IJCAT International Journal of Computing and Technology, Volume 1, Issue 3, April 2014
ISSN : 2348 - 6090
www.IJCAT.org

Firewall Policy Modelling and Anomaly Detection

[1] Suhail Ahmed

[1] Computer Science & Engineering Department, VTU University,
SDIT, Mangalore, Karnataka. India

Abstract - In this paper an anomaly management framework for firewalls based on a rule-based segmentation technique is presented to facilitate not only more accurate anomaly detection but also effective anomaly resolution. Based on this 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 among those rules. In particular, we articulate a grid-based representation technique, providing an intuitive cognitive sense about policy anomaly.

Keywords **- Firewall, Policy anomaly management.**

1. Introduction

A firewall in an information security program is similar to a building's firewall, in that it prevents specific types of information from moving between the outside world, known as the un trusted network (ex. the Internet) and the inside world, known as the trusted network. 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. Al-Shaer and Hamed [1] reported that their firewall policies contain anomalies even though several administrators including nine experts maintained those policies. In addition, Wool [2] recently inspected firewall policies collected from different organizations and indicated that all examined firewall policies have security flaws.

The process of configuring a firewall is tedious and error prone. Therefore, effective mechanisms and tools for policy management are crucial to the success of firewalls. Recently, policy anomaly detection has received a great deal of attention [1], [4], [5], [6]. Corresponding policy analysis tools, such as Firewall Policy Advisor [1] and FIREMAN [5], with the goal of detecting policy anomalies have been introduced. Firewall Policy Advisor only has the capability of detecting pair wise anomalies in firewall rules. FIREMAN can detect anomalies among multiple rules by analyzing the relationships between one rule and the collections of packet spaces derived from all preceding rules. However, FIREMAN also has limitations in detecting anomalies [4].

On the other hand, due to the complex nature of policy anomalies, system administrators are often faced with a more challenging problem in resolving anomalies, in particular, resolving policy conflicts. 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.

Since the policy conflicts in firewalls always exist and are hard to be eliminated, a practical resolution method is to identify which rule involved in a conflict situation should take precedence when multiple conflicting rules (with different actions) can filter a particular network packet simultaneously. To resolve policy conflicts, a firewall typically implements a first-match resolution mechanism based on the order of rules. In this way, each packet processed by the firewall is mapped to the decision of the first rule that the packet matches.

However, 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 precedence 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 precedence. This situation can cause severe network breaches such as permitting harmful packets to sneak into a private network, or dropping legal traffic which in turn could encumber the availability and utility of network services. Obviously, it is necessary to seek a way to bridge a gap between conflict detection and conflict resolution with the first-match mechanism in firewalls.

In this paper, a novel anomaly management framework for firewalls based on a rule-based segmentation technique is implemented to facilitate not only more accurate anomaly detection but also effective anomaly

resolution. Based on this 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 (either conflicting or redundant) among those rules. We also introduce a flexible conflict resolution method to enable a fine-grained conflict resolution with the help of several effective resolution strategies with respect to the risk assessment of protected networks and the intention of policy definition.

2. Overview of Anomalies in Firewall Policies

A firewall policy consists of a sequence of rules that define the actions performed on packets that satisfy certain conditions. The 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. Table 1 shows 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.

Table 1: An Example for Firewall Policy

Rule	Protocol	Src IP	Src Port	Dst IP	Dst Port	Action
r1	UDP	10.1.2.*	*	172.32.1.*	53	deny
r2	UDP	10.1.*.*	*	172.32.1.*	53	deny
r3	UDP	10.1.*.*	*	192.168.1.*	53	allow
r4	UDP	10.1.1.*	*	192.168.1.*	53	deny
r5	UDP	10.1.1.*	*	*	53	allow

Several related work has categorized different types of firewall policy anomalies [1], [5]. Based on following classification, we articulate the typical firewall policy anomalies:

2.1 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, 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.

2.2 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, r_5 is a generalization of r_4 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.

2.3 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.

2.4 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.

3. Structure and Technique

3.1 Packet Space Segmentation and Classification

In order to precisely identify policy anomalies and enable a more effective anomaly resolution, we introduce a rule-based segmentation technique, which adopts a binary decision diagram (BDD)-based data structure to represent rules and perform various set operations, to convert a list of rules into a set of disjoint network packet spaces. This technique has been recently introduced to deal with several research problems such as network traffic measurement [9], firewall testing [10] and optimization [11]. 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: subset (line 5), superset (line 10), partial match (line 13), or disjoint (line 17). Therefore, one can utilize set operations to separate the overlapped spaces into disjoint spaces.

