A Novel Management Framework for Policy Anomaly in Firewall

Ms. S. Selvakanmani[1]
[1]Assistant Professor
[1]Department of Computer Science & Engineering
[1]Velammal Institute of Technology, Velammal Gardens, Chennai, Tamil Nadu.

Abstract—The advent of emerging technologies such as Web services, service-oriented architecture, and cloud computing has enabled us to perform business services more efficiently and effectively. However, we still suffer from unintended security leakages by unauthorized actions in business services. Firewalls are the most 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. Unfortunately, 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. In this paper, we represent an innovative policy anomaly management framework for firewalls, adopting a rule-based segmentation technique to identify policy anomalies and derive effective anomaly resolutions. We also discuss a proof-of-concept implementation of a visualization-based firewall policy analysis tool called Firewall Anomaly Management Environment (FAME). In addition, we demonstrate how efficiently our approach can discover and resolve anomalies in firewall policies through rigorous experiments using Automatic rule generation technique.

Key words: FAME, policy anomaly, firewall, segment.

I. INTRODUCTION

Firewall is a system that protects the resources of a private network from intrusions coming from other networks. The firewall contains rules used to allow or deny incoming traffic. These rules form the security policy of the firewall. The large size and complexity of modern networks result in large and complex firewall policies (e.g. they may contain up to 50000 rules). Safely designing a policy is a hard task as a large number of cases are to be considered for avoiding malicious access situations. Moreover, a network administrator may want to update in real time the active policy in order to replace it by new one. The process of updating firewall policies is difficult and error prone. Indeed, it has to conciliate keeping network services running and avoiding security holes.

To enhance the efficient and secure data transfer between networks of distributed firewalls by generating the rules and setting actions according to the requirement of the organization. Changes in the requirement or changes in the rule by different administrator in the dynamic distributed environment cause policy conflict in firewalls. It enables the systematic rule generation and action setting to reduce the manual work done by the administrator in distributed firewalls.

In a distributed fire wall policy anomaly detection, resolution, rule generation and action setting is difficult task by administrator. This work adopts the Firewall Anomaly Management Environment (FAME) policy anomaly management framework, an anomaly management framework applied to the distributed firewall environment. Due to the dynamic changes of the security requirements of the organization policy could be overwritten or new policy can be generated to satisfy the requirements. In FAME policy anomaly can be identified by means of the rule based segmentation technique. A network packet space defined by a firewall policy can be divided into a set of disjoint packet space segments. Each segment associated with a unique set of firewall rules accurately indicates an overlap relation, which is either conflicting or redundant among those rules.

The objective of this work is to provide an innovative Automatic Rule Generation and action setting in distributed firewalls by adopting the policy anomaly management framework and rule based segmentation technique to identify policy anomalies and derive effective anomaly resolutions.

The rest of this paper is organized as follows. In Section 2, the literature review of the existing system is described. The architecture of the proposed work is given in the Section 3, the implementation details of the work are given in Section 4 with screen shots and finally, Section 5 concludes the work, followed by references.

II. LITERATURE REVIEW

Hamid Ali. F.A et.al [2] describes in their paper that firewall is a subject in form of hardware or software or both, which is use to protects a network from intrusion by outsiders. It is regulating the traffic that can pass through a router that connecting to the network infrastructure. It is preventing from unauthorized users to access the network whether from outside via the Internet or inside, Local Area Network (LAN) itself. Firewall system has many methods and techniques that imply and describes on each of their types or technologies. In their paper, they discussed about how all the methods in types of firewall functioning and give the advantages and disadvantages on decision making in a network security policy especially on using firewall system.

Yi Yin et.al proposed a detection system for packet filtering in firewall either accepts or denies network packets based upon a set of pre-defined filters called firewall policy. [3] Generally, Firewall policy is designed under the instruction of security policy. A network security policy is a generic document that outlines the needs for computer network access permissions. And it determines how firewall filters are designed. If inconsistencies, such as redundant filters, insufficient filter or contradict filters, exist between security policy and firewall policy, firewall policy could not filter packets exactly, and the network protected by the firewall will be affected. To resolve this problem, an inconsistency detection system can be used to detect the inconsistencies between the security policy and firewall

policy. When the administrator could not get host IP addresses, port number and other specific values, according to the network configurations, the proposed system could transform the network security policy and firewall policy to the same range value, represent and analyze their spatial relationships to detect their inconsistencies.

