ANOMALY DISCOVERY AND RESOLUTION IN DISTRIBUTED FIREWALL POLICY

V. Archana
PG Scholar, Department of CSE, PSN Engineering College, Tirunelveli
Email: archanavivek121@gmail.com.

T. Kumesh
Assistant Professor, Department of CSE, PSN Engineering College, Tirunelveli

Abstract - 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. In particular, we articulate a grid-based representation technique, providing an intuitive cognitive sense about policy anomaly. 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 experience.

Index Terms - Firewall, policy anomaly management, access control, visualization tool.

1. Introduction

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 and system environments. For instance, 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], [3], [4], [5].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 [3]. For each firewall rule, FIREMAN only examines all preceding rules but ignores all subsequent rules when performing anomaly analysis. In addition, each analysis result from FIREMAN can only show that there is a misconfiguration between one rule and its preceding rules, but cannot accurately indicate all rules involved in an anomaly.

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. 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.

2. Related Works

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

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, r2 correlates with r5, 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 r2 is a preceding rule of r5, every packet within the intersection of these rules is denied by r2. 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, r1 is redundant with respect to r2 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 r1 can match r2 as well with the same action. Anomaly detection algorithms and corresponding tools were introduced by [1], [5] as well. However, existing conflict classification and detection approaches only treat a policy conflict as an inconsistent relation between one rule and other rules. Given a more general definition on policy conflict as shown in Definition 1, we believe that identifying policy conflicts should always consider a firewall policy as a whole piece, and precise indication of the conflicting sections caused by a set of overlapping rules is critical for effectively resolving the conflicts.

Definition 1 (Policy Conflict): *A policy conflict pc in a firewall F is associated with a unique set of conflicting firewall rules cr ¼ fr1; . . . ;rng, 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.*
Similarly, we give a general definition for rule redundancy in firewall policies as follows, which serves as a foundation of our redundancy elimination approach.

Definition 2 (Rule Redundancy): *A rule r is redundant in a firewall F iff the network packet space derived from the resulting policy F0 after removing r is equivalent to the network space defined by F. That is, F and F0 satisfy following equations: SA F ¼ SA F0 and SD F ¼ SD F0 , where SA and SD denote allowed and denied network packet spaces, respectively.*

3. Anomaly Representation Based On Packet Space

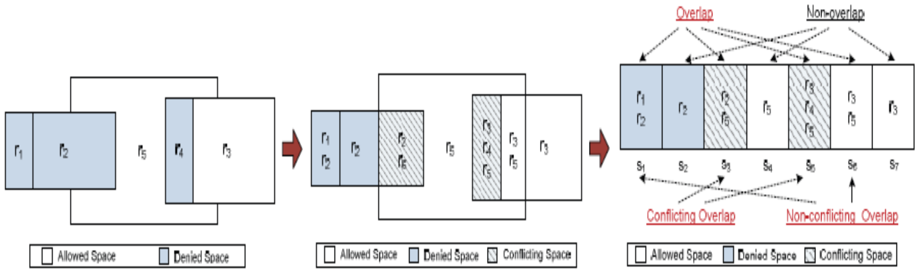
3.1 Packet Space Segmentation and Classification

Existing anomaly detection methods could not accurately point out the anomaly portions caused by a set of overlapping rules. 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]. Inspired by those successful applications, we leverage this technique for the purpose of firewall policy anomaly analysis. Algorithm 1 shows the pseudo code of generating packet space segments for a set of firewall rules R.2 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.
For the purposes of brevity and understandability, we employ a two-dimensional geometric representation for each packet space derived from firewall rules. Note that a firewall rule typically utilizes five fields to define the rule condition; thus, a complete representation of packet space should be multidimensional. 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.

