Methods for Firewall Policy Detection and Prevention

Hemkumar D
Asst Professor
Dept. of Computer science and Engineering
Sharda University, Greater Noida NCR

Mohit Chugh
B.tech (Information Technology)
Sharda University,Greater Noida, NCR

ABSTRACT

Firewall plays an important role in network security, acting as an interface between networks.It implements the policy of the network by deciding on which packets are allowed through based on rules as defined in Firewalls. Any error in definition of the rules, may affect the system security by letting unwanted traffic to pass or blocking desired traffic. However, managing rules for firewall, mainly for enterprise networks, has become complex and more error-prone.Firewall rules for filtering harmful packets have to be written, ordered and distributed carefully in order to avoid firewall policy anomalies that might cause network vulnerability. Therefore, inserting or modifying filtering rules in any firewall requires thorough externally and internally analysis to determine the appropriate rule placement and ordering in the firewalls. In this paper, we have adopted a rule-based segmentation technique to identify policy anomalies and additionally, we present a new algorithm that will resolve any anomaly present in the policy rules to generate a new anomaly free rule set.In addition, we demonstrate Policy anomaly management framework on how efficiently our approach can discover and resolve anomalies in firewall policies rigorously.

Key Terms:- Firewall, policy anomaly management,

1. INTRODUCTION

Network security is essential to the development of internet and has attracted much attention in research communities. Firewalls have been the frontier defense for secure networks against attacks and unauthorized traffic by filtering out unwanted network packets coming from or going to the secured network.

A firewall normally acts as an interface of a network to one or more external networks and keeps on regulating the network traffic passing through it. It also confirms which packets to allow to go through or to drop based on a set of "rules" defined by its administrator[2]. These rules have to be defined first and maintained with extra care because any small mistake the defining of rules may allow unwanted traffic to be able to enter or leave the network

Rules defined for Firewall are usually in the form of a criteria and an action to take if any packet matches the criteria. These actions are usually *accepted* and *reject* [1]. Firstly, a packet arriving at a firewall is tested with each rule sequentially. Secondly, whenever it matches with the criteria of a rule, the action specified in the rule is executed and further rules are skipped. Because of this only, firewall rules are order sensitive. Whenever a packet matches with more than a rule, the first rule is executed.

As the number of filtering rules increases and the policy becomes much more complex to resolve [2]. Firewall policy visualization is an effective solution to this policy management, that helps users to understand their policieseasily. It also helps users to find out complicated rule patterns and behaviors efficiently.

ISSN: 2278 – 7798

2. RELATED WORK

As it is well seen, the process of configuring a firewall is tedious and error prone. That is why, effective mechanisms and tools for policymanagement is very important to the success of firewalls.

In few recent years, policy anomaly detection has received a great deal of attention. Hence, policy analysis tools, such as Firewall Policy Advisor (FPA) and FIREMAN [2], with an objective of detecting policy anomalies have been introduced. FPA is only responsible of detecting pair wise anomalies in firewall rules whereas FIREMAN can detect anomalies in terms of multiple rules by analyzing the relationships between one rule and the collections of packet spaces derived from all previous rules. However, FIREMAN also has limitations in detecting anomalies. For each firewall rule, FIREMAN only examines all preceding rules but ignores all subsequent rules when performing anomaly analysis.

Bartal et al,also presented a firewall management toolkit termed as Firmato, attaining significant improvement towards managing the firewall in complex and multi firewall environment. Similarly, Wool et al. introduced another firewall analysis, which act upon modified queries lying on a place of filtering rules & extracts the more linked rules in the firewall security policy. These two tools tends to manage firewalls in a very complex environment

3. METHODOLOGY

The firewall policies contains sequencing of rules that define the actions being performed on data packets so that, it satisfy certain conditions [3]. A rule consists of different conditions that always perform some actions. Table given above illustrates an example of firewall policy [1] including 5 firewall rules- r1, r2, r3, r4, and r5.

The common anomalies for firewall policy are:

Rule	Protocol	Source IP	Source Port	Dest IP	Dest Port	Action
R1	UDP	110.12.3.*	*	170.40.1.*	93	Deny
R2	UDP	110.12.*.*	*	170.40.1.*	93	Deny
R3	TCP	110.12.*.*	*	170.48.*.*	42	Allow
R4	TCP	110.12.2.*	*	170.48.1.*	42	Deny
R5	TCP	110.12.2.*	*	*	*	allow

Table 1 -Overview of Possible Firewall Policy Anomalies between 2 rules.