Kotenko.I outlines an approach for the verification of filtering rules of firewalls. The approach is intended for detection and resolution of filtering anomalies in the specification of security policy of computer networks. [4] It is based on Model Checking technique. The system proposes the models of computer networks, the models of firewalls and filtering anomalies, as well as an algorithm of detection of such anomalies.

Wool. A presented the first quantitative evaluation of the quality of corporate firewall configurations, which appeared in 2004, [5] based on Check Point Firewall-1 rule sets. In general, that survey indicated that corporate firewalls often enforced poorly written rule sets. In addition to being larger, the current study includes configurations from two major vendors. It also introduces a firewall complexity. The study's findings validate the 2004 study's main observations: firewalls are poorly configured, and a rule - set's complexity is positively correlated with the number of detected configuration errors.

Hu.H et al uses an innovative policy anomaly analysis approach which focuses on XACML (Extensible Access Control Markup Language) policy. [6] XACML has become the standard for specifying and enforcing access control policies for various Web based applications and services. It introduces a policy-based segmentation technique to accurately identify policy anomalies and derive effective anomaly resolutions. With the help of XAnalyzer, policy designer could easily discover and resolve anomalies in an XACML policy.

From the review, it is found that Firewall in the current scenario, undergoes certain problems like: (i) Network Dependent (ii) Not able to protect the system from internal threats. (iii) Rule generation by the administrator is complex and error prone task. (iv) Cannot detect the policy anomalies in the distributed firewalls.

III. PROPOSED WORK AND SYSTEM ARCHITECTURE

A distributed firewall preserves central control of access policy, which eliminates the dependency on topology. The proposed work introduces new ARG (Automatic Rule Generation) algorithm for distributed firewalls. The ARG algorithm proposed for automatically generating rules, detecting and resolving policy anomaly in distributed firewalls. By automating the task of administrator in distributed environment, it reduces the complexity and increases flexibility.[1]

The proposed system architecture in Fig.1 which has the following advantages: (i) No restriction for topological boundary. (ii) Automatic rule generation detects and resolves the policy anomalies in distributed firewalls. (iii) Eliminates redundancy (iv) Reduces complexity and increases flexibility.

In the proposed work, rules and actions are generated or modified according to the changes in the requirements of the dynamic environment. When a client sends a data packet to network, firewall checks the packet characteristics and decides to allow/deny the packet flow into the network. [1] The firewall rule anomalies are identified using packet space segmentation technique, and then the risk of anomalies is assessed, based upon the risk, the firewall rules are re-ordered. Risk assessment is measured using an upper bound and lower bound threshold values.

The proposed work includes the following stages:

- Automatic rule generation
- Packet Space Segmentation
- Action Constraint Generation
- Rule Reordering
- Data Package

A. Automatic Rule Generation

When the client wants to send data packets to the network, some set of firewall rules should be satisfied to allow the packets in Fig 2. For this, network administrators from different location allocate certain firewall rules to the server. Here generation of firewall rules and actions are done automatically. This process is performed by taking certain specifications and constraints. [1] The specification are taken and mapped randomly to generate the firewall rules. The rules are generated in the rule engine, the action happens when a client sends data packet to rule engine.

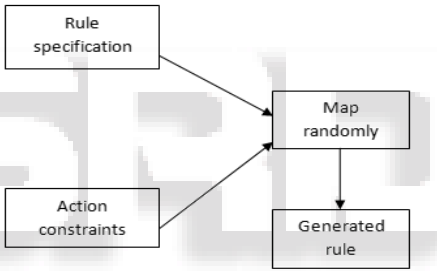

Fig. 2: Automatic rule Generation

B. Packet Space Segmentation

The major benefit of generating correlation groups (Fig 3) for the anomaly analysis is that anomalies can be examined within each group independently, because all correlation groups are independent of each other. Especially, the searching space for reordering conflicting rules in conflict resolution can be significantly lessened and the efficiency of resolving conflicts can be greatly improved. In order to precisely identify policy anomalies and enable a more effective anomaly resolution, a rule-based segmentation technique is introduced here. [12]Therefore, one can utilize set operations to separate the overlapped spaces into disjoint spaces.