Algorithm 1: Segment Generation for Network Packet Space of a Set of Rule R: Partition(R)
Input: A set of rules, R
Output: A set of packet space segment, S

1. **for each** r ϵ R **do**
2. s_r ⟵ PacketSpace(r);
3. **for each** s ϵ S **do**
4. **/*** s_r *is a subset of* s */*
5. **If** s_r C s **then**
6. S.Append(s/s_r);

7. $s \leftarrow s_r$;
8. Break;
9. /* s_r *is a superset of* s */
10. **Else if** $s_r \supset s$ **then**
11. $s_r \leftarrow s_r \setminus s$;
12. /* s_r *is a partially matches* s */
13. **Else if** $s_r \cap s \neq \phi$ **then**
14. S.Append(s/s_r);
15. $s \leftarrow s_r \cap s$;
16. $s_r \leftarrow s_r \setminus s$;
17. S.Append(s_r);
18. **Return** S;

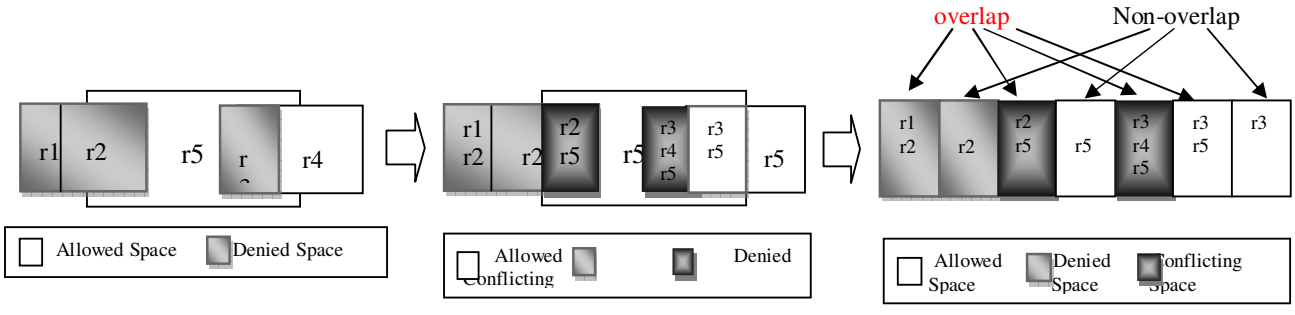
Fig.1a gives the two-dimensional geometric representation of packet spaces derived from the example policy shown in Table 1. We utilize colored rectangles to denote two kinds of packet spaces: allowed space (white color) and denied space (gray color), respectively. In this example, there are two allowed 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, r5 overlaps with r2, r3, and r4, 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 addressed in Algorithm 1 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). Fig. 1b demonstrates the segments of packet spaces derived from the example policy. Since 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 Fig. 1c. We can notice that seven unique disjoint segments are generated. 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.

3.2 Grid Representation of Policy Anomaly

To enable an effective anomaly resolution, complete and accurate anomaly diagnosis information should be represented in an intuitive way. When a set of rules interacts, one overlapping relation may be associated with several rules. Meanwhile, 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 associated rules can be viewed in the uniform representation of anomalies (Fig. 1c). However, 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, 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.

Fig.2 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 associated with a rule. For example, as shown in Fig. 2, we can notice that a conflicting segment (CS) s5, which points out a conflict, is related to a rule set consisting of three conflicting rules r3, r4, and r5 (highlighted with a horizontal red rectangle), and a rule r3 is involved in three segments s5, s6, and s7 (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.



(a) Two dimensional geometric representation of overlapping rules

(b) Packet Space segmentation

(c) Uniform representation

Fig. 1[3] Packet space representation derived from the example policy

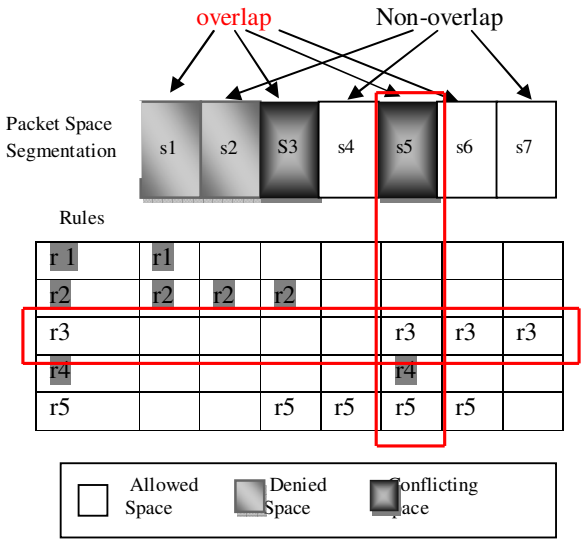


Fig. 2[3] Grid representation of policy anomaly

3.3 Strategy Based Conflict Resolution Algorithm

As shown in Fig 3, here we are using a strategy based conflict resolution algorithm where initially the threshold of the system is been initialized. The threshold of the packet which the user wants to send in been calculated, if the threshold generated is greater than the threshold of the system then the action performed is "deny", if the threshold generated is lesser than the threshold of the system then the action performed is "allow", if the threshold generated is equal to the threshold of the system then the system should ask the user what actions should be performed whether "allow or deny".

