ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

A Study on Firewall Policy Anomaly Representation Techniques

Lubna K[1], Robin Cyriac[2]

M.Tech Scholar, Dept of CSE, Rajagiri School of Engineering and Technology, Kochi, India[1]

Asst. Professor, Dept of CSE, Rajagiri School of Engineering and Technology, Kochi, India[2]

ABSTRACT: A firewall is a system that acts as an interface between private network and a public network. It implements the security policy based on the rules defined by the network administrator; which decides the packets can be allowed or blocked to the organization's private network. Manual definition of rules often results in anomalies in the policy. Existing research on this problem have been focused on analysis and detection of firewall policy anomalies. This paper discusses about two major firewall policy anomaly representations that is policy tree representation and a rule-based segmentation mechanism which uses grid-based representation. This grid-based segmentation mechanism overcomes some limitations of policy tree representation.

Keywords: Firewall Policy; Policy Anomaly Management; Policy Tree; Segmentation Technique; Grid Representation

I. INTRODUCTION

A firewall keeps a network secure by controlling the traffic to/from a network. It can be either software-based or hardware-based. Firewall monitors the suspicious and unauthorized traffic to Internet-based enterprises. A set of rules are defined by the system administrators to filter the incoming and outgoing packets by allowing or denying them. Due to the complexity and inter dependency of the policy rules, firewall policy management has become a challenging task.

Recently, policy anomaly management is getting much attention. Policy anomaly management tools such as FIREMAN [2] and Firewall Policy Advisor [3] have been introduced for policy management in firewalls. Firewall policy advisor can detect only pair wise anomalies in firewall rules. And FIREMAN can detect anomalies in multiple rules by analyzing the relationships between one rule and a collection of packet spaces derived from all preceding rules. Thereby firewall policy management of FIREMAN is incomplete and can only show there is some misconfiguration between one rule and its preceding rules, and cannot accurately point out all rules involved in the anomaly.

Due to complex nature of policy anomalies, resolving anomalies is a more challenging problem for system administrators.While defining the filtering facility, in order to determine the proper rule ordering and guarantee correct security policy semantics, great attention has to be given to rule relations and interactions. The complexity in writing a new rule or modifying an existing rule increases with more number of filtering rules. Also, changing the conflicting rules is more difficult due to large number of conflict rules and its complex nature. That is, one conflict may be associated with multiple rules and one rule may be associated with several conflicts. Besides, there will be more than one administrator to maintain firewall policies deployed in a network.

Therefore, an effective policy management technique and tool is needed that enable network administrators to analyze, verify and purify the correctness of written firewall rules. Based on the order of the rules, a firewall typically implements a first-match resolution mechanism. Thereby, each packet is mapped to the decision of the first rule that it matches. But, applying this first-match rule has limitations. That is, when the existing first match rule is not the desired rule to take precedence, then there is more chance of a conflict.

A novel anomaly management framework for firewalls based on a rule-based segmentation technique [1] provides more accurate anomaly detection and also effective anomaly resolution. In this technique, a network packet space is divided into a set of disjoint packet space segments. Each segment is associated with a unique set of firewall rules that indicates either conflict or redundancy (i.e., overlap relation) among rules. This technique adopts a grid-based visualization approach to represent policy anomaly diagnosis

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

in a better way. That also enables an efficient anomaly management.

This paper is organised as follows: Section II overviews different anomalies in firewall policies. Section III presents anomaly representation based on policy tree and packet space. The paper is concluded in Section IV.

II. OVERVIEW OF FIREWALL POLICY ANOMALIES

TABLE I
AN EXAMPLE FIREWALL POLICY

Rule	Protocol	Source IP	Source Port	Dest. IP	Dest Port	Action
r1	UDP	100.11.2. *	*	162.32.1.*	80	deny
r2	UDP	100.11.*. *	*	162.32.1.*	80	deny
r3	TCP	100.11.*. *	*	192.168.*.*	53	allow
r4	TCP	100.11.1. *	*	192.168.1.*	53	deny
r5	*	100.11.1. *	*	*	*	allow

The firewall policy consists of sequence of rules that define the actions performed on packets so that, it satisfy certain conditions. A rule consists of certain conditions that perform some actions. A condition in a rule comprises a set of fields that can identify specific packets matched by this rule. Table 1 shows an example of firewall policy which includes 5 firewall rules- r1, r2, r3, r4, r5. Several related works [2], [3] categorized different firewall policy anomalies. The typical firewall policy anomalies are:

