Solving Time Gap Problems Through The Optimization of Detecting Stepping Stone Algorithm

Mohd Nizam Omar[1], Mohd Aizaini Maarof[2] and Anazida Zainal[3]
[1, 2, 3]Group on Artificial Immune Network and Security (GAINS), Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, 81310 Skudai, Johore, Malaysia.
E-mail: mc023003@siswa.utm.my[1], {maarofma[2], anazida[3]}@fsksm.utm.my

Abstract

This paper describes an analysis of detecting stepping stone algorithm to defeat the time gap problem. It is found that current algorithm of detecting stepping stone is not optimized. Several weaknesses are identified and suggestions are proposed to overcome this problem. The suggestions are applied in the improved algorithm. Since the detecting stepping stone is listed as one of the response technique, it is suggested that the improved algorithm should be used as a remedial to the time gap problem.

1. Introduction

Intrusion Detection System (IDS) can be defined as a system that attempts to identify intrusion, such as unauthorized use, misuses, or abuses of computer systems by either authorized users or external perpetrators [6]. IDS can be divided into two categories, host- and network-based IDS [2]. From the input perspective, host-based IDS uses logs, system calls and so forth while network-based IDS use network packets as the main input [3].

IRS (Intrusion Response System) is an IDS that detects an attack and immediately responses to remove the intruder from network [9]. While IDS detects intrusion, IRS is responsible to respond after an intrusion is detected. IRS can perform various responses such as generating report, locking user account, terminating user session [8] and so on. Both IDS and IRS can use packet capturing program as their main source to detect and response.

The success of an attack depends on the time-gap between detection and response [7]. Some efforts to overcome the time-gap problem were accomplished [1] using the adaptive IDS and [13] applies the preventive approach. This paper, focus is given on the stepping stone algorithm. It is one of the tracing intruder techniques. According to [8] tracing intruder technique is listed as one of the response techniques. If the stepping stone algorithm can detect an intrusion in a shorter time, then the time for response technique can also be reduced. Therefore, it will reduce the time gap. In the study, five algorithms are analyzed from speed perspective: 1) Brute force algorithm [12], 2) Simple content-based I algorithm [12], 3) Thumbprint [10], 4) On/Off [12] and 5) Deviation [11].

This paper is organized as follows. Section 2 describes time gap. Section 3 explains the relationship between IRS and Stepping Stone Algorithm. Section 4 explains the Stepping Stone Algorithm. Section 5 focuses on analysis of the algorithms. Section 6 details the experiment. Section 7 discusses the result. Section 8 illustrates the optimized algorithm and Section 9 concludes the paper by outlining the future work.

2. Time Gap

Cohen [5] indicates that the success of an attack depends on the time gap between detection and response. If skilled attackers are given ten hours after they are detected before response is made, they will be successful 80% of the time. After thirty hours, the attackers almost never fail. Various methods can be used to optimize the time gap. Here, an improved Detecting Stepping Stone algorithm is used.

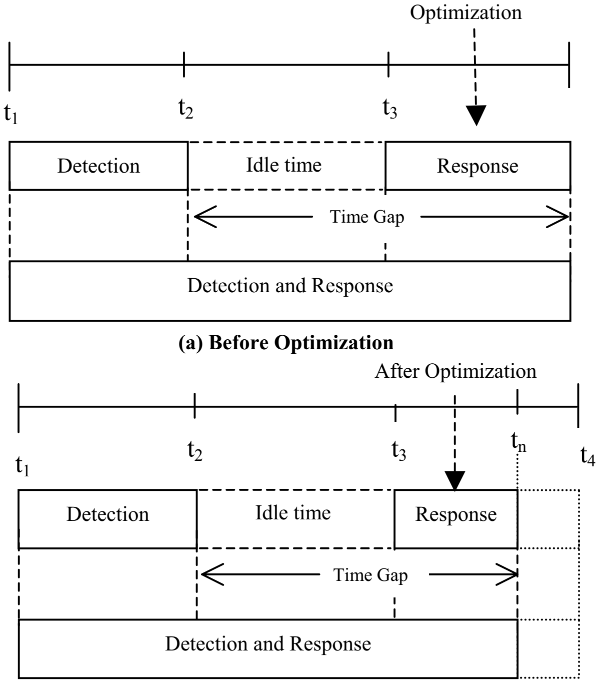
Figure 1 shows the time gap problem, before the optimization and after the optimization.

3. Intrusion Response to Stepping Stone Algorithm: The Relationship

Tracing intruder is one of the response techniques [8]. Tracing was chosen because according to Jang [17], it can identify intruders and prevent them from performing another intrusion. [18] Tracing Intruder is classified into IP-based and Connection-based. Connection-based is chosen because it not hardware dependent and it is more important than IP-based [18].

