AN INTRUSION DETECTION SYSTEM FOR MANAGING ANOMALIES BY USING RULE BASED SEGMENTATION

A.krishnakumari*

S.Muthukumar*

***Abstract*—** The growth and development of computing technology in the business Organization and institutions is very effective and admirable. This leads to the requirement of security systems that prevent unauthorized entry. Firewall is one of the growth and development of computing technology in the business competing security mechanisms to secure all the information through the internet of the private network. The effectiveness of the firewall system depends on the configurations of the firewall. 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. And also here we introduce a captcha supported firewall system to check the human access or the programmatic access .it permits only the human user to access the firewall policies with cost effective and less computation time**.**

***Keywords*—**Access control, Captcha, Firewall, Policy anomaly management, Visualization tool,

* IT Department, PSN college of Engineering and Technology, Tirunelveli, India

INTRODUCTION

With the global Internet connection, network security has gained significant attention in research and industrial communities. Due to the increasing threat of network attacks, firewalls have become important elements not only in enterprise networks but also in small-size and home networks. Firewalls have been the frontier defense for secure networks against attacks and unauthorized traffic by filtering out unwanted network traffic coming from or going to the secured network. The filtering decision is based on a set of ordered filtering rules defined according to predefined security policy requirements [1]. 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 pairwise 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.

In this paper we are using FAME tool First, FAME provides a grid-based visualization technique to accurately represent conflict diagnostic information and the detailed information for unresolved conflicts that are very useful, even for manual conflict resolution. Second, FAME resolves conflicts in each conflict correlation group independently. That means a system administrator can focus on analyzing and resolving conflicts belonging to a conflict correlation group individually.

This paper is organized as follows: In section II presents policy maintenance. In section III describes segmentation and classification. In section IV represent the representation of policy anomaly.

In section V, VI, VII we articulate our policy anomaly management framework. Also In section VIII we introduce the captcha technique.

I. POLICY MAINTENANCE

A. Insert rule

It is used for inserting a new rule in the policy by using the following parameters. Rule id**,** Protocol, Source Ip**,** Source port**,** Destination Ip **,** Destination port

B. Update rule

Update rule is used for updating a rule in the policy based on action constraint

C. Load policy

Load policy is used to load the pre -defined policy from anywhere.

D. Delete rule

Delete rule is used to delete the rule from the policy based on action constraint.

II. SEGMENTATION AND CLASSIFICATION (USING RULE BASED SEGMENTATION TECHNIQUE)

A. Subset

A part of a larger group of related things or a set of which all the elements are contained in another set.

B. Super set

A set that includes another set or sets

C. Partial match

Show entries that are similar to the rule you are searching or containing it.

D. Disjoint

Two sets are said to be disjoint if they have no element in common.

The algorithm for rule based algorithm is shown below

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 ∈ R do
    s_r ⟵ PacketSpace(r);
    foreach s ∈ S do
        /* s_r is a subset of s*/
        if s_r ⊂ s then
            S.Append(s \ s_r);
            s ⟵ s_r;
            Break;

        /* s_r is a superset of s*/
        else if s_r ⊃ s then
            s_r ⟵ s_r \ s;

        /* s_r partially matches s*/
        else if s_r ∩ s ≠ ∅ then
            S.Append(s \ s_r);
            s ⟵ s_r ∩ s;
            s_r ⟵ s_r \ s;

