Firewall Log Analysis and Dynamic Rule Re-ordering in Firewall Policy Anomaly Management Framework

Lubna K, Robin Cyiac, Kavitha Karun A
Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology
Kochi-39, Kerala, India.
lubzz11@gmail.com

***Abstract*— Today, there are more many ways to communicate than there were just a few years ago and among them, internet plays a major role. Firewalls are essential for a secure network communication to ensure that only trusted packets are transferred between the private and public network. In firewall, security policy is implemented based on the rules defined by the network administrator; that decides which packets can be allowed to an organization's private network. Manual definition of rules often results in anomalies in the policy. Therefore, an effective anomaly detection and resolution approach is needed. After resolving these conflicts, the rules can be re-ordered dynamically that improves the efficiency of the anomaly management framework. With firewall log analysis, frequently used rules can be set as primitive rules, to which more security can be added.**

***Keywords*—policy anomaly; rule re-ordering; firewall logs; association rule mining.**

I. Introduction

Firewall is the de facto core technology of today's network security. Firewalls are essential for a secure network communication to ensure that only trusted packets are transferred between the private and public network by filtering the network traffic based on the firewall policy defined for that specific firewall. Firewall policy is a set of firewall rules defined by a system administrator and is of the form <condition, action>. Condition is the criteria with which a packet is matched. Action is that when a packet satisfies a condition, then it has to take some decision whether that packet can be allowed or denied by the firewall.

The quality of policy configured in the firewall determines the effectiveness of security protection provided by a firewall. The complex firewall configurations as well as the lack of tools and systematic analysis can cause errors while designing and managing firewall policies. Also, for updating legacy rules in large enterprises, there may be more than one administrator. When a packet matches with more than one rule in a firewall policy, we can say that, there is some anomaly in firewall policy and, if the matching rules have different action, then that is a conflict. Normally, firewalls implement first-match resolution mechanism. That is, when a packet matches with more than one rule; it takes the first matching rule. In order to increase the effectiveness of firewall security, it is crucial to have policy management techniques and tools that users can use to examine, refine and verify the correctness of firewall rules.

The practical approach to resolve firewall policy anomaly is to re-order the firewall rules. Here, Dynamic rule re-ordering approach is used [6] along with firewall log analysis. This work is motivated by some of the important Internet flow properties that are observed as a result of the traffic analysis which was performed on several Internet packet traces collected at the edge routers of DePaul University and University of Auckland networks [4]. After studying the statistics of these traffic traces, it is observed that the following properties are pertaining to Internet flows [6]:

Skewed flow size: Mainly, Internet traffic has few heavy-weight flows. So for packet filtering, it is advisable to reduce the number of packet matches that is needed for heavy-weight flows in order to decrease the overall packet matching time.

Skewed flow durations: Mainly, Internet traffic has few long-lived flows. So in order to improve the packet filtering process, it is advisable to reduce the number of packet matches needed for long-lived flows.

In general, most of the packets of a flow match the same filtering rules, since these observations clearly indicate that a small number of the firewall policy rules are used for matching a significant portion of the traffic over a considerable amount of time. This emphasizes that the idea of considering the contribution of rule matching in filtering policy optimization is useful and practical for improving the overall matching performance in firewalls.

This paper is organised as follows: Section II overviews different anomalies in firewall policies. Section III describes about the system design and architecture. Section IV gives the implementation details. Section V discusses about the result obtained. The paper is concluded in Section VI.

II. Firewall Policy Anomalies

The firewall policy consists of sequence of rules that define the actions performed on packets so that, it satisfy certain conditions. A rule consists of certain conditions that perform some actions. The typical firewall policy anomalies [3] are: *Generalization:* A rule is a generalization of one or more of

978-1-4673-6126-2/13/$31.00 ©2013 IEEE

preceding rules if they have different actions and if a subset of packets matched by this rule also matches the preceding rules. *Shadowing:* A rule is said to be shadowed when, one or more of preceding rules that matches all the packets matched by this rule, in such a way that the shadowed rule is never activated. Shadowing can be considered as a critical error in the policy, because the shadowed rule never takes effect. *Correlation:* If a rule intersects with other rules but have different action, then this rule is said to be correlated with other rules. Here, the packets matched by the intersection of those rules may be denied by one rule, but permitted by others. *Redundancy:* A rule is redundant if there is another same or more general rule available that has same action on the same packet such that if the redundant rule is removed, the overall firewall policy will not be affected.

