IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Anomaly Management Framework And Visualization Approach Of A Firewall

Savita Panghal[1], VK Srivastva[2]

1. Baba Mastnath University, Asthal Bohar, Rohtak
2. Professor, Head of Dept. of Computer Science &Application, BMU, Asthal bohar, Rohtak

Abstract: As we know that firewall are widely emplaced in security apparatus to ensure the security of private networks in most organizations. Basically effectiveness of firewall depends on the quality of policy and its configuration. Nevertheless, the designing and managing its policies are untrustworthy because of its complex nature of configuration as well as the lack of systematic analysis apparatus and tools. This paper represents an anomaly management framework and visualization approach of firewall. We signify the possibility and relevancy of framework by a proof of concept prototype of a visualization which is based on firewall policy analysis tool known as firewall Anomaly Management Environment (FAME). We also introduce FIREMAN, an immobile analysis toolkit for firewall modeling and analysis. By treating firewall configurations as specialized programs.

Keywords: - firewall policies, anomaly management, visualization tool.

I INTRODUCTION

Firewall is basically survey all incoming and outgoing data in public and private network based on security rules to monitor doubtful traffic and unauthorized access to internet based organization. To execute a security policy in a firewall, system executive define a set of filtering rules that are obtain from the organizational network security requirements. In firewall policy management is a challenging exercise due to the difficulty and correlation of policy rules. Therefore it is worsen by the continual revision of network environments. However, a successful procedure and tools for policy management are critical to the success of firewalls. In a policy anomaly detection received a great share out of attention [1, 2, 3, 4, 5]. Equivalent policy analysis tools, like Firewall Policy Advisor [1, 2] and FIREMAN [5], having goal of detecting policy anomalies were introduced. Firewall Policy Advisor detect pair wise anomalies in firewall rules, And in FIREMAN detection of anomalies with multiple rules by analyzing the relationships between one rule and the collections of packet spaces derived from all preceding rules.

FIREMAN, a static analysis toolkit for firewall modeling and analysis. It apply static analysis techniques to check misconfiguration, such as policy violations, inconsistencies, and inefficiencies, in individual firewalls as well as among distributed firewalls. FIREMAN performs symbolic model checking of the firewall configurations for all possible IP packets and along all possible data paths. It is both sound and complete because of the finite state nature of firewall configurations. However, FIREMAN also has limitations in detecting anomalies [3]. For each firewall rule, FIREMAN only examines all preceding rules but ignores all subsequent rules when performing anomaly analysis. Thus, anomaly detection procedures of FIREMAN are incomplete. Moreover analysis result of FIREMAN is only showing that there is a misconfiguration between one rule and its preceding rules, but cannot accurately indicate all rules connected with an anomaly.

On other side, because of complex nature of policy anomalies, system administrators are frequently 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. Although, changing the conflicting rules is remarkably difficult, even impossible, in practice from many characteristics. First one, that number of conflicts in a firewall is usually large, now a firewall policy may consist of thousand of rules, which are often logically twisted with each other. Secondly, policy conflicts are frequently very complex. One rule may friction with many other rules, and one friction may be related with several rules. As well as, firewall policies place on a network are generally

Maintained by more than one administrator and an enterprise firewall may contain tradition rules that are designed by different administrators. Hence, without a prior knowledge on the administrators" intentions, changing rules will affect the rules" semantics and may not resolve conflicts correctly. Moreover, in some cases, a system administrator may knowingly introduce certain overlaps in firewall rules knowing that only the first rule is important. Actually, this is a commonly used technique to exclude specific parts from a certain action, and the proper use of this technique could result in a fewer number of rules [5]. In this case, conflicts are not an error, but planned, which would not be necessary to be changed.

Therefore the policy conflict in firewalls always exist and are hard to be removed, a practical resolution method is to identify which rule involved in a conflict situation should take precedence when multiple conflicting rules can filter a particular network packet concurrently. To resolve policy frictions, a firewall commonly implements a first-match resolution mechanism based on the order of rules. Through this, each packet processed by the firewall is mapped to the decision of the first rule that the packet matches. Although, applying the first-match strategy to cope with policy conflicts has limitations. When a conflict occurs in a firewall, the existing first matching rule may not be a desired rule that should take priority with respect to conflict resolution. In particular, the existing first matching rule may perform opposite action to the rule which should be considered to take priority. This situation can cause serious network breaches such as permitting harmful packets to sneak into a private network, or dropping legal traffic which in turn could hinder the availability and utility of network services. Certainly, it is necessary to seek a way to bridge a gap between friction detection and using first-match mechanism for resolving friction in firewalls.

