wireless sniffer passively sniffing the networks. It can detect wireless access points and wireless clients. It works on unix based systems and has ability to convert the sniffed packets into tcpdum/wireshark or Airsnort compatible files. It uses channel hopping methods to sniff more networks. It's also available as freeware and licensed tool [7].

Microsoft Network monitor

When the above three tools are used for both white hat and black purposes, Microsoft Network monitor is mainly used for legitimate purposes as it in needs the licensing. It captures, views, analyzes network data, and deciphers network protocols to trouble shoot network problems, application on network, monitor promiscuous mode traffic [9].
Section 3: Stepping-stone intrusion detection approaches

The intermediate connections between the source and destination are the stepping stones. Hackers craft their attacks using stepping stone to obtain anonymity. So detecting the stepping stones in illegal activity is complicated as normal/legitimate traffic also uses the stepping stones. Detecting the stepping stone can be done either in by passive monitoring or active perturbation methods and detecting is very crucial to identify malicious activity. Looks like is there lot of research going in this area. I will be summarizing five detection approaches which I came across in following paragraphs.

Connection Content-based Thumbprint

This was the early method proposed by Staniford-Chen and Herberlein, here the authors used the content of the connection, divided the connections into time intervals of one minute length and thumb printed each interval in such a way that each thumbprint has characteristic and uses minimum storage [11]. Different sessions are compared for similarities in the connection chain to identify the intruders. This algorithm does not work for the encrypted sessions as the content is not available [12].

Sleepy Watermark Tracing SWT

Wang and others proposed this method based on the content of the connection, it injects the some information as watermark into packets to trace the source when an intrusion is detected and keeps track of the tracing information. Thus this algorithm can trace the source effectively but needs infrastructure change and watermark overhead. This also do not work on encrypted sessions and not very effective if different stepping stones in the connection chain uses different encryption schemes[11].

Time Based Correlation

Zhang and Paxon used the time as the characteristic for their algorithm. They considered the “OFF” status if the network traffic is not travelling thru the host and “ON” if traffic is passing thru host. They correlated two connections in the same state i.e either OFF or On state on host to determine whether the host is stepping stone or not. This method works for encrypted sessions but needs the two connections to be synchronized [11].

Deviation Based

In this method Yoda and Etoh correlated the connections with unknown time shifts. Deviation is the difference between the average and minimum propagation delay between the connections. If the deviation is smaller implies the connections are in the same connection base. This method ignores the true positives [11].

Packet Number Difference Based PND

Blum and others proposed the algorithm that uses the difference between the number of SEND packets send between two connections to detect the stepping stones. When two connections are relayed then the difference should be bounded. [11]
Section 4: Round Trip time to detect Stepping Stone

Round trip time (RTT) can be defined as the total time (including the sum of processing delay, queuing delay, transmission delay and propagation delay) taken for TCP/IP packet to reach the destination and travel back to source. According to Yang and Huang, the differences in the TCP/IP packets round trip time can be used to detect the Stepping Stone intrusions. Yang and Huang applied Data mining techniques on the TCP SEND and echo packets to determine the round trip time for detection purposes [13].

Clustering–partitioning algorithm

Clustering–partitioning algorithm is one of the data mining techniques used to determine the RTT in the connection chain. The authors observed in their experiment that RTT of the connection chain changed/increased with increase in the number of clusters. i.e if the connection chain has four clusters, then the RTT of the connection chain is the RTT four clusters. Also authors observed that the partition of the clusters gave the different RTT for each disjoint partition. They used maximum-minimum distance clustering algorithm to build the clusters. Their experiment proved that when one level cluster used the RTT is produced is the expected value (i.e one level obeys poison distribution). They observed that when many clusters are applied in the connection chain, the clusters in the different levels of the chain have higher probability of including consecutive sends packets. Clustering–partitioning algorithm produces RTT’s for more send packets with better quality when compared to Conservative algorithm and Greedy algorithm [13].

Clustering–partitioning algorithm \((X, t, g)\):

Begin
1. Call MMD\((X, t)\), output clusters \(C_1, C_2, \ldots, C_v\).
2. For each cluster \(C\), (1) if \(t(i, j), t(i, k) \in C\), and \(j < k\), delete \(t(i, k)\), and (2) if \(t(i, j), t(k, j) \in C\), and \(i < k\), delete \(t(i, j)\).
3. For each cluster \(C\), compute the clustering ratio \(r_i = \frac{|C^r|}{\text{range}(C)}\), where \(C^r = \{t(i, j) \in C \mid \exists t(p, q) \forall C, \text{ and } |i - p| \leq g\}\).
4. Select clusters \(D_1, \ldots, D_v\) from the clusters \(C_1, C_2, \ldots, C_v\). They have significantly higher ratios among \(R = \{r_1, \ldots, r_v\}\). We consider \(r_i\) to be significantly higher than the rest of the values if two standard deviations are higher than the mean of \(R\).
5. Find among \(D_1, \ldots, D_v\), a maximum disjoint subset. If there is a tie, select the subset whose union is the largest.

Source [13]
Section 5: Hacker's evasion Techniques

Hacker usually evades the intrusion detection systems by manipulating the two unrelated TCP sessions look related or vice versa by employing either Time-jittering or Chaff-perturbation techniques.

**Time-jittering**

In this evasion technique hacker holds the SEND packets of a session for certain amount of time. Usually intruders generate the time gaps and incorporate those delays into SEND packets and send all those SEND packets in the original sequence order [12].

**Chaff-perturbation**

In this evasion, intruders introduce some meaningless packet into interactive session either making the relayed connections to un-relayed or vice versa. The number of SEND packets changes from the original transmission. Stepping-Stone intrusion detection system which depends on the number of SEND packets fails to resist the chaff-perturbation, since number of packets can be easily manipulated [12].

**Conclusion**

After reading the provided material and researching further to write this paper I understand that active research is going on in the Stepping-Stone intrusion detection. Stepping stones can be possible for legitimate reasons and not only as part of malicious activities. Detecting stepping-stone is complicated and existing algorithms/method has some kind of inefficiencies in detecting the stepping stone. Older methods do not work for encrypted data. Hackers can easily use time jittering or chaffing techniques to evade the detection to some extent with new methods. Right now hackers have some difficulty to use these techniques, but in no time they will fine tune these techniques to evade detections. So I feel more research is needed in this area.
References


