INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact factor: 4.295

(Volume3, Issue1)

Available online at: www.ijariit.com

Adaptive Packet Filtering Techniques for Linux Firewall

Prof. Sanjay Kadam
Bharati Vidyapeeth College of Engineering
Sector-7, C.B.D, Belpada, Navi Mumbai-400614, India.
sanjaykadam23@gmail.com

Prajakta .S. Tambade
Bharati Vidyapeeth College of Engineering
Sector-7, C.B.D, Belpada, Navi Mumbai-400614, India.

Atul .J. Jayant
Bharati Vidyapeeth College of Engineering
Sector-7, C.B.D, Belpada, Navi Mumbai-400614, India.

Abstract— Packet filtering techniques play an important role in many of network devices such as firewalls, IPSec Gateways. Firewall plays an important role in safeguarding any system from any external attacks to the system. It can be used to safeguard hosts as well as networks. This research focuses on studying the performance impact and the sensitivity of the Linux firewall (iptables) also improve by using these researches. And these are improving to become fast. A firewall designed in Linux, user can edit the source code and change it depending on the security requirements for the LAN. At any time one can configure the firewall to encrypt, to decrypt, accept, deny, or proxy all packets that are being sent between any two systems depending on the rules. On the basis of this the user can be blocked or given access to a network using a good tree algorithm. There are two approaches for the filtering, first by using the early rejection of unwanted flows without impacting other flows significantly. Second, we present a new packet filtering optimization technique that uses adaptive statistical search trees to utilize important traffic characteristics and minimize the average packet matching time. The proposed techniques timely adapt to changes in the traffic conditions by performing simple calculations for optimizing the search data structure. The proposed techniques can significantly minimize the packet filtering time with reasonable memory space requirements. [1].

Keywords— Linux, firewall, proxy, iptables, acl or net filter access.

I. INTRODUCTION

A firewall sits between the ‘internal’ network and ‘external’ internet. Most of systems have in-built firewall. Depending the firewall can be distinguished in two types. The desktop/personal firewall and network firewall. These both boils down to one thing that is no of users that can be used to protect. Packet classification is a used for making such firewalls and IPSec gateways, Intrusion Systems and many more.[11] The main purpose of packet filter is to categorize packets based on the rules made for packet filtering policy. The information used for classifying packets is usually contained in distinct header fields in the packet, which are protocol field, source IP, source port, destination IP, and destination port in Ipv4. Each filtering rules consists of an array of these header-sets. Each rule is associated with actions to be performed for packet filtering. Packets P having the header-field are matched with these rules. These actions then help to decide whether to ‘block’ or ‘allow’ the packet to a particular interface. Consider a firewall policy is made to only accept a filtering rule R having tcp, 140.49.23 to accept the packet, the firewall consists of N policies having rules R1, R2... Rn. Since any packets may match multiple rules but the highest priority is given to the first rule in the policy. But if there is no match then it is discarded because the default rule (last rule) is said to deny. As the time goes by there is need to make more optimized packet filtering policies as there is increase in the speed of internet thus increasing the no of users.[1]

First thing required to make a technique that can analyze the policies and make a rules set that can reject the maximum no of unwanted packets. The main objectives for using these techniques is to monitor the incoming and outgoing packets and judge which should be filtered or not and track down any intrusion and record it for security purpose. It is also used to shield the system for any external attacks. To create a user friendly firewall that can be cited by the user depending on the requirements and to create a less complex firewall that can be used easily.

II. EXISTING SYSTEM

A person who wants to communicate with someone in a different network thus he needs to use a secure program. Firewall is one efficient way of transferring the data or information in a secure manner. There are organizations that provide these services but there are some problems faced by the users: They bind the services for a limited period time. Many real time services are failed to reach users. They have to follow each and every rule given. They cannot customize their own firewall. The readymade firewall tends to give many problems. Linux is an open source system hence can be edited.

III. FIREWALL TECHNOLOGIES

This section focuses on the technologies used in various firewalls and how they work. The firewall taxonomy in Figure 1 shows the general types of firewalls.