In this paper, we represent a novel anomaly management framework based on a rule-based segmentation technique to make possible not only more accurate anomaly detection but also effective anomaly resolution. Furthermore, the outputs of prior policy analysis tools [1, 2, 5] are mainly a list of possible anomalies, which does not give system administrators a clear view of the creation of policy anomalies. Therefore information visualization technique [6] enables users to explores, analyze, reason and explain abstract information by taking advantage of their visual perception, our policy analysis tool adopts an information visualization technique to facilitate policy analysis. A grid-based visualization technique is introduced to represent outputs of policy anomaly analysis, authorize an efficient anomaly management. The implementation of our visualization- based policy analysis tool called Firewall Anomaly Management Environment (FRAME) is discussed as well as.

I Firewall Policies Overview

A firewall policy consists of a series of rules that define the actions performed on packets that satisfy certain conditions. Basically rules are specified in the form of (condition, action). A CONDITION in a rule is composed of a set of fields to identify a certain type of packets matched by this rule. Five fields are mainly used in a rule’s condition: 1. Protocol type 2. Source IP, 3. Source Port 4. Destination IP and 5. Destination port. Those fields are either a single value or a finite interval of non-negative integers. An ACTION in a rule describes the rule as well as describes the corresponding action performed on the matched packets and typically takes the value ”allow” which permits the packets passes through the firewall, or “deny”, which leads to the packets to be blocked.

A packet matches a rule if and only if the header information of the packet satisfies all fields in the rule. Along finding a matching rule, the correlate with decision for the packet is Obtain. A firewall policy with a series of rules usually follows a First-match semantic to assess a packet: the decision of the first matching rule is applied to the packet. If there is no matching rule that could be found in the firewall policy, a default action is performed. Most firewalls use “deny” as the default action, hinted every packet that could not be matched by any rules will be denied. As shown in figure 1 an example of a firewall policy, which includes five firewall rules r_1, r_2, r_3, r_4 and r_5 . Note that the symbol “*” utilized in firewall rules denotes a domain range. For instance, a single”*” appearing in the IP address field represents an IP address range from 0.0.0.0 to 255.255.255.255.

Order	Rules	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
1	r_1	UDP	10.1.2.*	*	172.32.1.*	53	Deny
2	r_2	UDP	10.1.*.*	*	172.32.1.*	53	Deny
3	r_3	TCP	10.1.*.*	*	192.168.*.*	25	Allow
4	r_4	TCP	10.1.1.*	*	192.168.1.*	25	Deny
5	r_5	*	10.1.1.*	*	*	*	Allow

Figure1. An example of firewall policy

II. FIREWALL POLICIES AND ANOMALIES IN IT.

Two rules in a firewall policy may overlap, which means one packet may match both rules. Furthermore, two rules in a firewall may friction, implying that those two rules not only overlap each other but also take different actions. Policy conflicts may lead to both security problems and availability problems and policy redundancies will affect the performance of a firewall. A Panoramic classification of policy anomalies has been expressive by several related work [2, 5]. Some existing classification, of policy anomalies are:-

i) Generalization: - In this generalization of one or a set of previous rules if a subset of the packets match with this rule is also matched by preceding rules but taking a different action. e.g., in Figure 1 r_5 is a generalization of r_4. These two rules indicate that all the packets from 10.1.1.* are allowed, except TCP packet from 10.1.1.* to the port 25 of 192.168.1.*. In this paper we have already discussed that generalization might not be an error.

ii)Shadowing:- In this 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. Though 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. As shown in figure 1, r_4 is shadowed by r_3 because r_3 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 r_4.

iii) 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. As shown in figure.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. Although, if their positions are swapped, the same packets will be allowed.

iv) Redundancy: - A rule is redundant if there is another same or more general rule available that has the same effect. e.g.:- r_1 is redundant with respect to r_2 in figure. 1, therefore 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.

Anomaly detection algorithms and corresponding tools were also introduced by [2, 5]. Nevertheless, prior work only treated a policy conflict as an inconsistent relation between one rule and other rules. Given a more general definition on policy conflict as given in definition I, we believe that identifying policy conflicts should always consider a firewall policy as a whole piece, and precise indication of the rule set involved in a conflict is critical for effectively resolving the conflict.