III. System Design and Architecture

A simple framework for anomaly detection and resolution is designed. The whole framework can be divided into two: an administrator end and a user end. System architecture is shown in figure 1. The administrator has to authenticate before performing any actions. After login, administrator have options for rule generation, anomaly detection and resolution and, rule re-ordering which is done dynamically analyzing the firewall log. The end user part selects the destination IP and destination port to which it has to send a file. And that is checked at the rule engine, with the defined firewall policy whether the transaction can be allowed or denied. Based on the user's transaction, firewall log is generated and the most frequently used firewall rules are mined that can be set as primitive rules.

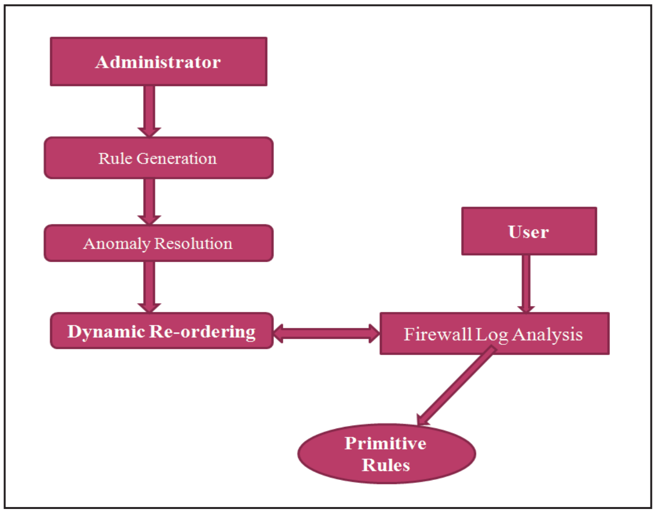

Figure 1. System Architecture.

A. *Rule Generation*

Rules are generated by system administrator with the fields: protocol, source IP, destination IP, source port, destination port and action. If there is more than one rule with the same fields, then we can say there exist some anomaly. Anomalies like shadowing, generalization, correlation and redundancy are detected comparing each rules. For each rule, a risk value [1] is allocated by the system administrator. Risk value is calculated based on a vulnerability scoring system, CVSS (Common Vulnerability Scoring System) [8]. The Common Vulnerability Scoring System (CVSS) provides an open framework for communicating the characteristics and impacts of vulnerabilities. CVSS consists of 3 groups: Base, Temporal and Environmental. . The Base group reflects the intrinsic qualities of vulnerability. The Temporal group represents the characteristics of vulnerability that change over time. The Environmental group reflects the characteristics of vulnerability that are unique to any user's environment.

Risk Value = CVSS BaseScore * Importance Value.

Here CVSS BaseScore can be calculated using the standard equation defined in [8]. Importance value is defined for each rule by the system administrator.

B. *Anomaly Detection and Resolution*

Anomalies can be detected by comparing each rule; that is, we have to check whether there is matching field values for more than one rule. If the matching rules have different action that is, for one rule it is allow and for other rule it is deny, then they are said to be conflicting rules. When conflicting rules are detected, that has to be resolved by selecting the rule with minimum risk value.

Algorithm 1: Dynamic Rule Re-ordering

Input: Rule set R, Packet set P

```
begin
  initialize Num: = x (set limit)
  for each i=0 to R do
    Pi interrogate with Ri;
    if Pi matches ri >= x then
      Reorder Ri
    else If Pi matches ri < x then
      Ri can’t be reordered.
    end if
  end for
end
```

End user part is shown in figure 2. In the end user part, there is an option for file transfer. While transferring, in the rule engine, it is checked whether the firewall rule matched with the specified source IP, source port, destination IP and, destination port; permits transaction or denies transaction. If the file to be transferred has its destination and source fields matched with the firewall rule that denies transaction, then that file cannot be send. Else if the file to be transferred has its destination and source fields matched with the firewall rule that allows transaction, then that file can be send.

C. *Firewall Log Analysis*

Firewall log analysis would generate a set of primitive rules with repeated and rare outcomes, which can be used to add more security for frequent rule in firewall log. Purpose is to mine the firewall log data belonging to dynamical database accumulating as firewall system in operation. For e.g.:- Particularly, for corporate large network, firewall system would generates millions of log data every day, and further, the need of dealing with large amount of logs certainly imposes additional work burden on policy management system.