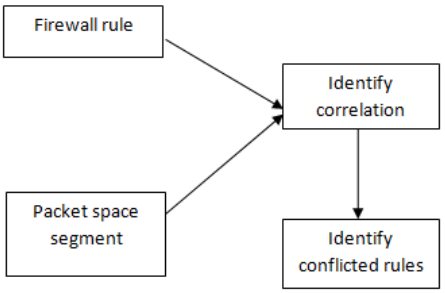

Fig. 3: Correlation of Packet space segment

C. Action Constraint Generation

Firewall policies are discovered and conflict correlation groups are identified, the risk assessment for conflicts is performed. The risk levels of conflicts are in turn utilized for both automated and manual strategy selections. A basic idea of automated strategy selection is that a risk level of a conflicting segment is used to directly determine the expected action taken for the network packets in the conflicting segment. If the risk level is very high, the expected action should deny packets considering the protection of network perimeters. [1]. On the contrary, if the risk level is quite low, the expected action should allow packets to pass through the firewall so that the availability and usage of network services cannot be affected. Thus, conflict resolution strategies (RS) can be generated automatically (as in Fig 4) for partial conflict segments by comparing the risk levels with two thresholds, upper threshold (UT) and lower threshold (LT), which can be set by system administrators in advance based on the different situations of protected networks.[6]

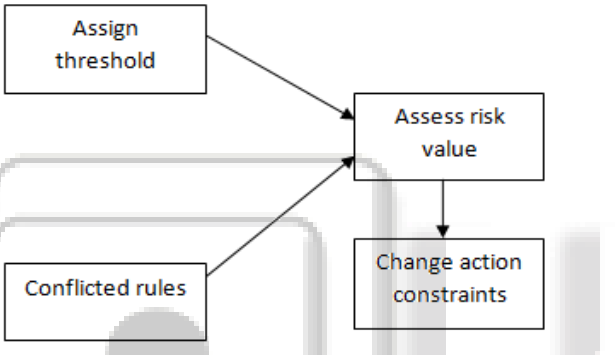

Fig. 4: Action Constraint Generation

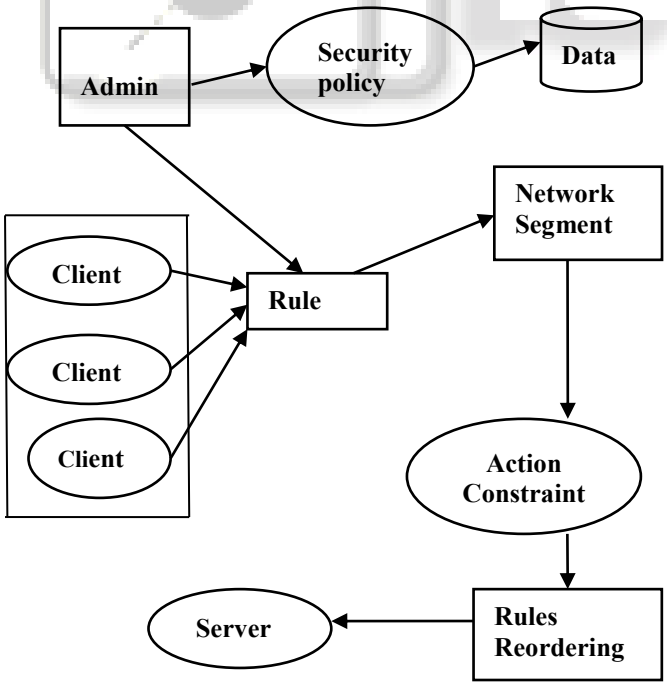

Fig. 5: System Architecture of the Proposed Work

D. Rule Reordering

The solution for conflict resolution is that all action constraints for conflicting segments can be satisfied by reordering conflicting rules. In conflicting rules in order that satisfies all action constraints, this order must be the optimal solution for the conflict resolution. For all conflicting rules in a correlation group, the greedy conflicting resolution algorithm [1] first calculates a resolving score for each conflicting rule individually. Then, the rule with the greatest resolving score is selected to solve the conflicts. A position range with the best conflict resolution is identified for the selected rule and moving the selected rule to the new position achieves a locally optimal conflicting resolution. [7]

E. Data Package

When conflicts in a policy are resolved, the risk value of the resolved policy should be reduced and the availability of protected network should be improved comparing with the situation prior to conflict resolution based on the threshold value data will be received in to the server. To evaluate the risk reduction [1] and availability improvement of conflict resolution approach, the results of conflict-resolved policies compared with the original policies as well as the best case and worst case with respect to the conflict resolution. [8]The best case of a conflict resolution is achieved when all action constraints assigned to the conflicting segments can be satisfied. The worst case considering the security risk is that all packets covered by conflicting segments are allowed to pass through a firewall. And the worst case considering the availability is that all packets covered by conflicting segments assigned with "allow" action constraints are denied.