Definition 1. (Policy Conflict). A policy conflict pc in a firewall F is associated with a unique set of conflicting firewall rules

cr ={r_1,………,r_k },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.

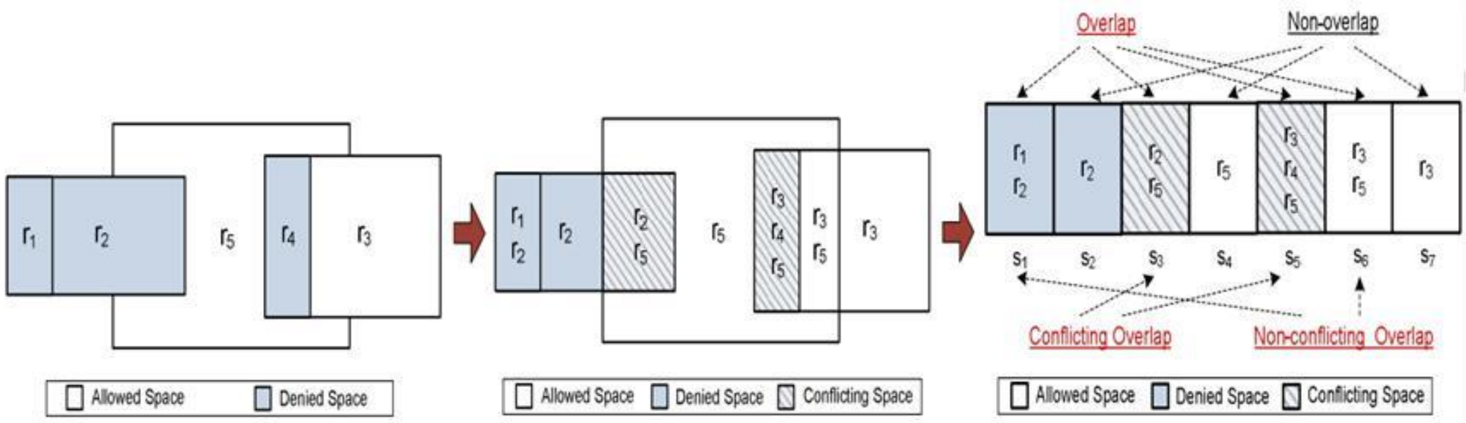
Definition 2:- (Rule Redundancy). A rule r in a firewall F is redundant if removing r from F fulfills that the network packet space derived from the new firewall F' is equal to the network packet space defined by F. that is, F and F' satisfy given equations: S $\frac{A}{F}$= S$\frac{A}{F'}$ $and\ S\ \frac{D}{F}$ =S $\frac{D}{F'}$, where S^A and S^D denote allowed and denied network packet spaces, systematically.

III PACKET BASED REPRESENTATION OF ANOMALY

A) Classification and Packet Space segmentation

According to the section II, present anomaly detection methods are not purely point out the anomaly portions caused by a set of overlapping rules. In order to exactly identify policy anomalies and enable 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. This technique has been recently introduced to deal with various research problems such as network traffic dimensions. [10], firewall testing [9] and optimization [7, 8]. Inspired by those successful applications, we take on technique for the purpose of firewall anomaly analysis. To make easier the correct explanation of analysis results, a brief and instinctual representation method is necessary. For the purposes of conciseness and understandability, we employ a two dimensional geometric representation for each packet space proceeds from firewall rules. A firewall rule typically utilizes five fields to define the rule condition; hence a complete representation of packet space should be multi-dimensional. Figure 2(a) gives the two dimensional geometric representation of packet spaces originates from the example policy shown in figure 1. We make use of colored rectangle to denote two kinds of packet spaces: allowed space (white color) and denied space (grey color), order by. In this example, there are two allowed spaces representing rules r_3 and $r_{5,}$ and three denied spaces depicting rules $r_{1,}$ r_2 and r_4.

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 originate from a set of overlapping rules, we taking on the rule-based segmentation technique to divide an entire packet space into a set of pair wise disjoint segments. We classify the policy segments as follows: non overlapping segment and overlapping segment, which is further divided into conflicting overlapping segment and non-conflicting overlapping segment. Each non-overlapping segment associates with one unique rule and each overlapping segment is related to a set of rules, which may conflict with each other (conflicting overlapping segment) or have the same action (non-conflicting overlapping segment). Figure 2(b) indicates the segments of packet spaces derived from the example policy. Hence the size of segment representation does not give any specific benefits in resolving policy anomalies; we further present a uniform representation of space segments in Figure 2(c). We can notice that seven unique disjoint segments are generated. Three policy segments s_2, s_4 and s_7 are non-overlapping segments. Other policy segments are overlapping segments, including two conflicting overlapping segments s_3 and s_5, and two non-conflicting overlapping segments s_1 and s_6.