Connection-based can be divided into three areas, host-based, network-based and active network-based [18]. Stepping stone detection is under network-based and currently being researched by [10], [11], [12], [14], and [15].

Reducing time gap (by providing fast stepping stone detection algorithm) will reduce respond time.



(a) Before Optimization

(b) After Optimization

Figure 1. Time Gap

4. Stepping Stone Algorithm

Stepping stone is a computer used in a chained connection to separate the attacker from the target. This is done to make the identification of the attacker more difficult [16]. Stepping stone detection can be divided into three categories [14] such as content-based [10], activity-based [11] and time-based [14]. In this paper, content-based is chosen. Five algorithms of detecting stepping stone are analyzed and a proposed enhancement to the algorithm that focuses on the time of detecting stone will be produced.

4.1 Brute Force

Zhang and Paxon used Brute Force algorithm to compare with their work [12]. The algorithm works as follows.

1. Extract the aggregate Telnet output (computer-side response), for all of the sessions in the trace, into a file.
2. For each different line in the output, count how many time it requires (sort | uniq –c ; in Unix).
3. Throw away all lines except those appearing exactly twice. These are good candidates for stepping stones, in that they are lines unique to either one or at most two connections.
4. Find the connections in which each of these lines appears. This is done by first building a single file listing every unique line in every connection along with the name of the connection, and then doing a database join operation between the lines in that file and those in the list remaining after the previous step.
5. Count up how many of the only-seen-twice lines each pair of connections has in common (using the Unix join utility).
6. Connection pairs with 5 or more only-seen-twice line in common are now candidates for being stepping stones.
7. Of those, discard the pair if both connections are in the same direction (both into the site or both out of the site).
8. Of the remainder, visually inspect them to see whether they are indeed stepping-stones. Most are; a few are correlated due to common activities such as reading the same mail message or news article.

4.2 Simple 1 – Last Login

Simple 1 – Last Login algorithm observes the frequency of when a new interactive session begins, the dialog includes a status line like:

Last login: Fri Jun 18 12:56:59 from w. x.y.z

According [12], the combination of the timestamp and the previous-access host leads to this line being frequently unique. But it does not provide full detail of this algorithm. The full list of Simple 1 – Last Login algorithm is as follows.

1. Extract the aggregate Telnet output (computer-side response), for all of the sessions in the trace, into a file.
2. Search invariance traffic characteristics contains *Last login: Fri Jun 18 12:56:59 from w.x.y.z.*
3. Compare the connection pair for each connection that has same invariance traffic characteristics.
4. Connections with the invariance traffic characteristics are now candidates for being stepping stones.

4.3 Thumbprint

Although [10] did not explicitly describe the algorithm, below is the recreation of the algorithm based on our study and observation. The algorithm of thumbprint works as follows.

1. Analyze each packet and associate it with a particular pair of machines and ports it is traveling between.
2. Use thumbprint function to get thumbprint value.
3. Compare this value to connection values.
4. Connection with close compared value is now candidates for being stepping stone.

4.4 On/Off

Since the on/off algorithm was not clearly shown in [12], the algorithm below is based on the observation done in Zhang's paper. The algorithm of on/off works as follows.

1. Use Bro, a real-time intrusion detection system to trace Internet traffic record.
2. Find Off period for each connection.
3. Correlate Off period among the connections.
4. Connection with similar Off period is now candidate for being stepping stone.

4.5 Deviation

Deviation research is done in [11]. Based on our review on [11] below is the Deviation algorithm.

1. Packets are collected at various traffic points in the Internet backbone networks.
2. Plot a graph of a packet stream with sequence numbers of the packet on the Y-axis and its capture time on the X-axis.
3. Analyze the deviation for packet stream for each connection.
4. Connection with small value of deviation is now a candidate for being stepping stone.

5. Analysis and Discussion of the Algorithm

Three general steps are identified in the detecting stepping stone algorithms, which are log extraction, identification and comparison. Log extraction is the processes where the information from raw source (network connection) is extracted so that it can be used for the next step in detecting stepping stone algorithm.

Identification is the second step in the detecting stepping stone algorithm. Even though each one of the studied algorithms exercises different techniques of identification, the main idea of this step is to identify network connection's unique identity.

The final general step is comparison. It refers to the process of comparing the unique identity of the network connection (obtained from the previous step).

The discussion in the following text will focus on how to improve the detection in terms of speed. Since log extraction deals with the method on how a log file is obtained, we can focus on the information that we need. The filter can be set with particular options that can give more accurate result. If the filter is not set, the result will yield unnecessary information and not fit to our needs. For example, using WinDump [4] packet capture facility yields different results from using filter.