4. Conclusion

In this paper a novel anomaly management framework that facilitates systematic detection and resolution of firewall policy anomalies is proposed. A rule-based segmentation mechanism and a grid-based representation technique were introduced to achieve the goal of effective and efficient anomaly analysis.
Future work includes usability studies to evaluate functionalities and system requirements of our policy visualization approach with subject matter experts. Also, we would like to extend our anomaly analysis approach to handle distributed firewalls. Moreover, we would explore how our anomaly management framework and visualization approach can be applied to other types of access control policies.

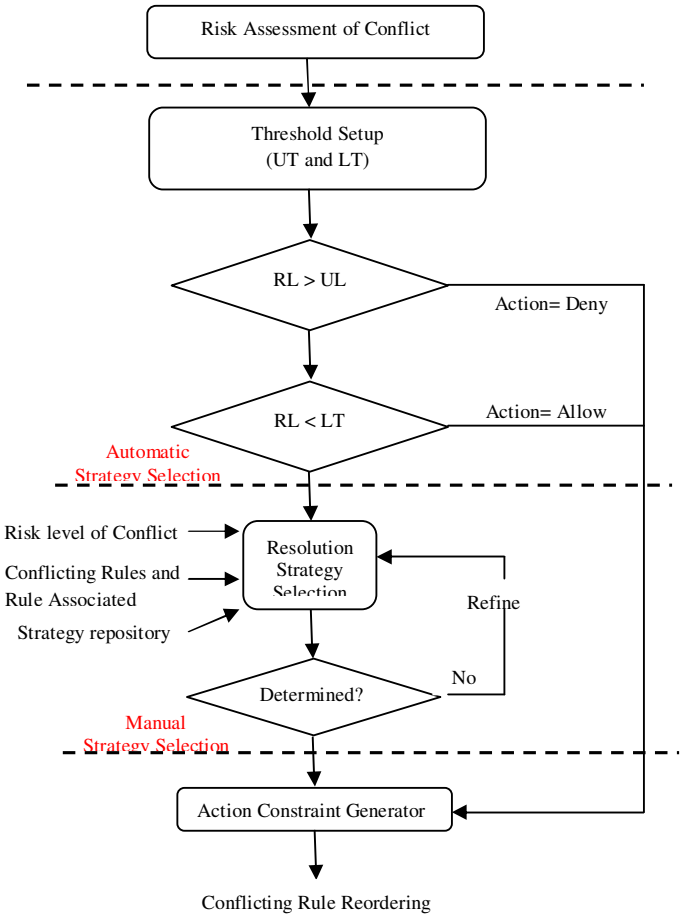


Fig 3[3] Strategy Based Conflict Resolution Algorithm

References

[1] E. Al-Shaer and H. Hamed, "Discovery of Policy Anomalies in Distributed Firewalls," IEEE INFOCOM '04, vol. 4, pp. 2605-2616, 2004.

[2] A. Wool, "Trends in Firewall Configuration Errors: Measuring the Holes in Swiss cheese," IEEE Internet Computing, vol. 14, no. 4, pp. 58-65, July/Aug. 2010.

[3] 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

[4] J. Alfaro, N. Boulahia-Cuppens, and F. Cuppens, "Complete Analysis of Configuration Rules to Guarantee Reliable Network Security Policies," Int'l J. Information Security, vol. 7, no. 2, pp. 103- 122, 2008.

[5] F. Baboescu and G. Varghese, "Fast and Scalable Conflict Detection for Packet Classifiers," Computer Networks, vol. 42, no. 6, pp. 717-735, 2003.

[6] 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.

[7] E. Lupu and M. Sloman, "Conflicts in Policy-Based Distributed Systems Management," IEEE Trans. Software Eng., vol. 25, no. 6, pp. 852-869, Nov./Dec. 1999.

[8] I. Herman, G. Melanc¸on, and M. Marshall, "Graph Visualization and Navigation in Information Visualization: A Survey," IEEE Trans. Visualization and Computer Graphics, vol. 6, no. 1, pp. 24-43, Jan.-Mar. 2000.

[9] 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.

Suhail Ahmed. Is an M. Tech Student of Computer Science & Engineering Department of SDIT, Mangalore. He graduated with BE (Honours') in Computer Science and Engineering from VTU University, Belgaum. He is Currently pursuing his Masters.