(A) Two dimensional geometric representation (b) Packet space segmentation (c) Uniform representation

of overlapping rules

Figure: - 2 Packet space representations originate from the example policy.

B) Policy Anomaly By Grid Representation

To enable an effective anomaly resolution, complete and accurate anomaly recognition information should be represented in an automatic way. When a set of rules interacts, one overlapping relation may be connected with several rules. For the moment, one rule may overlap with multiple other rules and can be involved in a couple of overlapping relations (overlapping segments). Different kinds of segments and connected rules can be viewed in the uniform representation of anomalies (Figure 2(c)). Nevertheless, it is still difficult for an administrator to figure out how many segments one rule is involved in. To address the need of a more precise anomaly representation Figure 3 shows a grid representation of policy anomalies for our example policy. We can easily determine which rules are covered by a segment, and which segments are connected with a rule. For example, as shown in Figure 3, we can notice that a conflicting segment s5, which points out a conflict, is related to a rule set consisting of three conflicting rules r_3, r_4 and r_5 (highlighted with a horizontal red rectangle), and a rule r_3 is involved in three segments s_5, s_6 and s_7 (highlighted with a vertical red rectangle). Our grid representation provides a better understanding of policy anomalies to system administrators with an overall view of related segments and rules.

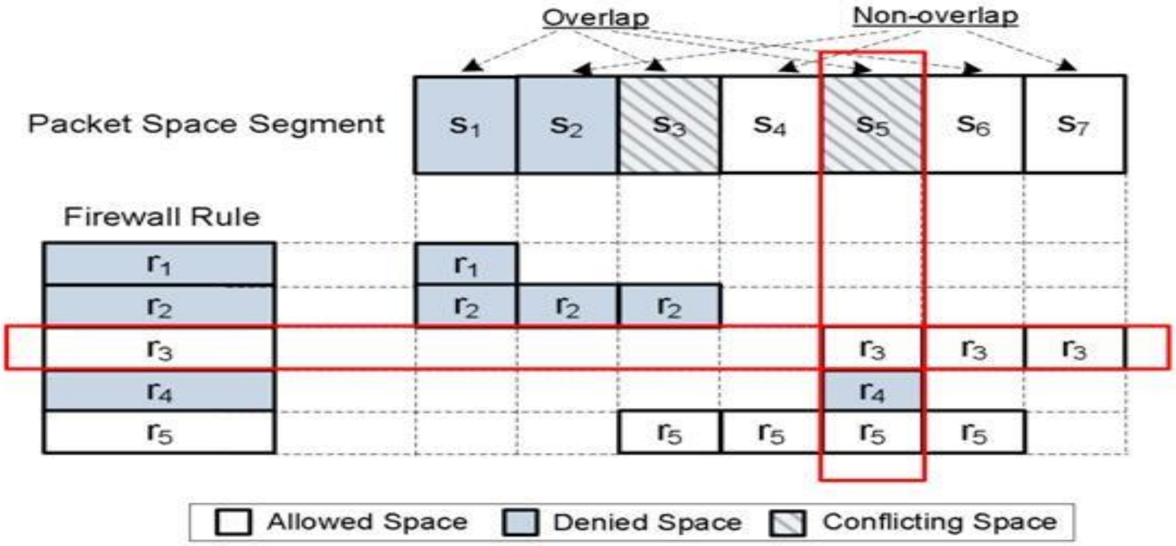


Figure 3: Policy Anomaly By Grid Representation

IV FRAMEWORK OF ANOMALY MANAGEMENT

Anomaly management framework is composed of two core functionalities: conflict detection and resolution, and redundancy discovery and removal, as depicted in Figure 4. Both functionalities are based on the rule-based segmentation technique. For conflict detection and resolution, conflicting segments are identified in the first step. Each conflicting segment associates with a policy conflict and a set of conflicting rules. Also, the correlation relationships among conflicting segments are identified and conflict correlation groups are derived. Policy conflicts belonging to different conflict correlation groups can be resolved separately; hence the searching space for resolving conflicts is reduced by the correlation process. The second step generates action constraints for each conflicting segment by examining the characteristics of each conflicting segment. A strategy-based method is introduced for generating action constraints. On the other side, it is also inefficient to deal with all conflicts together by reordering all conflicting rules concurrently. Thus, it is necessary to identify the dependency relationships among packet space segments for efficiently resolving policy anomalies.

