Finding TCP Packet Round-Trip Time for Intrusion Detection: Algorithm and Analysis

Jianhua Yang[1], Byong Lee[1], and Yongzhong Zhang[2]

[1] Department of Mathematics and Computer Science, Bennett College
900 E. Washington Street, Greensboro, NC 27401 USA
{jhyang, blee}@bennett.edu
[2] College of Management, the University of Shanghai for Science and Technology
516 Jungong Rd, Shanghai, 200093 China
yzhang@shtvu.edu.cn

Abstract. Most network intruders launch their attacks through stepping-stones to reduce the risks of being discovered. To uncover such intrusions, one prevalent, challenging, and critical way is to detect a long interactive connection chain. TCP packet round-trip time (RTT) can be used to estimate the length of a connection chain. In this paper, we propose a Standard Deviation-Based Clustering (SDC) Algorithm to find RTTs. SDC takes advantage of the fact that the distribution of RTTs is concentrated on a small range to find RTTs. It outperforms other approaches in terms of packet matching-rate and matching-accuracy. We derive an upper-bound of the probability of making an incorrect selection of RTT through SDC. This paper includes some experimental results to compare SDC with other algorithms and discusses its restrictions as well.

Keywords: Network security, intrusion detection, round-trip time, stepping-stone.

1 Introduction

The use of stepping-stones [1] to attack other computers is gaining popularity on the Internet. One way to prevent this kind of attack is to detect stepping-stones while they are connected. There have been many methods proposed to detect a stepping-stone, such as methods in [1], [2], [11], [12], [13]. The methods proposed in [1], [2] have the problem of not only being vulnerable to an intruder's manipulation, but also having high false positive rate in detecting stepping-stone intrusion. One way to overcome these problems is to examine the whole connection chain to estimate the number of connections (length). There is no good reason to use a connection chain of length three (3) or more to access a host. Yung [3] claims that such a method makes it more difficult for intruders to manipulate connections by delaying or inserting packets.

Estimating the length of a connection chain which starts from a host where a monitor program resides and ends at a victim host has been a focus of research on

D. Pointcheval, Y. Mu, and K. Chen (Eds.): CANS 2006, LNCS 4301, pp. 303 – 317, 2006.

detecting stepping-stone intrusion. Yung [3] proposed to use Ack-delay and Echo-delay comparison to detect an intrusion. His basic idea is to estimate the length of a downstream connection chain by computing the ratio between packet Ack-delay and Echo-delay. To estimate the packet Ack-delay and Echo-delay, Yung proposed to use a statistic method to match TCP send and echo packets. It can result in a correct match only when the echoed packet for one send packet comes back before the next packet is sent. To understand this point well, we here present an example in which we simply assume that each send packet is only echoed by one corresponding packet. For example, it would be trivial to match each send and its corresponding echo in a packet sequence $\{s_1, e_1, s_2, e_2, s_3, e_3, s_4, e_4\}$ in which s_i and e_j represent the timestamps of i^{th} send and j^{th} echo packet respectively. However, if the sequence became $\{s_1, s_2, s_3, e_1, e_2, s_4, e_3, e_4\}$, according to the method in [3], the matching result would be pairs (s_3, e_1), and (s_4, e_3), rather than pairs (s_1, e_1), (s_2, e_2), (s_3, e_3), and (s_4, e_4). Yang and Huang [4] proposed a method to detect stepping-stone intrusion by detecting a long interactive session. Their method is based on the idea that it is highly suspicious to access a host through three or more connections. The method used in [4] to estimate the length of a connection chain from the monitor host to the final destination is to find the RTTs for each level. The number of different levels indicates the chain has different number of connections. Computing the RTTs of TCP packets is essentially to match send and echo packets. Yang and Huang [4] proposed two algorithms to match TCP/IP send and echo packets: the Conservative and the Greedy algorithms. The Conservative algorithm can give accurate packet matching result but with low matching-rate. The Greedy algorithm could 'match' all the send packets but with some incorrect 'matches'. There is a tradeoff for packet matching-rate and matching-accuracy between these two algorithms.

Matching TCP send and echo packets is equivalent to computing TCP packet RTTs. This process is challenging and is significant in detecting stepping-stone intrusion. Whatever method in [3] or [4] is used, only when we have matched send and echo packets precisely can we detect intrusion accurately. In this paper, we propose a novel algorithm SDC to compute the RTTs of TCP packets more accurately than the methods proposed in [4]. We also evaluate the performance of SDC by computing the probability of making a correct selection of RTT through the Chebyshev inequality. SDC can match most of send packets with the same level of correctness as the Conservative algorithm produces, and it produces the same packet matching rate as the Greedy algorithm does, but with a higher matching-accuracy. The experimental results showed that SDC can get both high packet matching-rate and matching-accuracy.

The rest of this paper is arranged as follows. Section 2 discusses the algorithm SDC in details and presents its probabilistic analysis. Section 3 presents some experimental results. In Section 4 we talk about related work. Finally in Section 5 we summarize this paper, discuss the limitations of SDC, and present future work.

2 The Algorithm and Its Probabilistic Analysis

It is easy to match each send packet with its corresponding echo packet when the echo packet of a send is always received before the next packet is sent. This is