This section focuses more on the underlying technologies that devices that fall into those types utilize. In some cases, one technology discussed here can fall into multiple types in the taxonomy tree. The focus is on a wide range of firewall technologies, including the following: Personal firewalls, Packet filters, Network Address Translation (NAT) firewalls, and Circuit-level firewalls

Personal firewalls:-are used to protect single hosts only. They are mostly used for desktops, servers. In these firewalls the outbound traffic are permitted where as the inbound traffic are inspected.

Packet Filters: - these are devices that filter traffic based on simple packet characterisation. These devices are typically stateless.

NAT firewalls: - this firewall exists for a short period of time and is used mostly in all products.

Circuit-level firewalls:-these firewalls work in the session layer of the system and mostly moniter the “Handshaking” between the packets [10]

Packet filtering Techniques

- Early Traffic rejection: firewall rules are written to either accept or reject the packets of the incoming traffic. The packets which have the most likely resemblance are allowed or denied. A rejected packet has to go through a long list of rules thus taking a toll on the time and resources consumed hence in order to avoid this a default-deny rule must be set so that a packet does not has to go through the list if it is denied. There can be techniques implanted that can reduce this matching of discarded packets by introducing an optimal set of early rejection rules in firewall policies. If a packet does not match any of the field values of the allow rules the it should be discarded earlier so that it does not go through the long list for e.g., a rule list has the field set of the destination port or subnet if a packet does not have similar port or subnet can be rejected early on and avoid further matching.[11]

- Statistical optimization using trees: A tree can be used by providing it the properties and the traffic can be passed thus giving an optimal searching time. An adaptive alphabetical tree can be used to have shorter search paths for more frequently used field values. This results in a significant matching reduction for most of the traffic. The matching frequencies are all calculated from the traffic over a period of time then this statistics are used to make an alphabetical tree for the filtering field. The constructed tree is used to obtain an optimal statistics matching tree. Then this tree can be updated/reconstructed periodically to match the most recent characteristics.

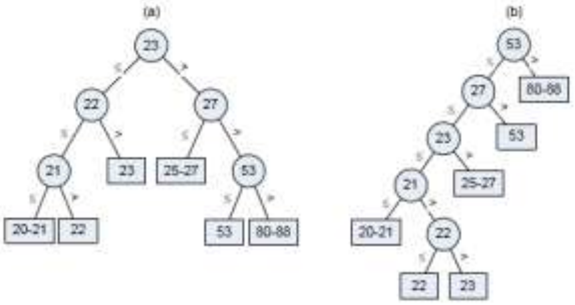
- ***Skewed field value frequencies:*** The field value frequency is the number of packets that carry this field value within a certain interval of time. The field frequency distribution is said to be skewed if few field values have high frequencies in comparison to the frequencies of other values in the same time interval. To measure this skewness we use information theory. The skewness factor *Sf* of a filtering field *f* is a value between 0 (for a non-skewed or uniform distribution) and 1 (for a totally skewed distribution). *If* is defined by the formula where *pi* is the probability of field value *vi* and it is calculated as the ratio of the number of packets matching *vi* to the total number of packet received. Also *n* is the number of possible values of field *f*.

$$s_f = 1 - \frac{\sum_{i=1}^{n} p_i \log(p_i)}{\log(n)}$$

- Time-correlated field value frequencies*:* The field frequency distribution is said to be *time-correlated* if the frequencies of the field value is similar over the two intervals. We use the *correlation factor Cf* of field *f* as a value between 0 (for an uncorrelated distribution) and 1 (for a totally-correlated distribution), and it is calculated as follows: where *pi* is the probability of field value *vi* in a certain time interval, and *qi* is the probability in the following interval. The quantities *µp* and *µq* represent the mean, while *σp* and *σq* represent the standard deviation of the probability distributions.