For example, by using the filter, the first command will capture all packets and the second command will capture packets between host with IP address x.y.z.com and a.b.c.com. This user-defined filter decides whether a packet is to be accepted and how many bytes of each packet should be saved [19]. Loris also said that if the data of the packet is not needed (line in the most part of the capture applications), filter can be set to keep the headers only. For this reason WinDump set a filter that tells the driver to save only the first 68 bytes of each packet [23]. [20] reaffirms that the reduction of data obtained can be accomplished if filter is used.

In Identification step, Brute Force algorithm uses the content of packet data as invariant traffic characteristics. This has caused the algorithm to read all data content, which requires longer time to accomplish. Long string as "*Last login: Fri Jun 18 12:56:59 from w.x.y.z*" that used by Simple algorithm 1 causes algorithm to take more time to accomplish. Zhang [21] suggested that other invariance traffic characteristics are; 1) Connection contents, 2) Inter-packet spacing, 3) ON/OFF patterns of activity, 4) Traffic volume or rate and 5) Combinations of the above. From our finding, how data is obtained in identification step depends on the location of field that needs to be captured.

In comparison step, the improvement focus on counting connection pairs with the chosen connection pairs of encounter-twice. This is done by reducing the number of chosen connection pairs from 5 to 4 or less without disturbing the detecting stepping stone algorithm.

Brute Force	Simple I	Thumbprint	On/Off	Deviation	
1. Extract log	1. Extract log	1. Extract log	1. Trace wide-area	1. Collect packet	Log Extraction
2. Count 3. Throw 4. Brute Force processes 5. Choose 5 pairs	2. Search string	2. Generate local thumbprint	2. Identify on/off period	2. Plot graph	Identification
6. Discard if same direction 7. Side effect	3. Choose connection	3. Compare thumbprint	3. Correlate on/off connection	3. Analyze the deviation	Comparison

Figure 2. Comparison of Stepping Stone Algorithm and the General Steps

Since inter-packet delay invariant traffic characteristic can do comparison by using a small packet sequence [15], comparison step can be improved by using this packet delay capabilities. Figure 2 shows the overall comparison of stepping stone algorithm and the general steps.

6. Experiment

This section describes the experiment conducted to demonstrate that the optimization can be done in each general steps discussed earlier. One experiment will be executed for each general step. The experiment represents the condition before and after optimization processes. Execution time will be taken before and after optimization processes. For log extraction step, execution time without using the filter and with the filter is done. For the same data set, log extraction has been done using this command:

i) *windump*
ii) *windump host w.x.y.com and a.b.c.com*

First command represents the log extraction without using the filter (i) and second command represents otherwise (ii). Here, filter refers to statement used to reduce the number of data captured by the *windump* program. *ptime* [22] software is used to capture the execution time for each commands. The testbed of this experiment is shown in Figure 3.

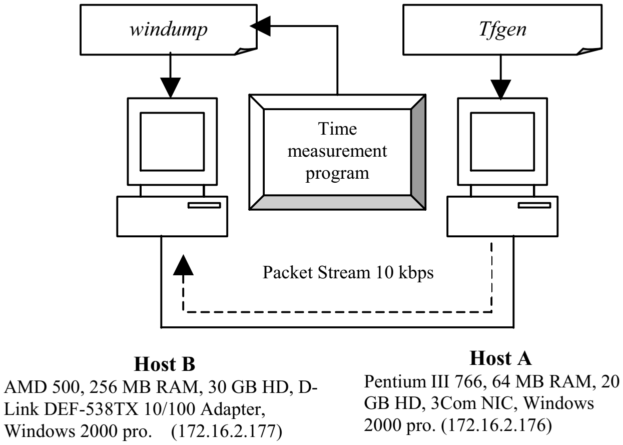
For identification step, two techniques are tested:

i) Simple I
ii) Packet Delay

First technique represents the technique that requires searching the string in data portion of network packet (i) and the second is the technique that requires only searching on the initial part of the network packet (ii). Other identifying techniques include thumbprint [10], deviation [12], on/off [11] and so on. Thumbprint uses the technique that searches character at the data portion of a network packet. Deviation uses the technique that searches unique data at the header of the network packet and On/Off research also uses technique that requires searching the unique data at the header of the network packet.

