Efficiently Managing Firewall Conflicting Policies

[1]K.Raghavendra swamy, [2]B.Prashant
[1]Final M Tech Student, [2]Associate professor,
Dept of Computer Science and Engineering[12],
Eluru College of Engineeering and Technology, Eluru,W.G Dist,A.P[12].

Abstract: Firewalls are a widely deployed security mechanism to ensure the security of private networks in most businesses and institutions. The effectiveness of security protection provided by a firewall mainly depends on the quality of policy configured in the firewall. However, designing and managing 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. This paper represents an innovative anomaly management framework for firewalls, adopting a rule-based segmentation technique to identify policy anomalies and derive effective anomaly resolutions. PolicyVis presented in this paper provides visual views on firewall policies and rules which gives users a powerful means for inspecting firewall policies

Keywords: Firewall, Policy Anomaly Management, Visualization Tool

I. Introduction

Firewall

A **firewall** can either be software-based or hardware-based and is used to help keep a network secure. Its primary objective is to control the incoming and outgoing network traffic by analyzing the data packets and determining whether it should be allowed through or not, based on a predetermined rule set. A network's firewall builds a bridge between an internal network that is assumed to be secure and trusted, and another network, usually an external (inter)network, such as the Internet, that is not assumed to be secure and trusted.

Firewalls usually function as routers which connect different network segments together. It is simply a perimeter defense device splitting network environment into internal (trusted) and external (distrusted) network for controlling and filtering incoming and outgoing network traffic. Its packet filtering decision depends on a set of policy rules (also named policy rule table) describing the security policy and posture the corporation takes, and thus to effectively avoid suspicious intruder executing illegal actions and damaging internal network.

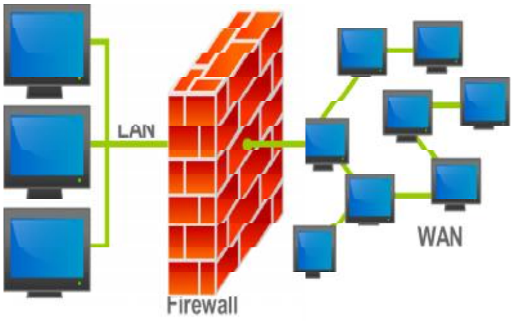


Figure 1: Firewall in Networks

Firewall Policy

A firewall is a network element that controls the traversal of packets across the boundaries of a secured network based on a specific security policy. A firewall security policy is a list of ordered filtering rules that define the actions performed on matching packets. Filtering actions are either to accept, which passes the packet into or from the secure network, or to deny, which causes the packet to be discarded.

To implement a security policy in a firewall, system administrators define a set of filtering rules that are derived from the organizational network security requirements.

Firewall policy management is a challenging task due to the complexity and interdependency of policy rules. This is further exacerbated by the continuous evolution of network environments. Therefore, effective mechanisms and tools for policy management are crucial to the success of firewalls.

The complex nature of policy anomalies, system administrators are often faced with a more challenging problem in resolving anomalies, in particular, resolving policy conflicts. An intuitive means for a system administrator to resolve policy conflicts is to remove all conflicts by modifying the conflicting rules. However, changing the conflicting rules is significantly difficult, even impossible, in practice from many aspects. First, the number of conflicts in a firewall is potentially large, since a firewall policy may consist of thousands of rules, which are often logically entangled with each other. Second, policy conflicts are often very complicated. One rule may conflict with multiple other rules, and one conflict may be associated with several rules. Besides, firewall policies deployed on a network are often maintained by more than one administrator, and an enterprise firewall may contain legacy rules that are designed by different administrators.

II. System Architecture

Firewall system needs a series of procedures, including log analysis, rule update, and configuration, to continuously maintain inner policy table for facilitating its security efficiency, hence, it is definitely a costly and error prone job for large networked organization.

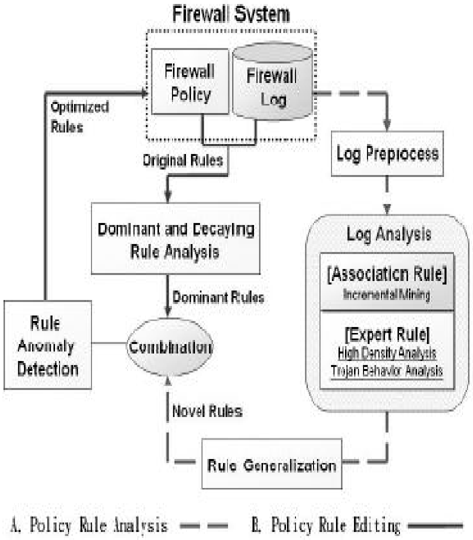
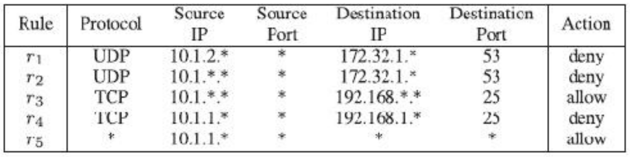