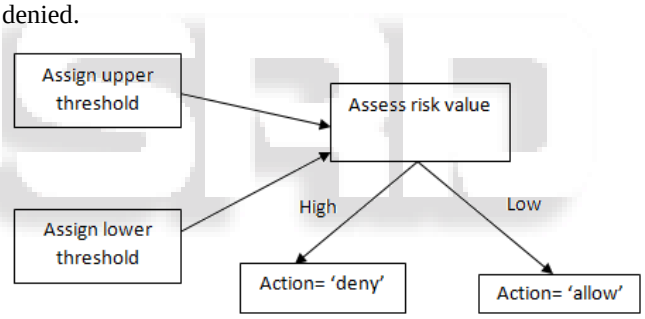

Fig. 6: Data Package

Each stage uses different techniques and algorithms along with its specification settings, for the working, which is discussed below.

1) Automatic Rule Generation Technique:

Specifications are set by the administrator for automatic rule generation. Specifications are automatically mapped with the constraints. Whenever the security requirements changed it must be reported to the administrator. After receiving the requirement report the administrator change the specification to match with that current requirement. [9]

2) The Random Mapping

According to the security requirements get from the members of the network the rule can be generated automatically by means of random mappings of the specifications.

3) Action Constraints Generation:

Based on the changes in the requirements the risk levels for allowing or denying the packet get changed. Changes in the risk level should reflect the changes in the action constraints. After the random mapping process the action constraints generated for that specific rule. [11]

- If Risk level is high then

$$action\ constraint\ =\ deny.$$

– If Risk level is low then

$$action\ constraint\ =\ allow.$$

Rule Generation: Final step is generating the rules after the above three steps. The rule contains the following things.

– Protocol (tcp/udp)
– Source and Destination IP (any)
– Source and Destination Port (any)
– Action (Allow/Deny)

4) Segment Generation Algorithm:

The algorithm for segment generation for the network packet shows the pseudo code of generating packet space segments for a set of firewall rules R. [12] this algorithm works by adding a network packet space s derived from a rule r to a packet space set S. A pair of packet spaces must satisfy one of the following relations.

a) All segments are pair wise disjoint
b) Any two different network packets p and p' within the same segment (si) are matched by the exact same set of rules:

GetRule(p) = GetRule(p'), for all p €€s_i, p'€ s_i, p ≠ p',

Where GetRule() is a function to return all matched rules of a network packet.

It adopts the rule-based segmentation technique addressed in segment generation algorithm to divide an entire packet space into a set of pair wise disjoint segments. [13]

5) The policy segments as follows:

Non-overlapping segment and overlapping segment, which is further divided into conflicting overlapping segment and nonconflicting 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 (nonconflicting overlapping segment).[11]

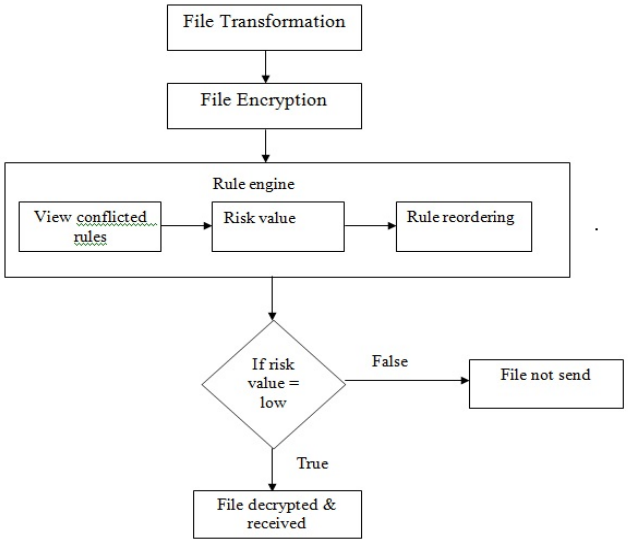

Fig. 6: Data Flow Diagram

The Data flow diagram for the proposed work is given in the Fig. 6. By definition, the Data flow diagram is a graphic tool used for expressing system requirements in a graphical form. The DFD also known as the "bubble chart" has the purpose of clarifying system requirements and identifying major transformations that to become program in system design. Thus DFD can be stated as the starting point of the design phase that functionally decomposes the requirements specification down to the lowest level of detail.

