Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Statistical dynamic splay tree filters towards multilevel firewall packet filtering enhancement

Zouheir Trabelsi[*], *Safaa zeidan, Mohammad M. Masud, Kilani Ghoudi*

College of Information Technology, College of Business and Economics, UAE University, Al-Ain, United Arab Emirates

ARTICLE INFO

Article history:
Received 26 October 2014
Received in revised form
19 May 2015
Accepted 25 May 2015
Available online 4 June 2015

Keywords:
Firewall performance
Packet filtering
Chi–Square Test
Early packet rejection and acceptance
Binary search on prefix length
Splay tree
Hash table
System stability

ABSTRACT

Network Firewalls are considered to be one of the most important security components in today's IP network architectures. Performance of firewalls has significant impact on the overall network performance. Firewalls should be able to sustain a very high throughput and ensure network services availability. In this paper, we propose an analytical dynamic multilevel early packet filtering mechanism to enhance firewall performance. The proposed mechanism uses statistical splay tree filters that utilize traffic characteristics to minimize packet filtering time. The statistical splay tree filters are reordered according to the network traffic divergence upon certain threshold qualification (Chi–Square Test). That is, the proposed mechanism is able to decide whether or not there is a need to update the dynamic splay tree filters' order for filtering the next network traffic window and predict the best order pattern. Furthermore, the importance of optimizing packet rejection and acceptance is done through the multilevel packet filtering process; where in each level, unwanted packets are rejected as early as possible. The proposed mechanism can also be considered as a device protection mechanism against denial of service (DoS) attacks targeting the default filtering rule. Early packet acceptance is done using the splay tree data structure which adapts dynamically according to network traffic flows. Consequently, repeated packets will have less memory accesses and therefore reduce the overall packets filtering time as demonstrated in the evaluation section.

1. Introduction

Firewalls use security policies to inspect incoming and outgoing network traffic. A security policy consists of a set of filtering rules. Each filtering rule is defined by a set of filtering fields, and associated with an action either to block or forward a packet to its destination. The last rule in a security policy is the default filtering rule which is usually assumed to be "Deny".

Firewall packet filtering is performed in a sequential order starting from the first rule until a matching rule is found. If no matching rule is found, the packet is processed by the default rule. This is called the naïve linear search algorithm of the rule-base. Thus, the computational complexity of the filtering process depends significantly on the length of each filtering rule (number of rule fields) as well as the depth of finding a matching filtering rule in the security policy (number of filtering rules).

* *Corresponding author.*
E-mail addresses: trabelsi@uaeu.ac.ae (Z. Trabelsi), safaa.z@uaeu.ac.ae (S. zeidan), m.masud@uaeu.ac.ae (M.M. Masud), kghoudi@uaeu.ac.ae (K. Ghoudi).
http://dx.doi.org/10.1016/j.cose.2015.05.010
0167-4048/

A significant drawback of the linear search is that, unwanted traffic targeting specific rules such as the default filtering rule may cause more harm than others by producing an overhead to the system. This overhead is proportional to the number of rules used in the security policy. Such unwanted network traffic may cause a DoS attack situation and consequently may degrade the firewall performance considerably. Thus, it is very important to reject such network traffic as early as possible.

In this paper, we propose a mechanism to optimize the early acceptance as well as early rejection packet filtering paths of a firewall. The mechanism builds the security policy filters using splay tree data structure that changes dynamically according to traffic flows. This characteristic of the splay tree allows for the early acceptance of repeated packets. On the other hand, the mechanism uses multi-filtering levels to reject unwanted network traffic as early as possible.