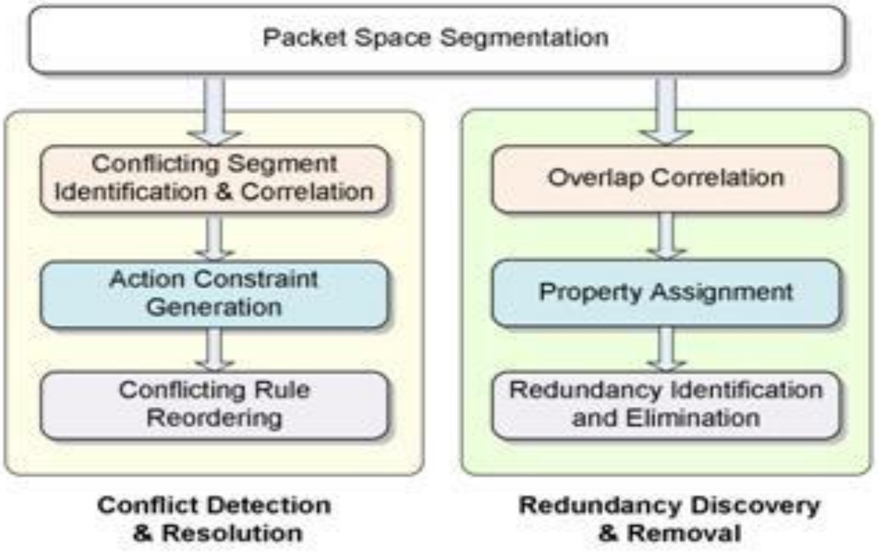

Packet Space Segmentation
Conflicting Segment Identification & Correlation
Action Constraint Generation
Conflicting Rule Reordering
Overlap Correlation
Property Assignment
Redundancy Identification and Elimination
Conflict Detection & Resolution
Redundancy Discovery & Removal

Figure 4 FRAMEWORK OF ANOMALY MANAGEMENT

V FAME: FIREWALL ANOMALY MANAGEMENT ENVIRONMENT

Our framework is realized as a proof-of-concept prototype called FAME. Figure 5 shows a high level architecture of FAME with two levels. The upper level is the visualization layer, which visualizes the results of policy anomaly analysis to system administrators. Two visualization interfaces, policy conflict viewer and policy redundancy viewer, are designed to manage policy conflicts and redundancies, respectively. The lower level of the architecture provides underlying functionalities addressed in our policy anomaly management framework and relevant resources including rule information, strategy repository, network asset information, and vulnerability information.