The type of users and their access can be better explained in the form of use case diagram in Fig. 7. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

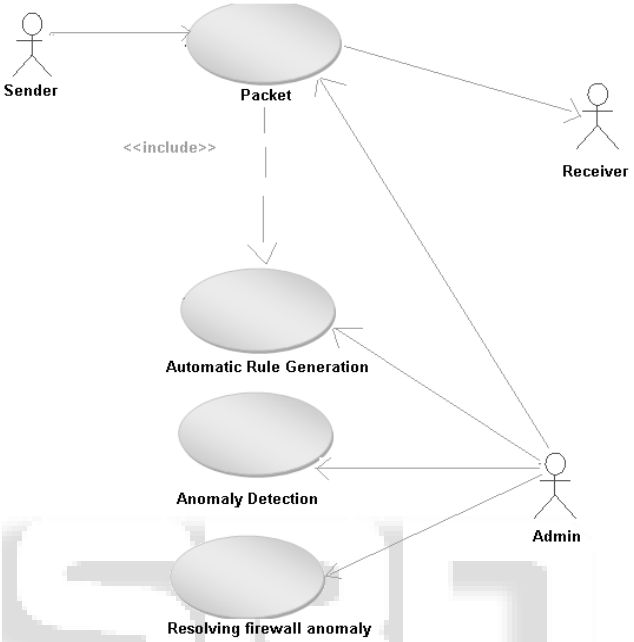

Fig. 7: Use Case Diagram

IV. IMPLEMENTATION OF THE PROPOSED WORK

The proposed work is implemented in Netbeans 6.9 by using Java and MS Access is used as the backend tool for storing the data. The screenshots of the proposed work is given.

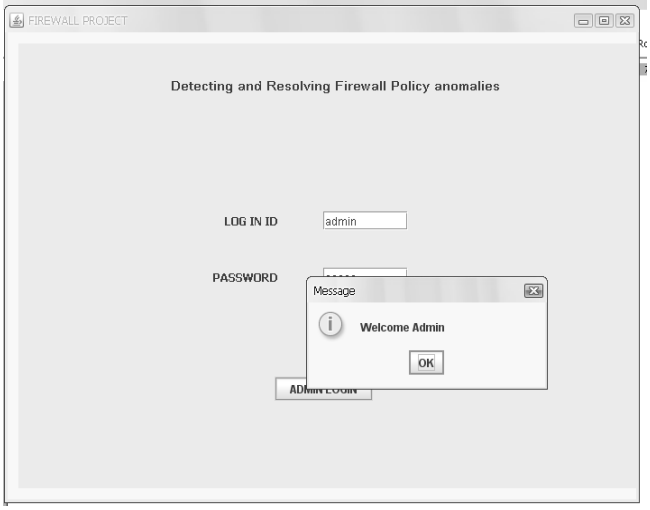

Fig. 8: Admin login

Fig. 8 gives the Admin Login form which is needed to give the entire protection of the system firewall. It is highly confidential information.

Rules are created in the form of a table which is given as follows, Table 1.

Rule Name	Protocol	Source IP	Source Port	Destination IP	Destination Port	Action

Table. 1: Rule Table

Here the rules are created as per the admin prototype mentioning the basic parameters(Fig. 9): Protocol , Source IP, Source port, destination IP and destination port.