Figure 2: Firewall Policy Architecture



Rule	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
r_1	UDP	10.1.2.*	*	172.32.1.*	53	deny
r_2	UDP	10.1.*.*	*	172.32.1.*	53	deny
r_3	TCP	10.1.*.*	*	192.168.*.*	25	allow
r_4	TCP	10.1.1.*	*	192.168.1.*	25	deny
r_5	*	10.1.1.*	*	*	*	allow

In Rule Analysis procedure, the process consists of following three modules: log preprocess, log analysis, and rule generalization. Firewall log data would be firstly parsed into log preprocess module to extract primary attributes of log data: [Date], [Time], [Protocol], [Source IP] (Src_IP), [Source Port] (Src_Port), [Destination IP] (Dst_IP), [Destination Port] (Dst_Port), and [Action], simplifying raw firewall log data for facilitating the processing efficiency. Then, system will utilize our proposed log analysis methods to derive valuable traffic rules from preprocessing log data. After log analysis step, a collection of traffic rules would be generated, and then system would perform rule generalization to generalize a set of novel rules reflecting current environment from previous results.

Anomalies in Firewall Policies

Table 1: Firewall Policy Example

- Shadowing: A rule can be shadowed by one or a set of preceding rules that match all the packets which also match the shadowed rule, while they perform a different action. In this case, all the packets that one rule intends to deny (accept) can be accepted (denied) by previous rule(s), thus the shadowed rule will never be taken effect. In Table 1, r4 is shadowed by r3 because r3 allows every TCP packet coming from any port of 10.1.1.* to the port 25 of 192.168.1.*, which is supposed to be denied by r4.
- Generalization: A rule is a generalization of one or a set of previous rules if a subset of the packets matched by this rule is also matched by the preceding rule(s) but taking a different action. For example, r5 is a generalization of r4 in Table 1. These two rules indicate that all the packets from 10.1.1.* are allowed, except TCP packets from 10.1.1.* to the port 25 of 192.168.1.*. Note that, as we discussed earlier, generalization might not be an error.
- *Correlation*: One rule is correlated with other rules, if a rule intersects with others but defines a different action. In this case, the packets matched by the intersection of those rules may be permitted

by one rule, but denied by others. In Table 1, *r*2 correlates with *r*5, and all UDP packets coming from any port of 10.1.1.* to the port 53 of 172.32.1.* match the intersection of these rules. Since *r*2 is a preceding rule of *r*5, every packet within the intersection of these rules is denied by *r*2. However, if their positions are swapped, the same packets will be allowed.

- *Redundancy*: A rule is redundant if there is another same or more general rule available that has the same effect. For example, *r*1 is redundant with respect to *r*2 in Table 1, since all UDP packets coming from any port of 10.1.2.* to the port 53 of 172.32.1.* matched with *r*1 can match *r*2 as well with the same action.

Policy Conflicts

A policy conflict pc in a firewall F is associated with a unique set of conflicting firewall rules cr={r1,……, rk}, which can derive a common network packet space. All packets within this space can match exactly the same set of firewall rules, where at least two rules have different actions: Allow and Deny.

Rule Redundancy

A rule r in a firewall F is redundant if removing r from F fulfills that the network packet space derived from the new firewall F0 is equal to the network packet space defined by F. That is, F and F0 satisfy following equations: SA F = SA F0 and SD F = SD F0 , where SA and SD denote allowed and denied network packet spaces, respectively.

Packet Space Segmentation

A more effective anomaly resolution, we adopt a rule-based segmentation technique, which can convert a list of rules into a set of disjoint network packet spaces.

Figure 2(a) gives the two dimensional geometric representation of packet spaces derived from the example policy shown in Table 1.

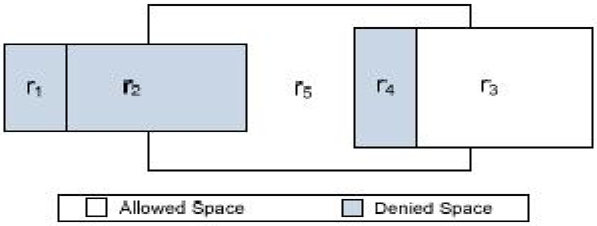


Figure 3: Two dimensional geometric representation of overlapping rules

We utilize colored rectangles to denote two kinds of packet spaces: allowed space (white color) and denied space (grey color), respectively. In this example, there are two allowed spaces representing rules r3 and r5, and three denied spaces depicting rules r1, r2 and r4.