Both techniques are coded in Java and data set is obtained from Ethereal software. Each technique identifies the data set and identification period is recorded. The testbed for this experiment is shown in Figure 3. The two comparison techniques used are:

i) Compare only one network packet characteristic
ii) Compare more than one network packet characteristics



Host B
AMD 500, 256 MB RAM, 30 GB HD, D-Link DEF-538TX 10/100 Adapter, Windows 2000 pro. (172.16.2.177)

Host A
Pentium III 766, 64 MB RAM, 20 GB HD, 3Com NIC, Windows 2000 pro. (172.16.2.176)

Figure 3. Extracting Log Testbed

7. Result

The main focus of this section is to discuss results obtained on execution time for each general step, before and after optimization processes.

7.1 Log Extraction

Figure 4 shows the time measurement of log extraction by using filter and without filter. Using the same raw data (periodical traffic pattern) for each set of tests i) extracting log using filter and ii) extracting log without using filter, graph shows that the time used by (i) is smaller than the one that does not use filter (ii). For example, on the 100th packet, time used for log extraction is around 1 millisecond by using filter and around 2 milliseconds without using filter. The null value of the time measurement actually shows there is no traffic data on that time. Thus, it can be concluded that extracting log file using filter may reduce the time.

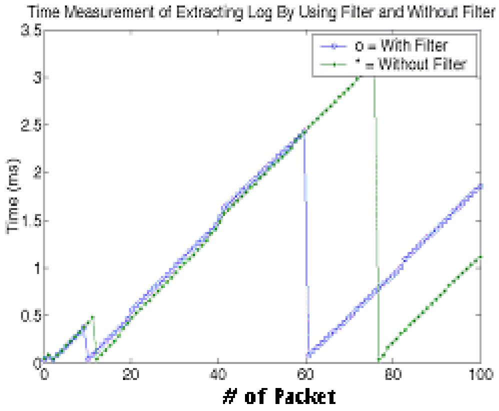


Figure 4. Time Measurement for Log Extraction With and Without Using Filter

7.2 Identification

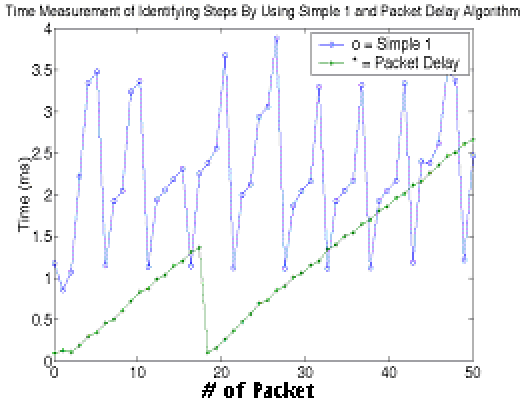


Figure 5. Time Measurement of Identification Step By Reading Data Portion and Initial Part of Network Packet

Figure 5 shows the time measurement for identification step obtained by reading only the initial part of network packet and the data portion of the network packet. In this experiment, the technique that identifies a network packet by reading the initial part of network packet shows better performance than the technique that requires to read the data portion of the network packet For example, on 10th packet, time measurement to read initial part of network packet shows 0.5 milliseconds is used while the other technique requires around 3.5 milliseconds. Thus it can be concluded that a better identification (shorter time) can be performed by reading only the initial part of the network packet.

7.3 Comparison

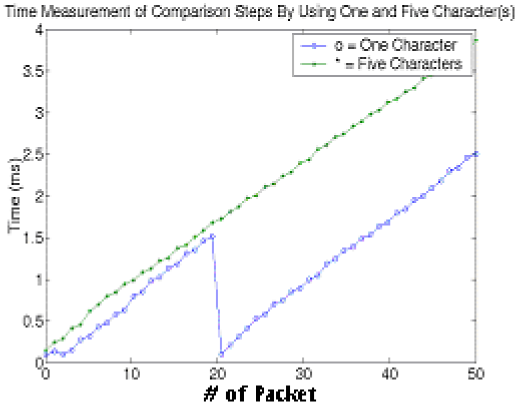


Figure 6. Time Measurement for Comparison Step Using One and Five Characteristics

Figure 6 shows that time measurement for comparison step between technique that uses only one characteristic and the technique that uses more than one characteristics of network packet. Comparison using only one character consumes shorter time compared to using five characters. This can be observed at 30th packet. Time taken for comparison using one characteristic takes around one millisecond and comparison using five characteristics requires 2.5 milliseconds. Thus, it can be concluded that comparison using one characteristic may reduce the time for comparison step.

8. Optimized Algorithm

In optimizing the stepping stone algorithm, we have adopted the techniques that produce shorter time in all the general steps. Below are the general steps of the optimized algorithm.