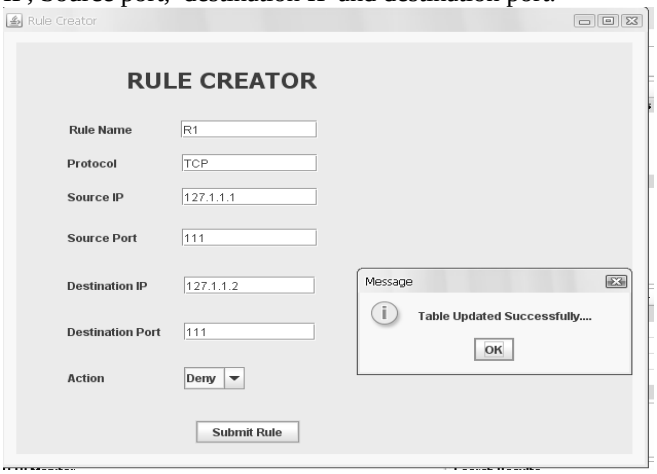

Fig. 9: Rule Creation

The working of the rule creation can be explained with the help of query processing, i.e. the message sent from the Peer node 1 to Peer node 2. In Fig. 10, the Node 1 is denied to send information to Node 2.

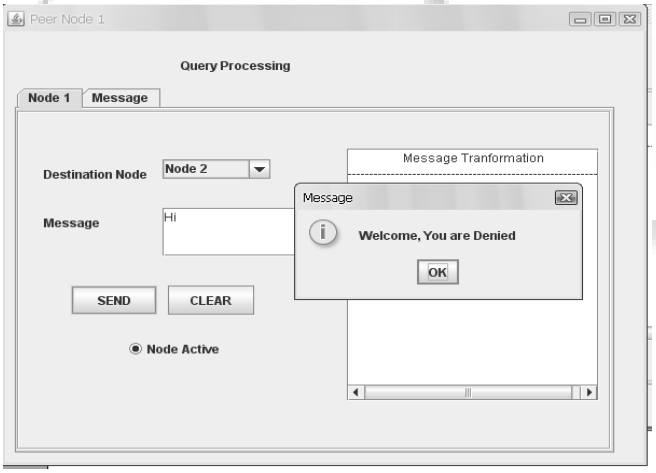

Fig. 10: Access Denied for Node 1

From the following Fig 11(a & b), the risk value is assessed for network constraints, in order to Allow or Deny data. If the risk value is maximum, it will *Deny* and if the value is minimum it will *Allow* action. [14]

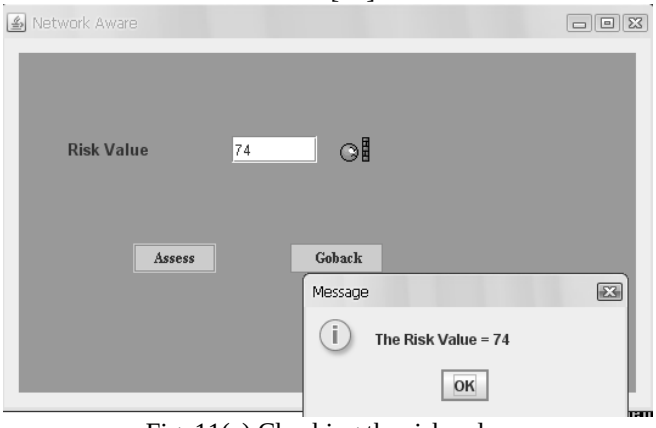

Fig. 11(a) Checking the risk value

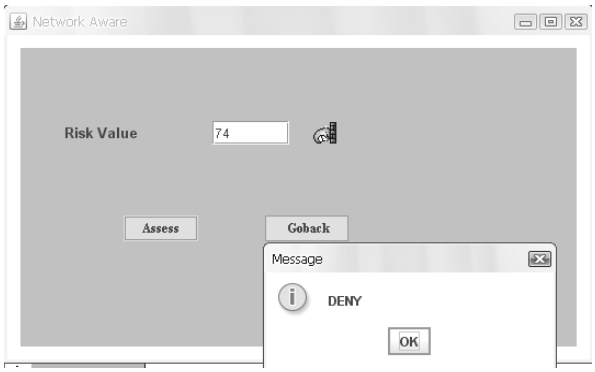

Fig. 11(b) Risk value is high = Deny

Fig 12. Shows the policy creation that are given as additional rules to the firewall in order to make it as much as efficient.[15]