1. Shadowing: A rule is shadowed when one or more of preceding rules that matches all the packets matched by this rule, in such a way that the shadowed rule is never activated. Shadowing can be considered as a critical error in the policy, because the shadowed rule never takes effect. For example, r4 is shadowed by r3 in Table 1. In r3 packets with tcp protocol having source IP (100.11.*.*) are accepted whereas, in r4 those packets with Source IP (100.11.1.*) is denied. Since, r4 is shadowed by r3; r4 will be neglected by accepting packets with all Source IP.
2. Generalization: A rule is a generalization of one or more of preceding rules if they have different actions and if a subset of packets matched by this rule also matches the preceding rules. For example, r5 is a generalization of r4 in Table 1, implies that all packets coming from the address 100.11.1.* will be accepted, except the tcp packet coming from 100.11.1.* to the port 53 of 192.168.1.*. It is considered just as an anomaly warning because the specific rule makes an exception of the general rule. This might cause blocking of an accepted traffic or a denied traffic to be permitted.
3. Correlation: If a rule intersects with rules but have different action, then this rule is said to be correlated with other rules. Here, the packets matched by the intersection of those rules may be denied by one rule, but permitted by others. For example, r2 is in correlation with r5 in Table 1. The two rules with this ordering imply that all UDP packets coming from any port of 100.11.1.* to the port 80 of 162.32.1.* match the intersection of these. Since, r2 precedes r5, every packet within the intersection of r5 will be denied by r2.
4. Redundancy: A rule is redundant if there is another same or more general rule available that has same action on the same packet such that if the redundant rule is removed, the overall firewall policy will not be affected. For example, r1 is redundant to r2 in Table 1, since all UDP packets coming from any port of 100.11.2.* to the port 80 of 162.32.1.* matched with r1 can match r2 as well with the same action.
5. Irrelevance Anomaly: If a rule cannot match to any traffic that might flow through the network, then the rule is called irrelevant. This happens when the source address and destination address fields of the rule do not match any domain reachable through this firewall.

III. FIREWALL POLICY REPRESENTATION

This paper discusses about two ways of firewall policy representation. One is a single rooted tree or policy tree [3] representation and another one uses packet space segmentation [1]. Fig. 1 represents the policy tree model of the filtering policy given in Table 1. The tree model provides simple representation of the filtering rules and also allows easy discovery of relations and discovery among these rules.

In policy tree, the root node represents the protocol field, the leaf node represents the action field and the intermediate nodes represent the rest of fields in order. A rule is specified by every tree path specified from the root to a leaf, or from a leaf to the root in a policy tree. Every rule should have an

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

action leaf that represents accept or deny of the rule. The dotted box below the leaf represents other rules that are in anomaly with it. The tree represents separate source address branch for both rules r1 and r2 as they share different field value. Whereas rules that have the same field values share same source address branch.

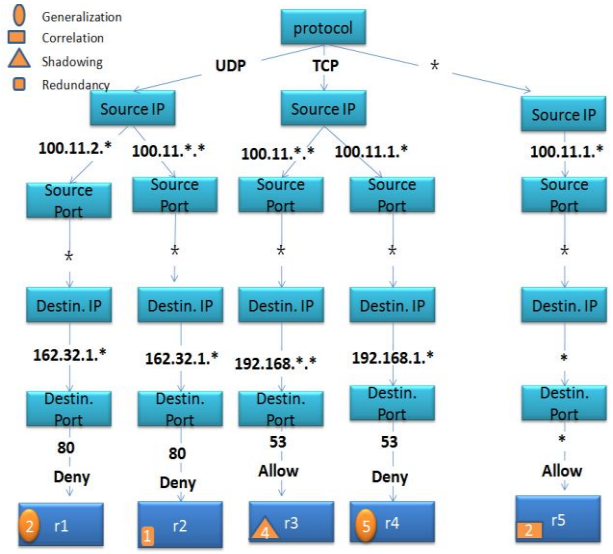

Fig. 1: Policy Tree for the Firewall Policy in Table 1

The policy tree representation technique can represent a policy conflict only as an inconsistent relation between one rule and other rules. In order to precisely identify policy anomalies, a rule-based segmentation technique is used [1], which adopts a binary decision diagram (BDD)-based data structure to represent rules and perform various set operations that converts a list of rules into a set of disjoint network packet spaces. This technique is recently introduced to deal with several research problems such as network traffic measurement [4], firewall testing [5] and optimization [6].

Algorithm 1 given in [1] shows the pseudocode of generating packet space segments for a given 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, superset, partial match, or disjoint. Set operations can be utilized to separate the overlapped spaces into disjoint spaces.

Algorithm 1: Segment Generation for a Network Packet Space of a Set of Rule R: Partition (R)
Input: A set of rules, R ecommended font sizes are shown in Table 1.
Output: A set of packet space segments, S.

1. foreach r ϵ R do
2. $s_r \leftarrow$ PacketSpace(r);
3. foreach 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 ← s_r;
8. break;
9. /* s_r is a superset of s */
10. elseif s_r Ↄ s then
11. $s_r \leftarrow s_r \setminus s$;
12. /* s_r partially matches s */
13. elseif $s_r \cap s \neq \Phi$ then
14. S.Append (s \ s_r)
15. $s_r \leftarrow s_r \cap s$;
16. $s_r \leftarrow s_r \setminus s$;
17. S.Append (s_r);
18. return S;

A set of segments S: {s1,s2,..sn} from firewall rules has the properties:

1. All segments are pairwise disjoint: i.e., $s_i \cap s_k = \Phi$, where $1 \leq i \neq k \leq n$;
2. Any two different network packets p_1 and p_2 within the same segment (si) are matched by the exact same set of rules: GetRule(p_1)= GetRule(p_2), for all $p \epsilon s_i$, where GetRule() returns all matched rules of a network packet.