I. Shadowing: A rule is said to be shadowed when one or more preceding rules that matches all the packets matched by a single rule, and therefore the shadowed rule is never activated. For example, r4 is shadowed by r3 in Table given above as an example. In r3 packets from tcp protocol with source IP (110.12.*.*) are accepted but in r4 the same packets with Source IP (110.12.2.*) is denied. Since, r4 is shadowed by r3 and r4 will be ignored by accepting packets with all Source IP.

II. Generalization: A rule is said to be generalized of one or more of preceding rules, if they have varying actions and if a subset of those packets matched by this rule and also matches with the preceding rules. For example, r5 is a generalization of r4 in Table 1.

III. Correlation: If a rule intersects with rules but have different action, then this rule is said to be correlated with other rules. For example, r2 is in correlation with r5 in Table 1. The two rules with this ordering imply that all UDP packets coming from any port of 170.40.1.* to the port 93 of 162.32.1.* match the intersection of these. Since, r2 precedes r5, every packet within the intersection of r5 will be denied by r2.

IV. 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. For example, r1 is redundant to r2 in Table 1, since all UDP packets coming from any port of 100.11.2.* to the port 80 of 162.32.1.* matched with r1 can match r2 as well with the same action.

ISSN: 2278 – 7798

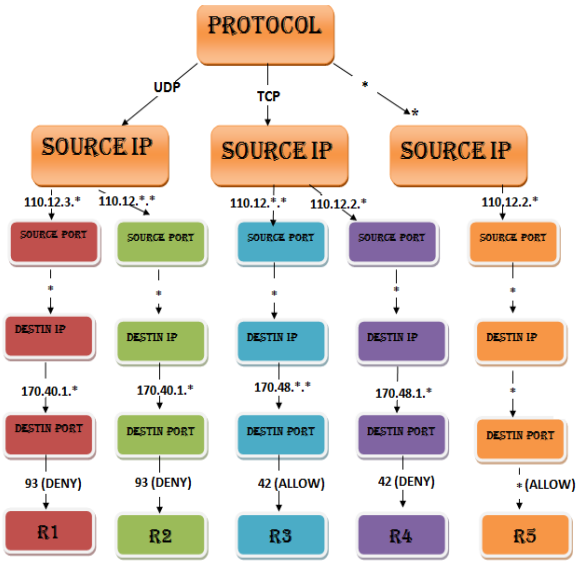
V. Irrelevance Anomaly: If a rule cannot match to any traffic that might flow through the network, then the rule is called irrelevant. This happenswhen the source address and destination address fields of the rule do not match any domainreachable through this firewall.

4.Firewall Policy Anomalies Representation

Firewall policy representation can be categorized in two ways - policy tree representation and packet space segmentation.

Policy Tree: Policy Tree provides simple representation of the filtering rules & helps to find out the relations among these rules.

In policy tree, Firstly, the root node represents the protocol field [3]. Secondly, the leaf node represents the action field and finally, the intermediate nodes represent the rest of fields in proper order. An appropriate rule representation is specified from the root to a leaf vice versa in a policy tree. Every rule must have an action leaf that represents the action (accept or deny) of the rule. The last box below the leaf represents the rules that are in anomaly with it.



Packet Space Segmentation: In order to identify policy anomalies more efficiently, we adopt another technique i.e"**rule-based segmentation technique"** [4] [5]. In this technique, firstly, a network packet space is defined by a firewall policy and that further can be divided into a set of disjoint packet space segments. Each segment associated with a unique set of firewall rules, accurately results an overlap relation among those firewall rules.

The algorithm used below is categorized for Anomaly discovery in our paper.

Algorithm: Segment Generation for a Network Packet (Anomaly Discovery)

Space of a Set of Rule R: Partition (R)

Input: A set of rules, R

Output: A set of packet space segments, S.

```
1. for each r ∈ R do
2.         sr← PacketSpace(r);
3.         for each s ∈ S do
4.                 /* sr is a subset of s */
5.                 ifsr⊂ s then
6.                         S.Append( s \ sr )
7.                         s← sr;
8.                         break;
9.                 /* sr is a superset of s */
10.                elseifsr⊃ s then
11.                        sr← sr\ s;
12.                /* sr partially matches s */
13.                elseifsr ∩ s ≠ Φ then
14.                        S.Append( s \ sr )
15.                        sr ← sr ∩ s;
16.                        sr ← sr \ s;
17.                S.Append( sr );
18. return S;
```