Two spaces overlap when the packets matching two corresponding rules intersect. For example, *r*5 overlaps with *r*2, *r*3 and *r*4, respectively. An overlapping relation may involve multiple rules. In order to clearly represent all identical packet spaces derived from a set of overlapping rules, we adopt the rule-based segmentation technique to divide an entire packet space into a set of pairwise disjoint segments.

We classify the policy segments as follows: nonoverlapping segment and overlapping segment, which is further divided into conflicting overlapping segment and non-conflicting overlapping segment.

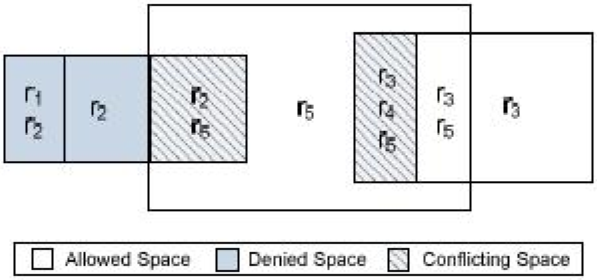


Figure 4: Packet Space Segmentation

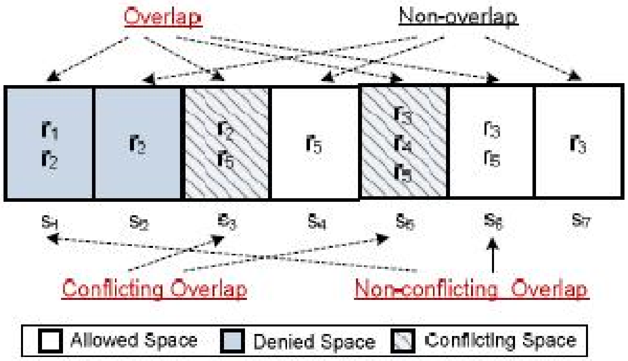


Figure 5: uniform representation

Three policy segments s2, s4 and s7 are non-overlapping segments. Other policy segments are overlapping segments, including two conflicting overlapping segments s3 and s5, and two non-conflicting overlapping segments s1 and s6.

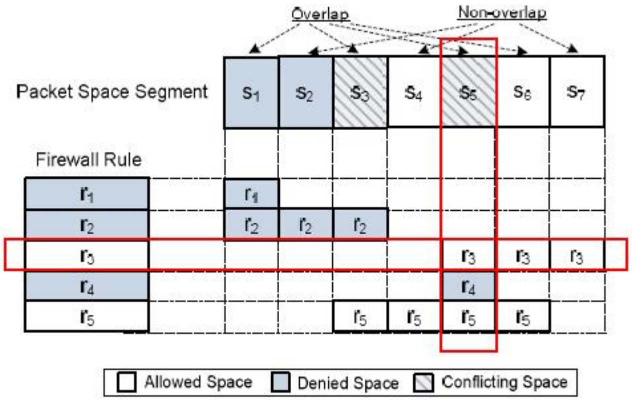


Figure 6: Grid Representation of policy Anomaly

Grid Representation of Policy Anomaly

In the above diagram, the administrator difficult to identify the one rule participates in different segments. We additionally introduce a grid representation that is a matrix-based visualization of policy anomalies, in which space segments are displayed along the horizontal axis of the matrix, rules are shown along the vertical axis, and the intersection of a segment and a rule is a grid that displays a rule's subspace covered by the segment. We can easily determine which rules are covered by a segment, and which segments are associated with a rule.