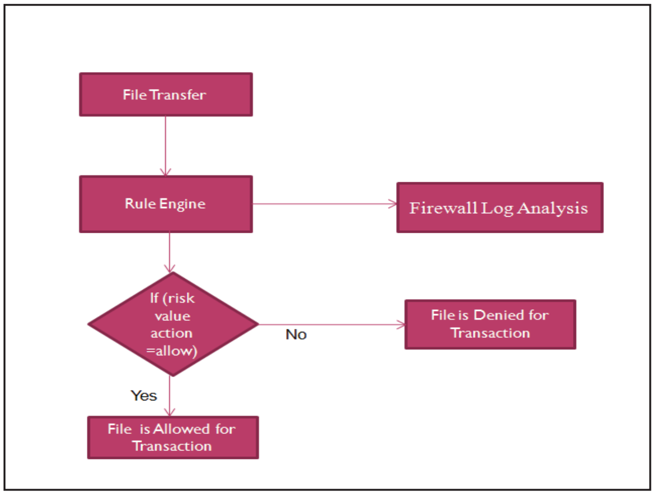
File Transfer
Rule Engine
Firewall Log Analysis
If (risk value action =allow)
No
File is Denied for Transaction
Yes
File is Allowed for Transaction

Figure 2. End User Aspect.

Here, frequently used rules are selected using a concept of association rule mining [5]. Traditionally, association rule mining (ARM) is defined as: Given a database of transactions, a minimal confidence threshold and a minimal support threshold, the goal is to find all association rules whose confidence and support are above the corresponding thresholds. In ARM, an association rule is of the form X => Y. X and Y is disjoint conjunctions of attribute-value pairs. ARM generates the largest item-set and the best rules from log dataset. Association rules are generated using apriori algorithm. The Apriori Algorithm is used for mining frequent itemsets for boolean association rules. Problem Statement for apriori is to find the frequent itemsets. Frequent itemsets are the sets of item which has minimum support. Apriori Property is that any subset of frequent itemset must be frequent.

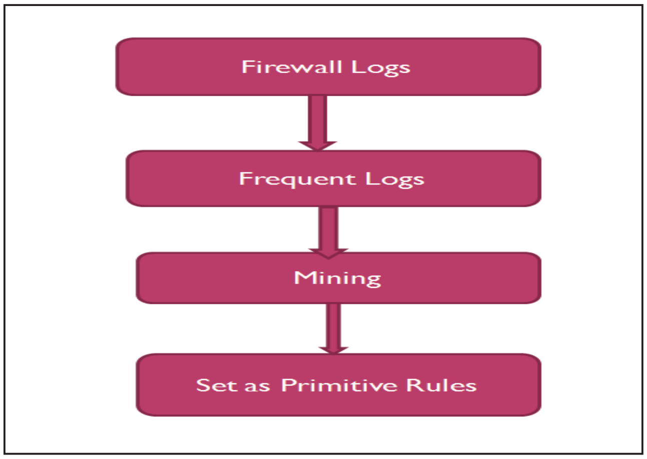

Figure 3. Firewall Log Analysis.

Figure 3 shows how firewall log analysis is implemented here in this anomaly management framework. Here, from the firewall logs, frequent logs are mined using a concept of association rule mining. That is, it will filter the frequently used rules from the whole firewall log that can be set as primitive rules to which more security can be added.

IV. Implementation and Discussion

A simple framework for firewall policy anomaly management is implemented by incorporating the concept of dynamic rule re-ordering and firewall log analysis. This is implemented in Java. And the minimum hardware requirement is Pentium IV processor with a speed of 3GHz, 1GB RAM and 80GB hard disk.

Challenges in implementation are: For dynamic re-ordering of firewall rules, there is a need to access firewall logs so, direct re-ordering is not possible. Hence, if there is no user log while running the dynamic rule re-ordering algorithm, then no CPU usage will be displayed. Also, when the firewall log is very large, it may take some more time than expected. Particularly, for large corporate network, firewall system would generate millions of log data per day. So the need for dealing with these huge logs certainly imposes additional work burden on policy management framework. But that can be solved by extracting only frequent logs with association rule mining.

V. Result and Analysis

Firewall policy anomaly management is a complex task because of the large number of interacting rules in a firewall and is also error prone since rule generation and updation are done manually. The effectiveness of security protection provided by a firewall mainly depends on the quality of policy configured in the firewall. Unfortunately, designing and handling firewall policies are often error prone due to the complex nature of firewall configurations as well as the lack of systematic analysis mechanisms and tools. Also, more than one administrator is responsible for updating legacy rules in large enterprises. Therefore, effective mechanisms and tools for policy management are crucial to the success of firewalls. There exist many efficient anomaly management tools like FAME [1], Fireman [2], and Firewall Policy Advisor [7].