Based on network traffic statistics, a decision is made regarding whether or not there is a need to reorder the splay tree filters and find the best order that suits the next traffic window. The proposed mechanism is based on the following six optimization levels: 1) Splay tree filters are built using splay tree data structure which adapts dynamically according to the network traffic flows allowing early acceptance for repeated packets (optimization in the acceptance path). 2) Splay tree filters are reordered in a descending manner according to their packet rejection rate. This reduces the time required for comparing packets with splay tree filters as well as early rejecting packets that do not match any splay tree filter (optimization in the rejection path). 3) Each splay tree filter has its own early rejection method by using a tree node that contains the minimum prefix length. If a packet does not match the node of minimum prefix length in a specific filter, it deserves the early rejection (optimization in the rejection path). 4) Packets will not propagate to the next adjacent splay tree filter field until they pass a cascaded filtering level (optimization in the rejection path). 5) The firewall will continue filtering packets using certain order of splay tree filters under a certain threshold qualification (Chi–Square stability test) to compromise between performance and cost overhead (Fig. 2). 6) The optimum traffic window size that minimizes the packet processing cost is obtained empirically and offline using a training traffic that represents the network under normal behavior. The training data allows identifying the best initial order of the splay filters when the network is in its normal status. Hence, in case of a DoS attack, the order of the splay filters may change according to the Chi–Square stability test decision, which is based on the DoS attack significance. These optimization levels are expected to minimize the total packet filtering time which results in improving the overall firewall performance, as demonstrated later in the evaluation section.

The proposed early packet rejection and acceptance technique using statistical splay tree filters exhibit lightweight implementations and do not require special support in the firewall (i.e., can be easily integrated). Moreover, the mechanism can be generalized for other filtering network and security devices, such as routers and intrusion detection systems (IDSs).

The paper is organized as follows: Section 2 discusses the related work. Section 3 gives an overview of splay trees and binary search on prefix length algorithm. Then Section 4 details the proposed mechanism with illustrations of the multilevel early packet rejection mechanism and Section 5 presents the mathematical model of the proposed mechanism. Section 6 then sketches the proposed algorithms with detailed description and Section 7 evaluates the effect of the proposed mechanism on the firewall performance. Finally, Section 8 concludes the paper with directions to future work.

2. Related work

Packet filtering in firewalls is done by sequentially searching the rule list until a matching rule is found. This is the basic search algorithm and it is very efficient in terms of memory, but its scalability is generally poor as the packet filtering time is proportional to the length and depth of the filtering rules. In addition, the orders of both filtering rules and rule-fields have a significant impact on the packet filtering time. The earliest research works on firewall performance focused on improving packet searching time using various algorithms, including Hardware-based solutions (Baboescu and Varghese, 2001; McAulay and Francis, 1993) where Content Addressable Memory (CAM) was used to match multiple filtering rules in parallel. However, these algorithms were limited by the power, cost and size of CAMs to suit only small size policies. There were also software based solutions, such as specialized data structures, heuristics, and geometric algorithms based approaches. Specialized data structures approaches (Srinivasan et al., 1999) built a cross-products table of all possible field values combinations, but the table size grew dramatically with the number of filtering rules. Heuristics approaches (Gupta and McKeown, 2001) pipelined lookup stages, resulting in high filter throughput, but the algorithms did not scale with the increase in number of filtering rules. Finally, Geometric algorithm based approaches (Feldmann and Muthukrishnan, 2000; Gupta and McKeown, 1999; Cohen and Lund, 2005; Thomas, 2000) introduced Fat Invert Segment (FIS) tree or decision tree based on geometric cutting.

Research works related to the proposed mechanism in this paper contributed to enhance the filtering optimization problem using one or more of the followings: traffic awareness techniques, early packet rejection and acceptance techniques, reordering of dependent and independent filtering rules as well as rule-fields.

Although all previous works (Baboescu and Varghese, 2001; McAulay and Francis, 1993; Srinivasan et al., 1999; Gupta and McKeown, 2001; Feldmann and Muthukrishnan, 2000; Gupta and McKeown, 1999; Cohen and Lund, 2005; Thomas, 2000) have significant contributions to the packet classification research field, their major objective was focused on improving the worst-case filtering time rather than optimizing the average filtering time. To overcome this optimization problem, research works (Gupta et al., 2000; Hamed et al., 2006a, 2006b; El-Atawy et al., 2007; Al-Shear et al., 2009) proposed approaches that used statistical structures in optimizing the average packet filtering time. In (Gupta et al., 2000), alphabetic tree was used to reduce the lookup time by only searching packet destination IP addresses against entries in the routing