Correlation of Packet Space Segment

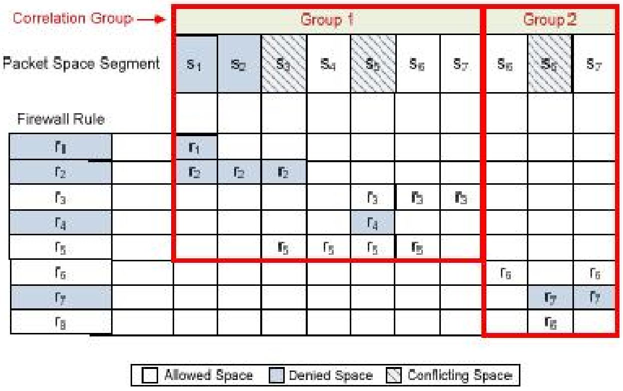


Figure 7: Segment Correlation Example

Actually , Several rules in this firewall policy are involved in multiple anomalies. For example, *r*2 is associated with three segments *s*1, *s*2 and *s*3. Also, we can identify *r*3, *r*5, *r*6 and *r*7 are also associated with multiple segments. Assume we need to resolve the conflict related to a conflicting segment *s*3 by reordering associated conflicting rules, *r*2 and *r*5. The position change of *r*2 and *r*5 would also affect other segments, *s*1, *s*2, *s*4, *s*5 and *s*6. Thus, a dependency relationship among those segments can be derived. We cluster such segments with a dependency relationship as a group called *correlation group*.

Action Constraint

An action constraint ac for a conflicting segment cs defines a desired action (either Allow or Deny) that the firewall policy should take when any packet in the conflicting segment comes to the firewall.

Anomaly Management Framework

Our anomaly management framework is composed of two main functionalities: conflict detection and resolution, and redundancy discovery and removal,

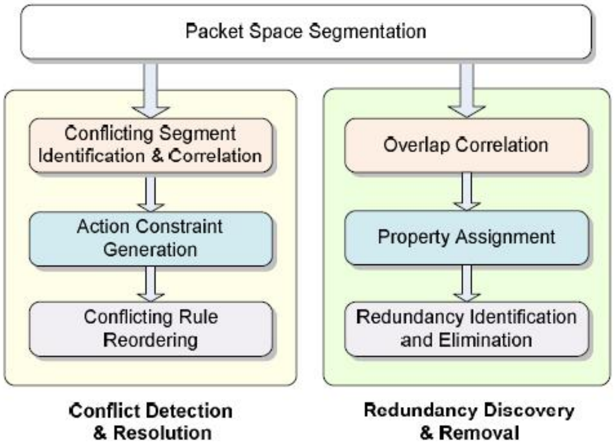

Packet Space Segmentation
Conflicting Segment Identification & Correlation
Action Constraint Generation
Conflicting Rule Reordering
Overlap Correlation
Property Assignment
Redundancy Identification and Elimination
Conflict Detection & Resolution
Redundancy Discovery & Removal

Figure 8: Firewall Policy Management Framework

Conflict Detection & Resolution

- First it identifies conflicting segments.
- The second step generates action constraints for each conflicting segment by examining the characteristics of each conflicting segment.
- The third step utilizes a reordering algorithm, which is a combination of a permutation algorithm and a greedy algorithm, to discover a near-optimal conflict resolution solution for policy conflicts.

Redundancy discovery and removal

- First it identifies segment correlation groups

- Second the process of property assignment is performed to each rule's subspaces. Consequently, redundant rules are identified and eliminated.

System Architecture of FAME

It consists of six components: segmentation module, correlation module, risk assessment module, action constraint generation module, rule reordering module, and property assignment module.

The segmentation module takes firewall policies as an input and identifies the packet space segments by partitioning the packet space into disjoint subspaces.

Once the segmentation of packet space is identified, FAME further identifies different kinds of segments and corresponding correlation groups

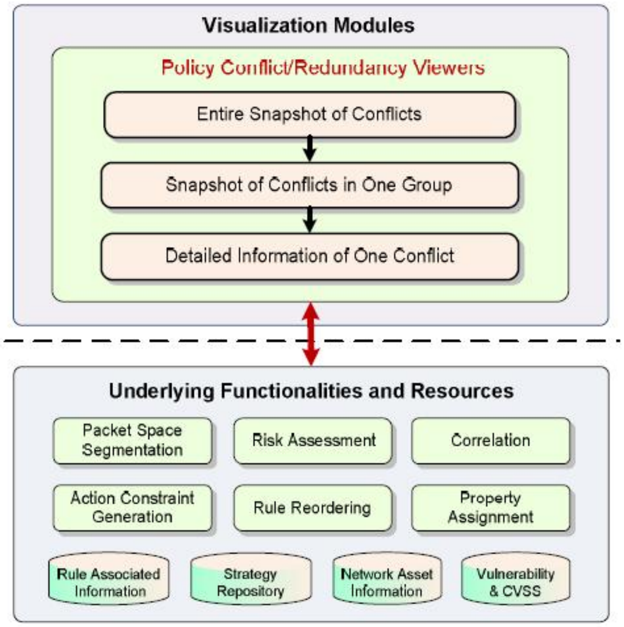

Visualization Modules
Policy Conflict/Redundancy Viewers
Entire Snapshot of Conflicts
Snapshot of Conflicts in One Group
Detailed Information of One Conflict
Underlying Functionalities and Resources
Packet Space Segmentation
Risk Assessment
Correlation
Action Constraint Generation
Rule Reordering
Property Assignment
Rule Associated Information
Strategy Repository
Network Asset Information
Vulnerability & CVSS

Figure 9 :Architecture of FAME

In risk assessment module, Nessus [3] is utilized as a *vulnerability scanner* to identify the vulnerabilities within a conflicting segment.