A two-dimensional geometric representation of each packet space derived from firewall rules is used here, which provides better understandability. Fig. 2a given in [1] provides the two-dimensional geometric representation of firewall rules defined in Table 1.Two spaces overlap when the packets matching corresponding two rules intersect. An overlapping relation may involve more than two rules.

The rule based segmentation technique addressed in Algorithm 1 [1] clearly represents all identical packet spaces derived from a set of overlapping rules. Here the policy segments are classified as: overlapping and nonoverlapping segments. Which is further divided into conflicting overlapping and nonconflicting overlapping segments. Each nonoverlapping segment will be specifying a unique rule and each overlapping segment are related to a set of rules.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 4, April 2013

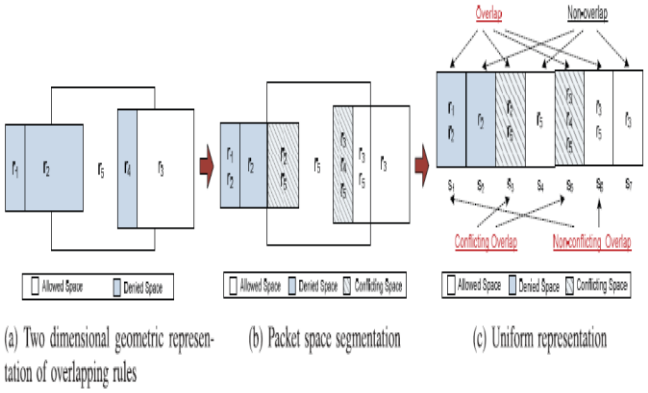

Fig. 2: Packet Space Representation Derived from Example Policy

Fig. 2b [1] demonstrates the segments of packet spaces derived from the example policy in Table 1. In Fig. 2c [1], seven disjoint uniform segments are represented. Here s2, s4 and s7 are nonoverlapping segments and s1, s3, s5 and s6 are overlapping segments. It is still difficult for the administrator to figure out the policy anomalies, that is how many segments one rule is involved in. To satisfy the need for more precise anomaly representation, [1] introduced a grid-representation that is a matrix-based visualization of policy anomalies, in which rules are represented along vertical axis and space segments are displayed along horizontal axis of the matrix. The intersection of a segment and a rule is a grid that displays a rule's subspace covered by the segment.

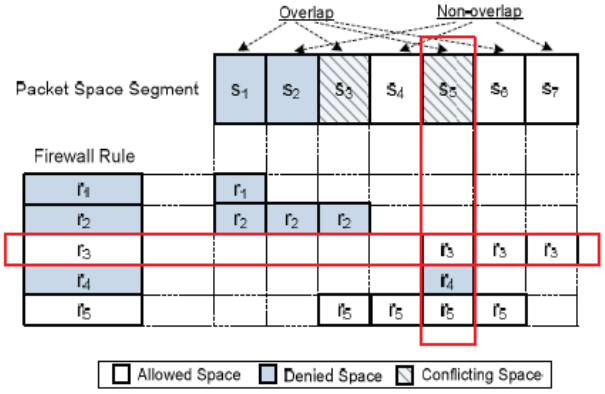

Fig. 3: Grid Representation of Policy Anomaly

A grid representation of policy anomalies for the example policy in Table 1 is shown in Fig. 3 [1]. In this, a conflicting segment (CS), which points out a conflict, is related to a set of conflicting rules r3, r4, and r5 and a rule r3 is involved in segments s5, s6, and s7. So, this grid representation provides a better understanding of policy anomalies to system administrators than with the policy tree representation.

IV. CONCLUSION

This paper provides an insight into two major firewall policy anomaly representation techniques. The policy tree representation provides a hierarchical representation of firewall policy rules. It can represent a policy conflict only as an inconsistent relation between one rule and other rules. Another approach, a rule-based segmentation mechanism and a grid-based representation technique was discussed that can achieve the goal of effective and efficient anomaly analysis. This anomaly analysis approach can be applied in distributed firewalls also.

REFERENCES

[1] H. Hu, G. J. Ahn, K. Kulkarni, "Detecting and Resolving Firewall Policy Anomalies", Proc. IEEE Transactions on Dependable and Secure Computing, Vol. 9, 2012.
[2] 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
[3] E. Al-Shaer and H. Hamed, "Discovery of Policy Anomalies in Distributed Firewalls," IEEE INFOCOM '04, vol. 4, pp. 2605-2616,2004.
[4] L. Yuan, C. Chuah, and P. Mohapatra, "ProgME: Towards Programmable Network Measurement," ACM SIGCOMM Computer Comm. Rev., vol. 37, no. 4, p. 108, 2007
[5] 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
[6] 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

BIOGRAPHY

Lubna K. is currently pursuing her Master of Engineering in Computer Science and Engineering with Specialization in Information Systems from Rajagiri School of Engineering and Technology, Kochi (India) under M.G University.