A pair of packet spaces must satisfy one of the following relations: subset from line 5, superset from line 10, partial match from line 13, or disjoint from line 17. So that, one can use set operations to separate the overlapped spaces into disjoint spaces.
Fig 2(a) depicts the 2D geometric representation of firewall rules defined in Table example .Two spaces overlaps when the packets matching corresponding two rules intersects. Here the policy segments are classified as: overlapping and non-overlapping segments, which is further divided into conflicting overlapping and non-conflicting overlapping segments.
Fig, 2(b) illustrates the segments of packet spaces derived from the example policy in Table.
In Fig. 2(c), seven disjoint uniform segments are represented. Here s2, s4 and s7 are non-overlapping

ISSN: 2278 – 7798

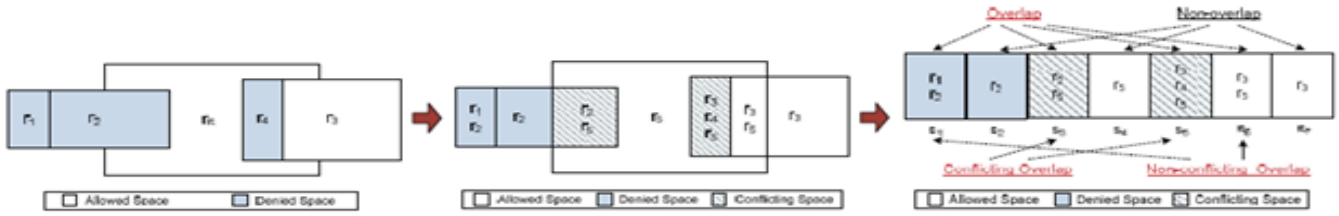


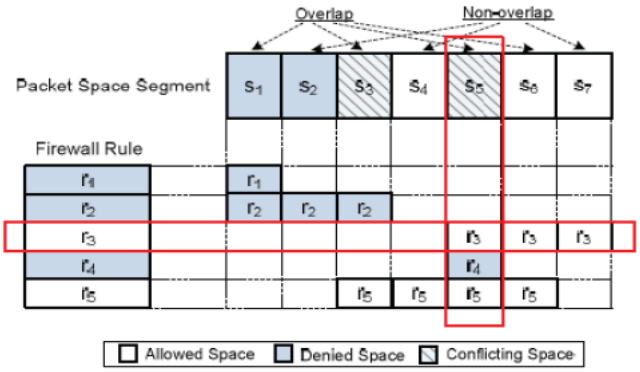
Fig 2(a) Fig 2(b) Fig 2(c)

segments and s1, s3, s5 and s6 are overlapping segments

5. GRID REPRESENTATION

It is still very difficult for the administrator to find out the policy anomalies, that is how many segments one rule is involved in. Therefore, Grid Representation was introduced to satisfy the need for more precise anomaly representation [4] ,that shows matrix-based visualization of policy anomalies, in which rules are represented along X axis , space segments are displayed along Y axis of matrix & the intersection of both displays a rule's subspace covered by that segment.

A grid representation of policy anomalies for the example policy in Table 1 is shown in Fig. 3. In this, a conflicting segment(CS), which has conflict, is related to a set of conflicting rules r3, r4, and r5 and a rule r3 is involved in segments s5, s6, and s7. That's why, this grid representation forms a better understanding and avoids some limitations of policy tree representation technique.



6. Resolving Anomalies of Firewall Polices

Here, we have shown a new algorithm to resolve Firewall Policy anomalies.This resolving algorithm maintains 2 universal lists variables – *old_rule_list* (containing rules of original firewall configuration) and *new_rule_list* (containing rules without anomaly).

Here, a rule is insert into *new_rule_list* individually from *old_rule_list*, keeping in mind only one objective to make them free from anomalies**.**Below is a RESOLVE ALGO, we've designed for resolving anomalies.

1. *old rules list* ← read rules from original firewall configured file
2. *new rules list* ← empty list
3. **for all** *r* ∈ *old rules list* **do**
4. **Insert(*r*, *new rules list*)**
5. **for all** *r* ∈ *new rules list* **do**
6. **for all***s* ∈ *new rules list* after *r* **do**
7. **if** *r*⊂ *s* **then**
8. **if** *r.action*= *s.action***then**
9. Remove *r* from *new rules list*
10. **break**

Algorithm Insert is responsible to insert a rule into the *new_rules_list*in such manner that the list remains anomaly free all the time.

