ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

GENERALISATION ANOMALY FREE TRAFFIC AWARE ALGORITHM TO INCREASE FIREWALL PERFORMANCE

Anuradha Shenoy
M.Tech Student, Dept. of ISE
B.M.S College of Engineering
Bengaluru,India

Abstract

Firewall is an important element of Network Security. They have become an essential part not only in Organisation level but also in smaller networks such as Home Network. It is commonly implemented as Packet Filter. The packet filter firewall works by comparing both incoming and outgoing packets against a set of predefined rules called Access Control List (ACL).As every packet is compared against these rules, it is important to have these rules well organised for very good performance. The presented paper proposes an algorithm which analyses incoming traffic to dynamically reorder the rules without giving rise to generalisation anomaly.

Keywords - Firewall Policies; Singly Linked List; Set Theory;

INTRODUCTION

Firewall acts as a secure wall which protects a network from both unauthorised and malicious users. It can be implemented as a hardware, a software or combination of both [1].Firewall uses a set of rules defined by the System Administrator to filter the incoming and outgoing traffic. Due to the interdependency among the rules, Firewall Policy Management has become a challenging task.

The need to optimise the firewall performance comes from the fact that rule set can be considerably large and complex. Maintaining a large rule set is a very tedious job. Reordering or modifying such a rule set can lead to firewall anomaly. Also large rule set will increase the average packet comparison time against the rules. This paper proposes a method to reorder the firewall policies based on the traffic flow without giving rise to generalisation anomaly.

This paper is organised as follows. Section II discusses various related works in this area. Section III describes the Firewall Policies Format and Section IV explains about different Firewall Anomalies. Section V introduces the Algorithm framework. Mathematical Proof is shown in Section VI. Conclusion and Future Work are given in Section VII.

RELATED WORK

A good amount of research has been carried out in optimising the firewall performance. [3] Proposes a method where firewall performance is increased using the traffic flow. Here some calculated statistics is used to adjust to the present traffic conditions by dynamically optimizing the ordering of the rules in the firewall. However, this method does not take previous traffic flow into consideration.

In [4] a "Firewall compressor" algorithm is proposed in which the overall size of the firewall policy is reduced by analysis. This is achieved by analysing the original rules and combining them to produce fewer rules i.e. all the redundant rules are removed from the firewall policy.

According to [5], the firewall optimization problem is to reorder the rules in such a way that the most frequently hit rules are near the top of the rule set. As a result of this the performance of the firewall increases. To achieve this rules are associated with a weight that equals the number of matches of these rules for current flow of traffic. Rule dependencies are taken into consideration while reordering the rules. The algorithm initially operates on not optimized rule set which is free of redundancy anomaly. Each rule is associated with a weight equal to proportion of packet matches. This initial list is used to create a heap which contains rules sorted in the order of rule weight ignoring any rule dependencies if exists. Another list is created which is initially empty and fills it with rules as the algorithm executes. The algorithm executes as long as there are rules in the heap that need to be processed and when it finishes executing it contains the optimized rule set.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

FIREWALL TYPES

Firewalls can be divided into different types based on the where the communication is being traced, where the communication gets intercepted and the state is being traced.

Network layer or packet filters

Here a set of rules in established by the System Administrator. For packets matching the rules appropriate actions are taken or default rule may be applied.

Application Layer Firewall

Here the firewall works on the application level of the TCP/IP stack. It intercepts all traffic that are travelling to or from the application.

Proxies

A proxy server can act as a firewall by responding to input packets in the manner of an application while blocking other packets.

Network Address Translation

Firewalls may have network address translation (NAT) functionality and the hosts protected behind a firewall commonly have addresses in the private address range. Firewalls here hide the true address of protected hosts.

FIREWALL POLICIES

Firewall Policies are set of rules which will help to secure the network. Testing a packet in a firewall means the header of the packet is tested against these rules. On finding a match corresponding Action to accept or deny is performed. The rules are stored in the rule set in the following format,

<Order><Protocol><source_IP><source_port><dest_IP><dest_port><Action>

Order here is the number at which the rule is stored. Protocol specifies the transport layer protocol present in the packet. Source_IP and dest_IP specifies the Source and Destination of the packet. Source_port and dest_port specifies the Source and Destination Port of the packet. The Action field takes the value Accept or Deny. When a packet matches all the field of a rule the Action corresponding to that rule is taken. If the packet does not match any of the predefined rule then default action will be to deny it.