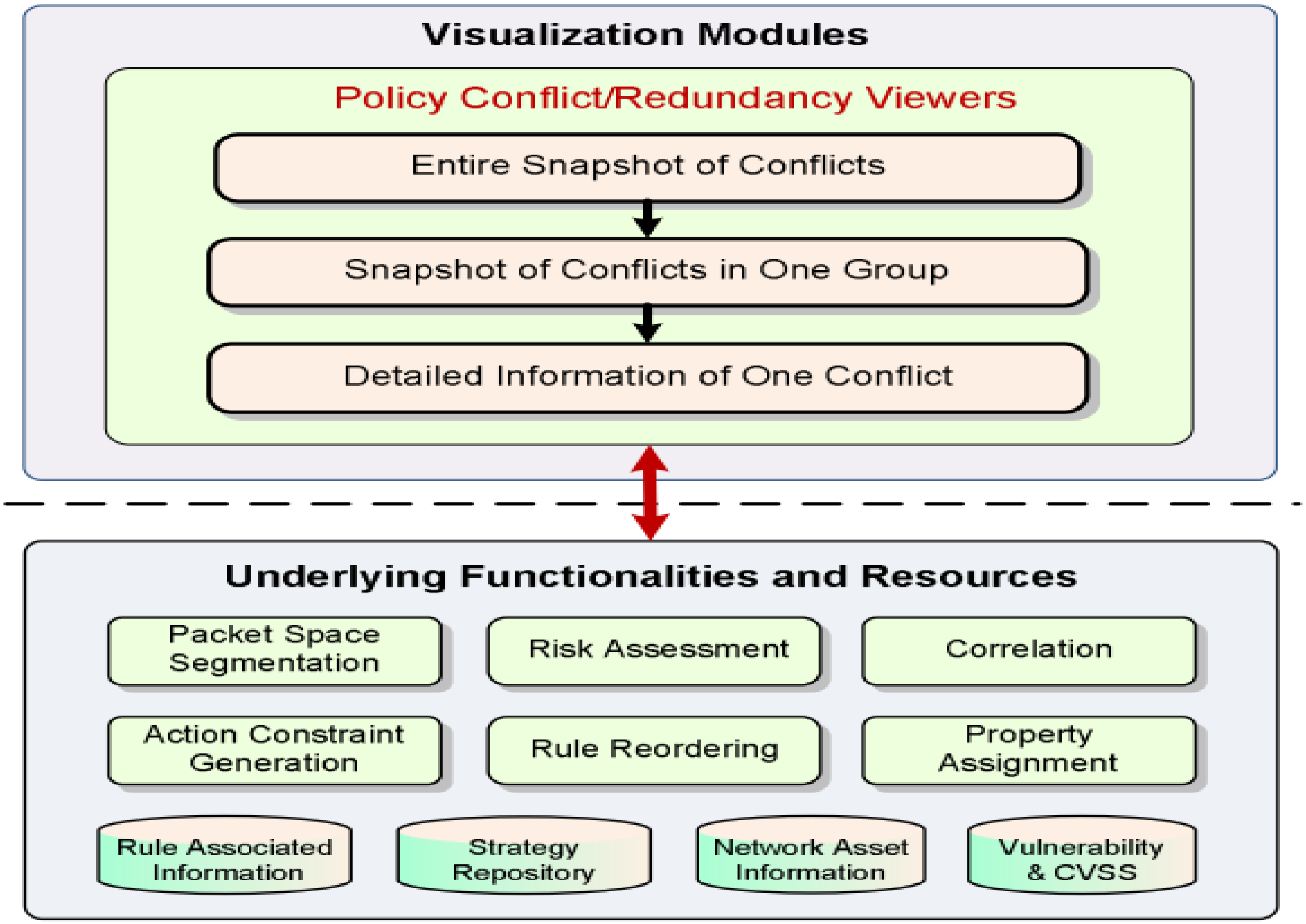

Visualization Modules
Policy Conflict/Redundancy Viewers
Entire Snapshot of Conflicts
Snapshot of Conflicts in One Group
Detailed Information of One Conflict
Underlying Functionalities and Resources
Packet Space Segmentation
Risk Assessment
Correlation
Action Constraint Generation
Rule Reordering
Property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

Figure 5. FAME: ARCHITECTURE

VI CONCLUSIONS

In this paper we present a novel anomaly management framework that facilitates systematic detection and resolution of firewall policy anomalies. A rue based segmentation technique was introduced for achieving the goal of successful and systematic anomaly analysis. We have described our anomaly management environment called FAME. for future work usability studies to evaluate functionalities and system requirement of our policy visualization approach with subject matter experts.

VII 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] 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.

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

[5] L. Yuan, H. Chen, J. Mai, C. Chuah, Z. Su, P. Mohapatra, and C. Davis. Fireman: A toolkit for firewall modeling and analysis. In 2006 IEEE Symposium on Security and Privacy, page 15, 2006.

[6] I. Herman, G. Melançon, and M. Marshall. Graph visualization and navigation in information visualization: A survey. IEEE Transactions on Visualization and Computer Graphics, pages 24–43, 2000.

[7] A. El-Atawy, T. Samak, E. Al-Shaer, and H. Li. Using online traffic statistical matching for optimizing packet filtering performance. In IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications, pages 866–874, 2007.

[8] G. Misherghi, L. Yuan, Z. Su, C.-N. Chuah, and H. Chen. A general framework for benchmarking firewall optimization techniques. IEEE Transactions on Network and Service Management, 5(4):227–238, Dec. 2008.

[9] 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.

[10] L. Yuan, C. Chuah, and P. Mohapatra. ProgME: towards programmable network measurement. ACM SIGCOMM Computer Communication Review, 37(4):108, 2007.

[11] Hongxin Hu University at Buffalo, The State University of New York

https://www.researchgate.net/publication/228362672_FAME_A_firewall_anomaly_management_environment