1. **if***new rules list* is empty **then**
2. insert*r* into *new rules list*
3. **else**
4. *inserted*← *false*
5. for all *s* ∈ *new rules list* **do**
6 **if** *r* and *s* are not disjoint **then**
7. *inserted* ← Resolve(*r*, *s*)
8. **if***inserted* = *true* **then**
9. **break**
10. **if***inserted* = *false* **then**
11. Insert *r* into *new rules list*

ISSN: 2278 – 7798

7. PROPOSED ANOMALY-MANAGEMENT FRAMEWORK

Our proposed anomaly management framework is composed of two core methodologies (as in figure) based on rule based segmentationtechnique - conflict detection and resolution, and redundancy discovery and removal.

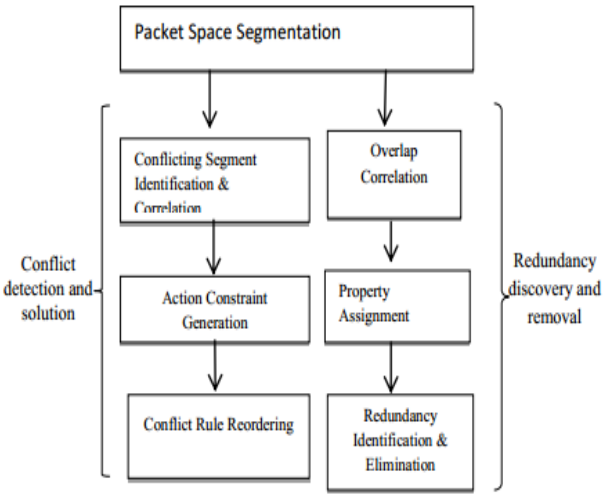


Fig 4 – Proposed Framework

For first methodology, conflicting segments are identified associating with a policy conflict and a set of conflicting rules and further, conflict correlation groups (CG) are derived. The second step provides a strategy based method for generating action constraint for each conflicting segment and in last, it illustrates a reordering algorithm, which is a combination of permutation and greedy algorithm to find out a near-optimal conflict resolution solution for all policy conflicts.Finally, redundant rules will be identified first and then gets eliminated.Once conflicts are identified,it finds which rule should take precedence when a single network packet is matched by a set of rules involved in the conflict.

Our conflict resolution mechanism assigned an action constraint (desired action i.e. either Allow or Deny) to each conflicting segment that thefirewall policy should obey when any packet within the conflicting segment comes to the firewall. Then, to resolve a conflict, we only guarantee that the action taken for each packet comes within the conflicting segment can satisfy the corresponding action constraint.

STRATEGY	ACTION CONSTRAINT
Deny Overrides	Action = "deny"
Allow Overrides	Action = "allow"
Recency Overrides	Action of the newest rule
Specificity	Action of the most specific rule
High-majority-overrides	Action of the rules with greater number
First-match	Action of the first-matched rule
High-authority-overrides	overrides Action of the rule with the highest

Table 2 - Generating Constraint from Conflict Resolution Strategy

A key characteristic of this mechanism is that we do not need to move a rule expected to take precedence to the first match rule all the time. Any rule associating with the conflict on the same action (as a rule with the precedence) can be moved to the first-match rule, guaranteeing the same effect with respect to the conflict resolution. Thus, this mechanism is effective to obtain an optimal solution for conflict resolution.

8. RESULTS

Evaluation of Conflicting Segment Generation and Correlation:-

Table 3 depicts the evaluation results generated by the segmentation and correlation engine of FAME. This includes the number of conflicting segments, conflict correlation groups, and the execution time required by the segmentation module of FAME for identifying detecting conflicts [3] [5]. From the table only, the number of large conflict correlation groups and the number of conflicting rules in the largest correlation group give us the evidences that manual conflict resolution for a large size of firewall policies is impossible, with no point to discuss.

POLICY	NO. OF RULES	SEGMENTATION		CORELATION		FIRST MATCH		PROPOSED	
		CS(#)	TIME(SEC)	CG(#)	TIME(SEC)	RC	TIME(SEC)	RC	TIME(SEC)
A	15	7	0.141	3	0.066	4	0.318	5	0.423
B	21	8	0.198	3	0.071	3	0.568	4	0.568
C	30	12	0.274	5	0.089	7	0.719	9	0.732
D	57	20	0.427	9	0.118	15	1.438	17	2.547
E	93	27	0.468	10	0.147	24	6.792	32	8.117
F	137	42	0.613	10	0.196	36	13.119	43	57.756
G	447	78	0.916	13	0.221	57	31.712	61	92.331

Also, we can conclude that the segmentation and correlation processes we discussed are efficient enough to handle a larger size of firewall policies, such as policy G in the table.