FIREWALL ANOMALIES

Firewall can have thousands of rules. Maintaining such a firewall is a tedious job. Updating such a rule set may generate erroneous set of rules which are unable to perform their intended job. These errors in the rule set are called anomalies that have to be detected and removed from rule set for the efficient working of any firewall. Till date, five types of anomalies are discovered and studied namely, Shadowing Anomalies, Correlation Anomalies, Generalization Anomalies, Redundancy Anomalies, and Irrelevance Anomalies.

Shadowing Anomaly

Shadowing Anomaly occurs when the rule which comes first in the rule set matches all the packets and the rule which placed after the first rule in the rule set does not get chance to match the packets because the first rule has matched all the packets. It is a very critical problem since the rule coming later to the previous rule will never get activated.

Correlation Anomaly

Correlation Anomaly occurs if two rules match some common set of packets i.e. rule one matches some packets which are also matched by rule two and the action performed by both the rules is different. As the actions performed by these rules are different these rules cannot be reordered.

Generalisation Anomaly

Generalisation Anomaly occurs if the first rules matches all the packets which can be also matched by the second rule but the action performed is different in both the rules. The rule, which comes later in the rule list, is covered by the previous rule and also it has no effect on incoming packets. The super set rule is called General rule and the subset rule is called Specific rule. If the generalization relation exists then the superset rule cannot be placed before the subset rule.

Redundancy anomaly

Redundancy anomaly occurs if both of them matches some packets and the action performed is also the same. Removing one of the redundant rule in the firewall policy will not change the policy. Redundant rules can become bottleneck for the throughput. As every packet gets checked against these rules, a redundant rule will cause an additional redundant check. The System Administrator should check for such redundancy and remove such rules.

Irrelevance Anomaly

Irrelevant anomaly occurs if for a given time interval it does not matches any of the packets either incoming or outgoing. A firewall policy list can be very huge. As there are no traffic which matches this rule identifying and removal of such rule will

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

increase the performance. There is no such clear rule for the removal of this anomaly.

ALGORITHM FRAMEWORK

Here Access Control List is implemented as Singly Linked List even though theoretically height balanced tree structure will give better result [2]. Each node in the Linked List corresponds to a rule in Access List. Every incoming packet is compared against these rules, on finding a match appropriate actions are taken. The aim of this paper is to have a minimal comparison thereby maximising the performance of firewall. This is achieved here by reordering the rules based on traffic flow.

This paper defines a traffic aware algorithm for firewall policy check. It compares each packet against the rules and keeps track of the frequency match of each rule. After comparing threshold number of packets the rules are reordered.

REORDER function arranges the rules initially in the descending order of their frequency match and then checks for the generalisation anomaly.

RearrangeRules function arranges the rules in the descending order of their traffic hit percentage.

SUBSET function first generates a subset list for the given rule. The subset list is such that it succeeds the rule and performs a complimentary action. From the subset list generated it checks for subset super set relation. If such rules exists *SUBSET* function is again applied. All the rules from the subset list is removed and placed before the rule.

FindMatch function that compares the attributes of the incoming Packet *P* with the attributes of Rule *R*.

DoAction function performs the action corresponding to Rule

Update function keeps track of the recent traffic characteristics

The following is the structure of a Packet and Rule

Struct *Packet* {
Source port number
Source IP address
Destination port number
Destination IP address
}

Struct *Rule* {
Source IP address range
Wildcard Mask
Action
}

The following is the algorithm framework

SUBSET (Rule *R*, List *L*)
1: Generate List *L1* such that it contains all the Rules that are in
L which are subset of *R* and succeeding *R* and Actions
Performed by them are complimentary to each other
2: **for all** Rule *R1* in the List *L1* **do**
3: *SUBSET* (*R1*, *L1*)
4: **end for**
5: Remove the all the Rules in *L1* from *L*
6: Place all the rules before the Rule *R*

Figure 1. The algorithm to find the subset

REORDER (List *L*)
1: *RearrangeRules* (List *L*);
2: **for all** Rule *R* in List *L* **do**
3: *SUBSET*(*R*, *L*)
4: **end for**