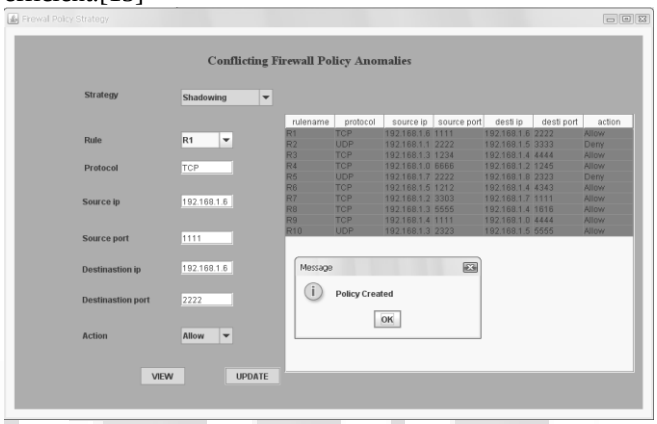

Fig. 12: Policy creation

Shadowing - Here the updated firewall anomaly detection for shadowing is shown in Fig. 13 a

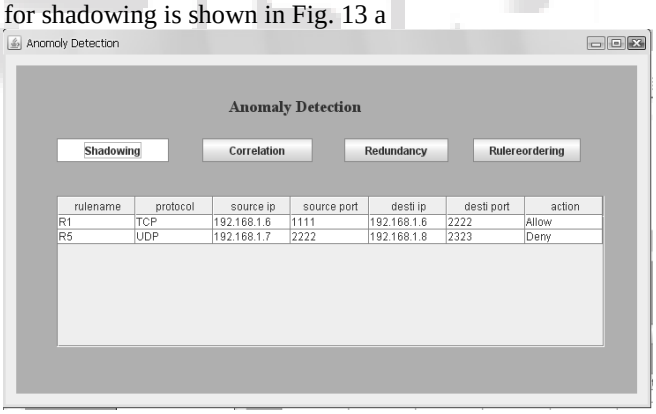

Fig. 13(a) : Shadowing

Correlation - Here the updated firewall anomaly detection for correlation in Fig 13.b[16]

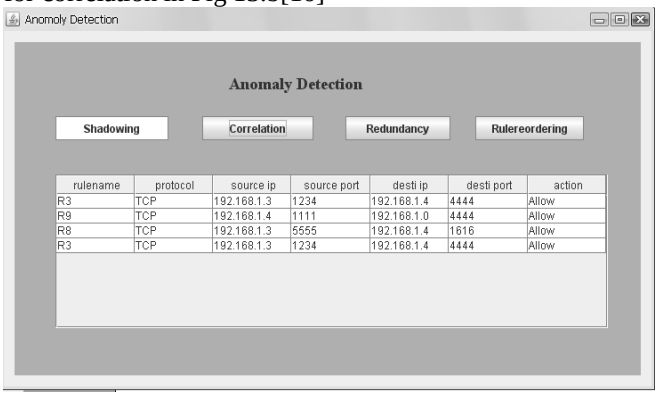

Fig. 13(b): Correlation

Redundancy - Here the updated firewall anomaly detection for redundancy is shown in Fig 14.a.

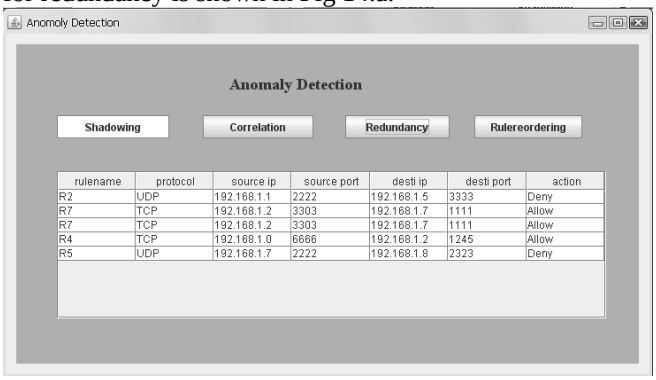

Fig. 14(a) : Redundancy

Rule Reordering - Finally in this image the updated firewall anomaly detection for rule reordering is shown in Fig 14.b.