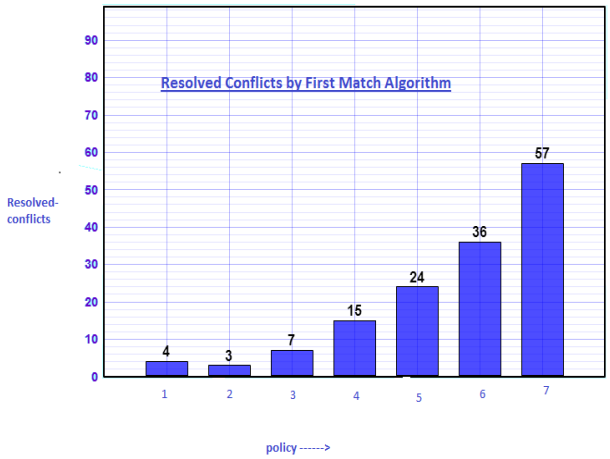
ISSN: 2278 – 7798

Evaluation of Conflicting Rule Reordering Algorithm :-

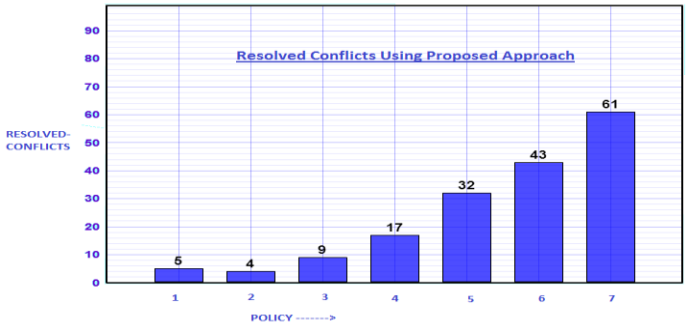
In order to evaluate our proposed method, we measured the effectiveness and efficiency of three algorithms implemented in the rule reordering module of FAME using two metrics[3], resolved conflicts (RC) and resolving time.

The permutation algorithm [2] illustrates to achieve an optimal conflict resolution for all policies except policy G. Because it shows that the resolving time required by the permutation algorithm increases exponentially as the number of conflicting segments increases. Hence, the permutation algorithm is not feasible to the policies with a large size of conflicting rules.

In contrast to greedy algorithm [2], it can only achieve a near-optimal conflict resolution for all firewall policies. If the size of conflicting rules increases, the time taken by the greedy algorithm also increases.



For the combination algorithm, the results in Table 3 illustrates that the number of resolved equal to the optimal solution achieved by the permutation algorithm alone. And it represents higher efficiency and effectiveness in conflict resolution.



9. Conclusion

It is concluded that the Firewall security requires proper management in order to provide proper security services and may not necessarily make the network any secure.

However to overcome from this issue , we demonstrated Policy anomaly management framework with a new algorithm to resolve Firewall Policy anomalies that facilitates systematic resolution of firewall policy anomalies and in the end of the paper, we illustrates that the proposed framework (combining two algorithms) has effective and efficient result as compared to others.

We believe that there isstill lot to do in firewall policy management area in developing anomaly analysis methodology more practical and helpful in forming a securable network management.

10. References

[1] Hongxin Hu, Gail-JoonAhn, “Detecting and Resolving Firewall Policy Anomalies”, 2012.

[2] Jitha C K, Sreekesh Namboodiri, “Firewall Policy Anomalies- Detection and Resolution”, 2013.

[3] Lubna, Robin Cyriac, “A Study on Firewall Policy Anomaly Representation Techniques”, 2013.

[4] Anbarasan.A, Balasubramani.G, Madhan.C, Naveenkumar.P, Mrs. N.S.Nithya, “Detecting and Resolving Firewall Policy Anomalies Using Rule-Based Segmentation”, 2013.

[5] B.Srikanth , Smt.K.Venkata Ramana, “Firewall Policy Anomaly Detection and Resolution Using Rule Based Approach”, 2013.

ISSN: 2278 – 7798