    S.Append(s_r);
return S;
```

R =set of rules (r1, r2, r3, r4),r=one rule from the set of rules R,s_r = packet space of r,s= packet space,S = set of packet space segment.

III. REPRESENTATION OF POLICY ANOMALY

A. Overlap

Two spaces overlap when the packets matching two corresponding rules intersect. An overlapping relation may involve multiple rules are Non conflict overlapping have the same action Conflict overlapping which may conflict with each other

B. Non overlap

Each non overlapping segment associates with one unique rule.

IV. ANOMALY MANAGEMENT FRAMEWORK

The block diagram describe the anamoly management framework is shown in Figure 1.

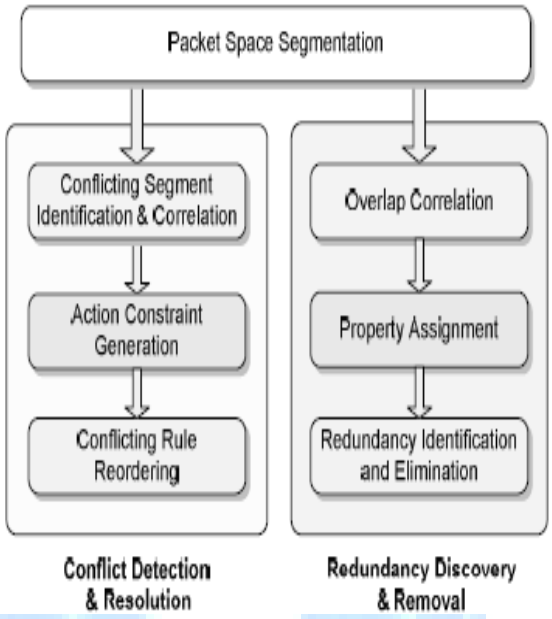

Fig .1. Anamoly management framework

V. CONFLICT DETECTION AND RESOLUTION

A. Conflicting segment identification

Conflicting segments are identified in the first step. Each conflicting segment associates with a policy conflict and a set of conflicting rules.

B. Correlation

We cannot resolve a conflict individually by only reordering conflicting rules associated with one conflict without considering possible impacts on other conflicts. On the other hand, it is also inefficient to deal with all conflicts together by reordering all conflicting rules simultaneously. Therefore, it is necessary to identify the dependency relationships among packet space segments for efficiently resolving policy anomalies.

The major benefit of generating correlation groups 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.

C. Action constraint generation

Our conflict resolution mechanism introduces that an action constraint is assigned to each conflicting segment. An action constraint for a conflicting segment defines a desired action (either Allow or Deny) that the firewall policy should take when any packet within the conflicting segment comes to the firewall. Then, to resolve a conflict, we only assure that the action taken for each packet within the conflicting segment can satisfy the corresponding action constraint.

$$RL(cs) = \frac{\sum_{v \in V(cs)} (CVSS(v) \times IV(s))}{\alpha \times |V(cs)|},$$

C. Conflicting rule reordering

The most ideal solution for conflict resolution is that all action constraints for conflicting segments can be satisfied by reordering conflicting rules. In other words, if we can find out conflicting rules in order that satisfies all action constraints, this order must be the optimal solution for the conflict resolution. Unfortunately, in practice action constraints for conflicting segments can only be satisfied partially in some cases.

VI. REDUNDANCY DISCOVERY AND REMOVAL

Overlap correlation**:** In this step the overlap correlation group is detected. Property assignment: In this step, every rule subspace covered by a policy segment is assigned with a property. Four property values, removable (R), strong irremovable (SI), weak irremovable (WI), and correlated (C), are defined to reflect different characteristics of each rule subspace. Removable property is used to indicate that a rule subspace is removable. Redundancy identification and elimination: The removable rules are identified and removed.

VIII.CAPTCHA

Captcha is a type of challenge-response test used in computing as an attempt to ensure that the response is generated by a person. The process usually involves a computer asking a user to complete a simple test which the computer is able to grade. These tests are designed to be easy for a computer to generate, but difficult for a computer to solve, so that if a correct solution is received, it can be presumed to have been entered by a human. A common type of CAPTCHA requires the user to type letters or digits from a distorted image that appears on the screen, and such tests are commonly used to prevent unwanted internet bots from accessing websites.

IX.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. A common type of CAPTCHA requires the user to type letters or digits from a distorted image that appears on the screen, and such tests are commonly used to prevent unwanted internet bots from accessing websites

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] 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.

[4] F. Baboescu and G. Varghese, "Fast and Scalable Conflict Detection for Packet Classifiers," Computer Networks, vol. 42, no. 6, pp. 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," Proc. IEEE Symp. Security and Privacy, p. 15, 2006.

[6] 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.

[7] 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.

[8] 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.

[9] L. Yuan, C. Chuah, and P. Mohapatra, "ProgME: Towards Programmable Network Measurement," ACM SIGCOMM Computer Comm. Rev., vol. 37, no. 4, p. 108, 2007.

[10] 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.