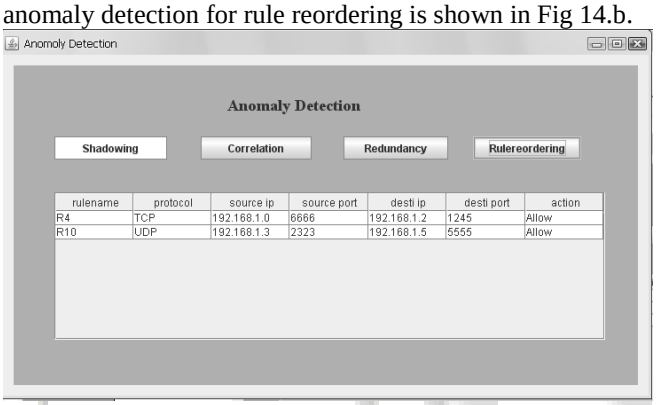

Fig. 14(b): Rule Reordering

V. CONCLUSION AND FUTURE ENHANCEMENT

A novel anomaly management framework that facilitates automatic rule generation, detection and resolution of firewall policy anomalies in distributed environment was proposed. A rule-based segmentation mechanism is introduced to achieve the goal of effective and efficient anomaly analysis. Anomaly management environment called FAME is adapted and demonstrated in distributed firewalls. The proposed anomaly analysis methodology is practical and helpful for system administrators to enable an assurable network management of organization in distributed environment.

Future work includes usability studies to evaluate functionalities and system requirements of the policy visualization approach with subject matter experts. Moreover, the anomaly management framework and visualization approach can be applied to other types of access control policies.

REFERENCES

[1] E Hongxin Hu, Gail-Joon Ahn and Ketan Kulkarni, "Detecting and Resolving Firewall Policy Anomalies", IEEE transactions on dependable and secure computing, vol. 9, no. 3, May/June 2012.

[2] Bin Hamid Ali. F.A, "A study of Technology in Firewall System", Business, Engineering and Industrial Applications (ISBEIA), 2011 IEEE Symposium.

[3] Yi Yin, Xiaodong Xu, Katayama.Y and Takahashi.N, "Inconsistency Detection System for Security Policy and Firewall Policy", Networking and Computing (ICNC), 2010 First International Conference.

[4] Kotenko.I, Verification of security policy filtering rules by Model Checking, Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2011 IEEE 6th International Conference.

[5] Wool.A, "Trends in Firewall Configuration Errors" Internet Computing, IEEE 2010.

[6] Hu. H, Ahn. G and Kulkarni. K "Anomaly discovery and resolution in web access control policies", Access Control Models and Technologies, 2011 IEEE 16th International symposium

[7] J. Alfaro, N. Boulahia-Cuppens, and F. Cuppens, "Complete analysis of configuration rules to guarantee reliable network security policies," International Journal of Information Security, vol.7, no. 2, pp. 103–122, 2008.

[8] F. Baboescu and G. Varghese, "Fast and scalable conflict detection for packet classifiers," Computer Networks, vol. 42, no. 6, pp. 717– 735, 2003

[9] 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, p.15, 2006.

a. H. Hu, G. Ahn, and K. Kulkarni, "Anomaly Discovery and Resolution in Web Access Control Policies," Proc. 16th ACM Symp. Access Control Models and Technologies, pp. 165-174, 2011.

[10] El-Atawy, K. Ibrahim, H. Hamed, and E. Al-Shaer, "Policy Segmentation for Intelligent Firewall Testing," Proc. First Workshop Secure Network Protocols (NPSec '05), 2005.

[11] G. Misherghi, L. Yuan, Z. Su, C.-N. Chuah, and H. Chen, "A General Framework for Benchmarking Firewall Optimization Techniques," IEEE Trans. Network and Service Management, vol. 5, no. 4, pp. 227-238, Dec. 2008.

[12] M. Frigault, L. Wang, A. Singhal, and S. Jajodia, "Measuring Network Security Using Dynamic Bayesian Network," Proc. Fourth ACM Workshop Quality of Protection, 2008.

[13] M. Sahinoglu, "Security Meter: A Practical Decision-Tree Model to Quantify Risk," IEEE Security and Privacy, vol. 3, no. 3, pp. 18-24, May 2005.

[14] R. Sawilla and X. Ou, "Identifying Critical Attack Assets in Dependency Attack Graphs," Proc. 13th European Symp. Research in Computer Security (ESORICS), 2008.

[15] X. Ou, W. Boyer, and M. McQueen, "A Scalable Approach to Attack Graph Generation," Proc. 13th ACM Conf. Computer and Comm. Security, pp. 336-345, 2006.