The action constraint generation module takes conflicting segments as an input and generates action constraints for each conflicting segment. Action constraints are generated based on strategies assigned to each conflicting segment.

The rule reordering module takes conflict correlation groups and action constraints of conflicting segments as inputs and generates optimal or near-optimal conflict resolution for policy conflicts using a combined reordering algorithm in our framework.

The property assignment module takes segment correlation groups as inputs and automatically assigns corresponding properties to each rule subspace covered by policy segments. The assigned properties are in turn utilized to identify redundant rules.

Firewall Policy Visualization Tool

A tool called Policy Visualization which visualizes firewall rules and policies in such a way that efficiently enhances the understanding and inspecting firewall policies.

Conclusion

In this paper, we have proposed a anomaly management framework that facilitates efficient detection and resolution of firewall policy anomalies. A rule-based segmentation technique was introduced to achieve the goal of effective and efficient anomaly analysis. In addition, we have described an implementation of our anomaly management environment called FAME, clearly demonstrating that our proposed anomaly analysis methodology is practical and useful for system administrators to enable an assurable network management.

Firewalls provide proper security services if they are correctly configured and efficiently managed. Firewall policies used in enterprise networks are getting more complex as the number of firewall rules and devices becomes larger. As a result, there is a high demand for an effective policy management tool which significantly helps user in discovering firewall policy's properties and finding rule anomalies in both single and distributed firewalls. PolicyVis presented in this paper provides visual views on firewall policies and rules which gives users a powerful means for inspecting firewall policies.

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

References

[1] E. Al-Shaer and H. Hamed. Firewall Policy Advisor for anomaly discovery and rule editing. In Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on, pages 17–30, 2003.

[2] E. Al-Shaer and H. Hamed. Discovery of policy anomalies in distributed firewalls. In IEEE INFOCOM, volume 4, pages 2605–2616, 2004.

[3] E. Al-Shaer, W. Marrero, A. El-Atawy, and K. ElBadawi. Network Configuration in A Box: Towards End-to-End Verification of Network Reachability and Security. In Proceedings of the 17th IEEE International Conference on Network Protocols (ICNP), pages 123–132, 2009.

[4] J. Alfaro, N. Boulahia-Cuppens, and F. Cuppens. Complete analysis of configuration rules to guarantee reliable network security policies. International Journal of Information Security, 7(2):103–122, 2008.

[5] F. Baboescu and G. Varghese. Fast and scalable conflict detection for packet classifiers. Computer Networks, 42(6):717–735, 2003.

[6] Y. Bartal, A. Mayer, K. Nissim, and A. Wool. Firmato: A novel firewall management toolkit. ACM Transactions on Computer Systems (TOCS), 22(4):381–420, 2004.

[7] S. Bellovin. Distributed firewalls. Journal of Login, 24(5):37–39, 1999.

[8] C. Brodie, C. Karat, and J. Karat. An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. In Proceedings of the second symposium on Usable privacy and security, page 19. ACM, 2006.

[9] E. Chew, M. Swanson, K. Stine, N. Bartol, A. Brown, and W. Robinson. Performance measurement guide for information security. NIST Special Publication, pages 800–55, 2008.

[10] A. El-Atawy, K. Ibrahim, H. Hamed, and E. Al-Shaer. Policy segmentation for intelligent firewall testing. In 1st Workshop on Secure Network Protocols (NPSec 2005), 2005.

Raghvendra Swamy.K received his Mastres Degree in computer Science From Sri Y.N College, Narsapur,, in 2003, the M.TECH. degree in CSE from Eluru College of Engineering and Technology,Eluru in 2013. At present, He is engaged in**"Efficiently Managing Firewall conflicting policies".**

B.Prashant Received His B.Tech Degree In EEE

From Bapatla Engineering College, Bapatla, Guntur(Dt), in 2002, M.Tech. Degree in CSE from Nova College Of Engineering And Technology,Jangareddygudem in 2011. He has 8 years of experience in teaching. Currently he is working as Associate Professor in Eluru College of Engineering and Technology, Eluru. Area Of Interests Artificial Intelligence And Neural Networks, Memory Management, Data Mining.