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 segments, $S$.
foreach $r \in R$ do
    $s_r \longleftarrow PacketSpace(r)$;
    foreach $s \in S$ do
        /* $s_r$ is a subset of $s$*/
        if $s_r \subset s$ then
            $S.Append(s \setminus s_r)$;
            $s \longleftarrow s_r$;
            $Break$;
        /* $s_r$ is a superset of $s$*/
        else if $s_r \supset s$ then
            $s_r \longleftarrow s_r \setminus s$;
        /* $s_r$ partially matches $s$*/
        else if $s_r \cap s \neq \emptyset$ then
            $S.Append(s \setminus s_r)$;
            $s \longleftarrow s_r \cap s$;
            $s_r \longleftarrow s_r \setminus s$;
    $S.Append(s_r)$;
return $S$;
```

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.



(a) Two dimensional geometric representation of overlapping rules

(b) Packet space segmentation

(c) Uniform representation

Fig. 1. Packet space representation derived from the example policy.

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.

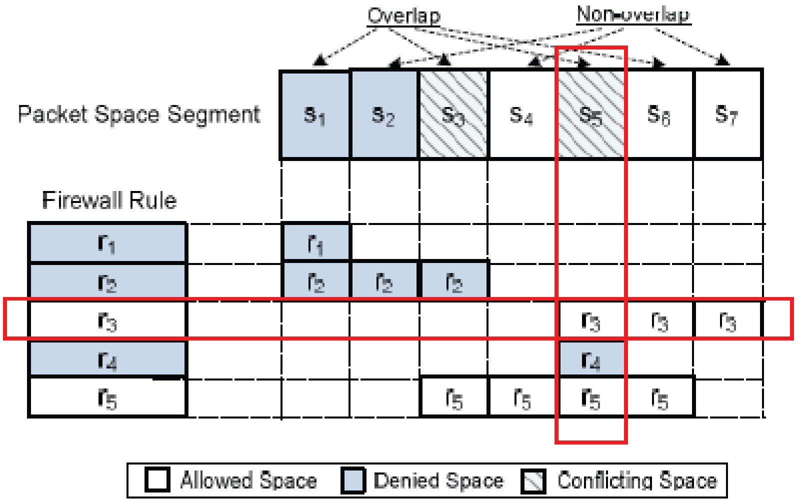


Fig. 2. Grid representation of policy anomaly.

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.

4. Anomaly Management Framework

Our policy anomaly management framework is composed of two core functionalities: conflict detection and resolution, and redundancy discovery and removal, as depicted in Fig. 3. 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 (CG) are derived. Policy conflicts belonging to different conflict correlation groups can be resolved separately; thus, the searching space for resolving conflicts is reduced by the correlation process. The second step generates an action constraint for each conflicting segment by examining the characteristics of each conflicting segment. A strategy-based method is introduced for generating action constraints.

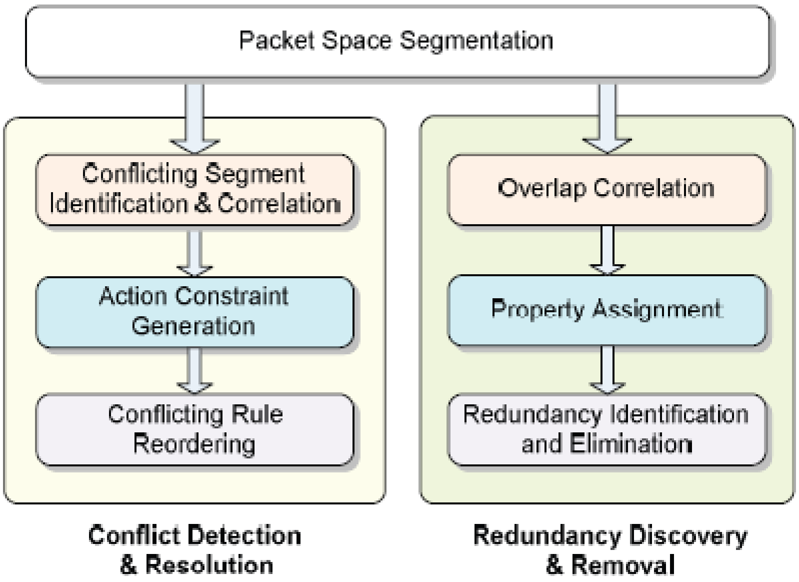

Packet Space Segmentation
Conflicting Segment Identification & Correlation
Action Constraint Generation
Conflicting Rule Reordering
Overlap Correlation
Property Assignment
Redundancy Identification and Elimination
Conflict Detection & Resolution
Redundancy Discovery & Removal

Fig. 3. Policy anomaly management framework.

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. Regarding redundancy discovery and removal, segment correlation groups are first identified. Then, the process of property assignment is performed to each rule's subspaces. Consequently, redundant rules are identified and eliminated.

5. Conclusion

In this paper, we have proposed a novel anomaly management framework that facilitates systematic detection and resolution of firewall policy anomalies. A rule-based segmentation mechanism and a grid-based representation technique were introduced to achieve the goal of effective and efficient anomaly analysis. In addition, we have described a proof-of-concept implementation of our anomaly management environment called FAME and demonstrated that our proposed anomaly analysis methodology is practical and helpful for system administrators to enable an assurable network management.
Our 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.

REFERENCES

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

[2] Wool, A., "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] Yuan, L., Chen, H., Mai, J., Chuah, C., Su, Z., Mohapatra, P., and Davis, C., "Fireman: A Toolkit for Firewall Modeling and Analysis,"Proc. IEEE Symp. Security and Privacy, p. 15, 2006.

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

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

[6] A. 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.

[7] Mayer, A., Wool, A., and Ziskind, E., "Fang: A Firewall Analysis Engine," Proc. IEEE Symp. Security and Privacy, pp. 177-189, 2000.

[8] Gouda, M., and Liu, X., "Firewall Design: Consistency, Completeness, and Compactness," Proc. 24th Int'l Conf. Distributed Computing Systems (ICDCS '04), p. 327, 2004.

[9] Ioannidis, S., Keromytis, A., Bellovin, S., and Smith, J., "Implementing a Distributed Firewall," Proc. Seventh ACM Conf. Computer and Comm. Security, p. 199, 2000.

[10] Hari, A., Suri, S., and Parulkar, G., "Detecting and Resolving Packet Filter Conflicts," Proc. IEEE INFOCOM, pp. 1203-1212, 2000.

[11] Fu, Z., Wu, S., Huang, H., Loh, K., Gong, F., Baldine, I., and Xu, C., "IPSec/VPN Security Policy: Correctness, Conflict Detection and Resolution," Proc. Int'l Workshop Policies for Distributed Systems and Networks (POLICY '01), pp. 39-56, 2001.