$$C_f = \frac{\sum_{j+1}^{n} (p_j-\mu_q)(q_j-\mu_q)}{n.\,\sigma_p.\sigma_q}$$

- Statistical matching tree*:* A statistical search tree can be built using the values of each filtering field in order to minimize the average matching time. This tree basically inserts values of higher occurrence probability (matching frequency) at higher tree levels than the values with less probability. This way, field values that commonly exist in the traffic will exert less number of packet matches in comparison to uncommon values, resulting in a significant reduction in the matching of most popular flows, reducing the overall average filtering time of all flows. [7]



Field	Value	Statistics
dsp_port	25-27	0.11
dsp_port	23	0.01
dsp_port	53	0.19
dsp_port	80-88	0.60
dsp_port	20-21	0.08
dsp_port	22	0.01

Matching tree construction using alphabetic trees

The alphabetic tree stores field values in the leaves based on given weights such that the inherent order of the stored values is preserved. So, at each internal node we have the left subtree containing nodes that have values less than those at the right hand-side. This added constraint of enforcing an order on the placement of values in the tree enables the matching algorithm to branch left or right based on the value extracted from the packet as in the case of binary search trees and eliminates the need for preprocessing of the packet field values.

CONCLUSION

This research shows how a firewall can be designed in Linux system, it is easy to implement on any other platform oriented system the reason being it is an open source and can be edited. It shows how a firewall in Linux can be implemented easily and gives the option of activating or deactivating the firewall. It also helps in knowing the unique features of Linux. To implement this there are two approaches that has been discussed in this paper. The early rejection filtering helps in optimising time consumed for matching the rejected packets. These denied packets can de rejected early in the process so that packets don’t have to go through the same process again. The second approach of using adaptive search trees for matching the traffic rules and making alphabetic tree which can be updated or reconstructed easily. This technique can be used for large traffic and give more optimal time.

The security of the system is one of the biggest concerns now days. With a growth in use of internet there is need for safe surfing which can be done with the help of firewall over a large network. In future firewall will also work like Intrusion Detection Device. It can be designed to 6share workload and can provide a strict security system.

REFERENCES

[1] E. Al-Shaer and H. Hamed. Discovery of policy anomalies in distributed firewalls. In *IEEE INFOCOM'04*, March 2004.
[2] F. Baboescu and G. Varghese. Scalable packet classification. In *ACM SIGCOMM'01*, 2001.
[3] F. Baboescu and G. Varghese. Fast and scalable conflict detection for packet classifiers. *Computer Networks*, 42(6), 2003.
[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.
[5] A. L. Edwards. *An Introduction to Linear Regression and Correlation*.W. H. Freeman and Co, San Francisco, 1993.
[6] A. El-Atawy, H. Hamed, and E. Al-Shaer. Adaptive statistical optimization techniques for firewall packet filtering. Technical Report CTI-TR- 05-019, DePaul University, 2005.
[7] A. Feldmann and S. Muthukrishnan. Tradeoffs for packet classification. In *IEEE INFOCOM'00*, March 2000.
[8] A. Garsia and M. Wachs. A new algorithm for minimum cost binary trees. *SIAM Journal on Computing*, 6(4):622–642, 1977.
[9] P. Gupta and N. McKeown. Algorithms for packet classification. *IEEE Network*, 15(2):24–32, 2001.
[10] P. Gupta and N. McKeown. Packet classification using hierarchical intelligent cuttings. In *Interconnects VII*, August 1999.
[10] **"**Firewall Design Techniques and its Development in Linux System**"** Prof. Vinit A. Sinha *Assistant Professor (MCA Department) PRMIT&R, Badnera Amravati (M.S), India*.
[11] H. Hamed, E. Al-Shaer, and W. Marrero. Modeling and verification of IPSec and VPN security policies. In *IEEE ICNP'05*, Nov. 2005.
[12] K. Lan and J. Heidemann. On the correlation of internet flow characteristics. Technical Report ISI-TR-574, USC/ISI, 2003.
[13] A. J. McAulay and P. Francis. Fast routing table lookup using CAMs. In *IEEE INFOCOM'93*, March 1993.
[14] Passive Measurement and Analysis Project, National Laboratory for Applied Network Research. Auckland-VIII Traces. http://pma.nlanr.net/Special/auck8.html, December 2003.