Figure 2. The algorithm to reorder the rules

TrafficAware(Packet *Pi*)
1: **for all** Rule *R* in List *L* **do**
2: *FindMatch* (*Pi*, *R*)
3: **if** Rule *R* matches with Packet *Pi* **then**
4: *Do-Action* (*R*)
5: **break**
6: **else**
7: **continue**
8: **end if**
9: **end for**
10: **if** *FLAG* is equal to 0 **then**
11: *Do-Action* (Deny)
12: **end if**
13: Increment *count* by 1
14: Call *UPDATE* function
15: **if** *count* equals *threshold* **then**
16: *REORDER* (*L*)
17: *Count*=0
18: **end if**
19: *FLAG*=0;

Figure 3: TrafficAware algorithm

MATHEMATICAL PROOF

The *REORDER* and *SUBSET* function helps to remove the generalisation anomaly. This anomaly occurs when there exists subset superset relation between the rules. Reordering such rule set should

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785
(ISRA), Impact Factor: 2.114

guarantee superset rule is not present before the subset rule. Proving *SUBSET* and *REORDER* function will remove generalisation anomaly by using Set Theory.

Here the firewall rules are represented in the form of set. The elements of the set are the IP Addresses matched by the rule. Assumption done here is that no redundancy anomaly exists. Consider the following example set

Rule 1. {b}
Rule 2. {b,c}
Rule 3. {a,b,c,d}
Rule 4. {e,f}

This rule set is free of any anomaly. *TrafficAware* algorithm will finds the count of each rule. Rule 3 has hit percentage 15%, hit percentage of Rule 2 is 10%, Rule 1 is 5% and that of Rule 4 is 70% for current traffic flow.

After applying *RearrangeRules*, the rules will be reordered in the following way.

Rule 1. {e,f}
Rule 2. {a,b,c,d}
Rule 3. {b,c}
Rule 4. {b}

Rule 2 is the super set of the Rule 3 and Rule 4. Due to this rearrangement of the rules according to traffic flow the traffic may get blocked instead of allowing it or vice versa. *SUBSET* function is called to identify and reorder such Rule set.

SUBSET function first finds the subset of the superset Rule. *SUBSET* of Rule 1 are Rule 2 and Rule 3.For each rule in the subset if there exists subset superset relation *SUBSET* function will be called. Here allowing it or vice versa. *SUBSET* function is called to identify and reorder such Rule set.

The Rule 2 which is the superset of Rule 3 and Rule 4 is preceding it. As a result of this Rule 3 and Rule 4 will never be Rule 4 is the subset of Rule 3.Hence after applying the algorithm *SUBSET* recursively the subset of a rule will be removed from the list and placed before the superset. The rule set after applying the *SUBSET* function

Rule 1. {e,f}
Rule 2. {b}
Rule 3. {b,c}
Rule 4. {a,b,c}

CONCLUSION AND FUTURE WORK

There is a strong need to increase the performance of the firewall as it is a barrier between the internal and external network. Lot of research has been carried out in this area and this paper attempts to add to this. In this paper, a proposal to dynamically reorder the firewall policies is shown. The results shown indicate a considerable improve in the performance. Generalisation Anomaly is handled in this paper. As a future work Irrelevance Anomaly needs to be handled. Also in the current paper no past information of traffic flow is analysed.

REFERENCES

[1] H. Ling-Fang. "The Firewall Technology Study of Network Perimeter Security." In Proceedings of the IEEE Asia-Pacific Services Computing Conference, 2012, pp. 410- 413.

[2] Sudarsan, A. Vasu, A. Ganesh, D. Ramalingam and V. Gokul. "Performance Evaluation of Data Structures in implementing Access Control Lists." International Journal of Computer Networks and Security, vol. 24, issue 2, pp. 1303-1308, 2014.

[3] H. Hamed, A. El-Atawy & E. Al-Shaer. "On Dynamic Optimization of Packet Matching in High-Speed Firewalls." IEEE Journal on Selected Areas in Communications, vol. 24, issue 10, pp. 1817-1830, 2006.

[4] A.X. Liu, E. Torng, and C. R. Meiners. "Firewall compressor: An algorithm for minimizing firewall policies." In Proceedings of the 27th Conference on Computer Communications, 2008, pp. 176–180.

[5] Mothersole and M. Reed. "Optimizing Rule Order for a Packet Filtering Firewall." In Proceedings of the Conference on Network and Information Systems Security (SAR-SSI), 2011, pp. 1-6.