A practical conflict resolution approach is firewall rule re-ordering. Existing systems use a permutation algorithm to find an optimal solution that extensively finds the permutations of conflicting rules. This algorithm exhaustively computes the resolving score of conflicted rules for all permutation. However, the fundamental limitation of using this algorithm lies in computational complexity which is O (n!). Hence it is complex and time consuming.

Normal sorting technique can be used when firewall policy anomalies are resolved prior to rule re-ordering and that re-orders the whole firewall rule in ascending order. If the firewall policy consists of large number of rules, then the re-ordering time will be more for normal sorting process. With dynamic rule re-ordering, we can see that the re-ordering time is reduced since that uses firewall logs and re-order the frequently used rules only. Also with firewall log analysis, there is an option to select the frequently used rules that can be set as primitive rule, to which more security can be added.

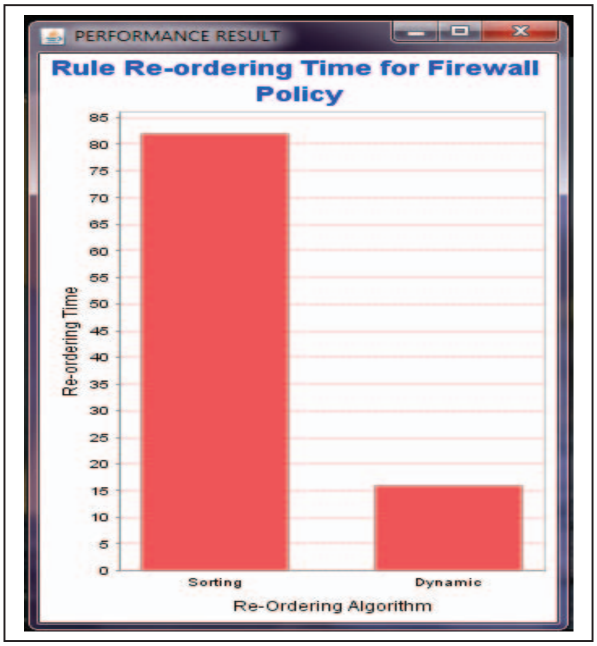

Figure 4. Rule Re-ordering Time for Firewall Policy

VI. Conclusion

We all know that Firewalls are essential for a secure network communication to ensure that only trusted packets are transferred between the private and public network. It implements the security policy based on the rules defined by the network administrator; that decides which packets can be allowed to an organization's private network. Manual definition of firewall rules often results in policy anomalies. Therefore, an approach to improve the firewall policy anomaly management is implemented. That is, after resolving these conflicts, the rules can be re-ordered dynamically that can improve the efficiency of the anomaly management framework. Also, with firewall log analysis, rules that needs more security can be set as primitive rules.

References

[1] H. Hu, G. J. Ahn, K. Kulkarni, "Detecting and Resolving Firewall Policy Anomalies", IEEE Transactions on Dependable and Secure Computing, 2012.

[2] L. Yuan, H. Chen, J. Mai, C. Chuah, Z. Su, P. Mohapatra, and C. Davis, "Fireman: A Toolkit for Firewall Modeling and Analysis", Proc. IEEE Symp. Security and Privacy, May 2006

[3] E. Al-Shaer and H. Hamed, "Discovery of Policy Anomalies in Distributed Firewalls", Proc. IEEE INFOCOM, 2004.

[4] Passive Measurement and Analysis Project, National Laboratory for Applied Network Research. Auckland-VIII Traces. http://pma.nlanr.net/Special/ auck8.html", December 2003.

[5] Agrawal, Rakesh, T. Imieliński and A. Swami, "Mining Association Rules Between Sets of Items in Large Database", Proc. ACMSIGMOD 1993.

[6] H.Hamed and E.Al-shaer, "Dynamic Rule-ordering Optimization for High-speed FirewallFiltering", ASIACCS'06, March 2006.

[7] E. Al-Shaer and H. Hamed. "Design and Implementation of Firewall Policy Advisor Tools", School of Computer Science Telecommunications and Information Systems, Aug 2002.

[8] CVSS Scoring System,"http://nvd.nist.gov/cvss.cfm.