1. Use filter to extract log.
2. Identify the unique identity by reading the initial part of network packet.

3. Compare unique identity using less characteristic of unique identity.

9. Conclusion and Future Work

It is concluded that the current algorithm of detecting stepping stone can be optimized to solve time gap problem. By adopting techniques which have shorter time measurement, time gap problem will be remedied. Thus, in solving time gap problem, the improved algorithm is proposed to be used in detecting stepping stone application. This will benefit the tracing intruder system and response system.

The initial experiment conducted in this research provides an impetus for improvement of the stepping stone algorithm. For the future work, extensive experiments on each of the algorithms discussed in this paper will be performed to find the best general steps for the stepping stone algorithm. If this is achieved, the time gap problem between detection and response can be greatly reduced.

10. References

[1] Ragsdale, D. J., C. A. Carver, J. W. Humphries, and U. W. Pooch, Adaptation Techniques for Intrusion Detection and Intrusion Response System. 1998

[2] Mukherjee, B., T. L. Heberlein, and K. N. Levit, Network intrusion detection. IEEE Network, 8(3): 26-41, May/June 1994.

[3] Kerschbaum, F., E. H. Spafford, D. Zamboni, Using embedded sensors for detecting network attack. Proceeding of the First ACM Workshop on Intrusion Detection Systems, November 2000

[4] Fulvio, R. and D. Loris, An Architecture for High Performance Network Analysis, Proceedings Sixth IEEE Symposium on, 2001, 686-693

[5] Cohen, F. B., Simulating Cyber Attacks, Defenses, and Consequenses, http://all.net/journal/ntb/simulate/simulate.html

[6] Puketza, N. J., K. Zhang, M. Chung, B. Mukhejee, and R. A. Olsson, A Methodology for Testing Intrusion Detection System. IEEE Transactions On Software Engineering, Vol. 22, No. 10, 1996.

[7] Huagang,X., LIDS Hacking HOWTO, Document for LIDS, v1.0

[8] Carver, A. C., Intrusion Response Systems: A Survey, Department of Computer Science, Texas A&M University, Collage Station, USA.

[9] Kulin, H. T., H. L. Kim, Y. M. Seo, G. Cheo, S. L. Min, C. S. Kim, Caller Identification System in the Internet Environment, Proceeding of the USENIX Security Symposium IV, 1993

[10] Staniford-Chen, S., L. T. Heberlein, Holding Intruders Accountable on the Internet. Proceeding of IEEE Symposium on Security and Privacy, 1995

[11] Yoda, K. H., Finding a Connection Chain for Tracing Intruders, In F.Guppens, Y. Deswarte, D. Gollmann and M. Waider, editors, 6th Eropean Symposium on Research in Computer Security – ESORICS 2000 LNCS-1985, Toulouse, France.

[12] Zhang,Y., V. Paxon. Detecting Stepping Stone. In Proceeding of 9th USENIX Security Symposium.

[13] Foundstone, Inc. Managed Security Service. 2 Venture Street, Suite 100, Irvine, CA 92618.

[14] David, D. L., F. A. Georgina, S. Umesh, P. Vern, C. Jason, S. Stuard, Multiscale Stepping-Stone Detection: Detecting Pairs of Jitttered Interactive Streams by Exploiting Maximum Tolerable Delay, Fifth International Symposium of Recent Advance in Intrusion Detection, October 16-18, 2002, Zurich, Switzerland.

[15] Xinyuan, W., R. S. Douglas, W. S. Flix, Inter-Packet Dalay Based Correlation for Tracing Encrypted Connection Through Stepping Stones, Esorics 2003 Symposium, October 14.

[16] Martin, A., and C. Markus. Intrusion Detection Systems–Technologies, Weaknesses and Trends, 25-02-2003

[17] Jang, H., S. Kim, A Self-Extension Monitoring for Security Management, Computer Security Application, ACSAC' 00, 16th Annual Conference 2000. Pages(s): 196-203.

[18] Dong-il, S., Trend & Technique of Intruder Traceback, ITU-T Workshop on Security, Seoul, Korea, 13-14 May 2002.

[19] Loris, D., Development of an Architecture for Packet Capture & Network Traffic Analysis,Thesis, Mar 2000.

[20] Rainer, O., G. Oliver, and S. Markus, VisuSniff: A Tool For The Visualization Of Network Traffic, Proceedings of the Second Program Visualization Workshop, HornstrupCentret, Denmark, 27-28 Jun 2002, ISSN 0105-8517, pp. 118-124

[21] Yin, Z., P. Vern, Detecting Stepping Stones, Slide Presentation, available at http://gaia.cs.umass.edu/security/slides/yugu.ppt, Jun 2003.

[22] Available at http://www.pc-tools.net/win32/freeware/ptime/