

Effective Resolution of Firewall Policies

¹B.G.Ratna Kumari, ²N.Tulasi Radha, ³P. Suresh Babu

^{1,2,3}Dept. of CSE, PYDAH College of Engineering & Tech., Vishakapatnam, AP, India

Abstract

Firewalls are a well known and advanced security mechanism which ensure the security of private networks in different businesses organization and institutions. The reliability of security provided by a firewall merely depends on the quality of policy configuration of the firewall. Due to the complex nature of firewall configurations as well as the lack of systematic analysis mechanisms and tools it is very difficult task to manage the anomalies in its policies. Detection and recovery of anomalies in firewall policies became a challenging job for the researchers.

In this paper we have introduced a novel idea to detect and prevent the anomalies from firewall policies. We have adopted a rule-based segmentation technique to identify firewall anomalies policies and derive effective and secure resolutions. We introduced a grid-based representation technique, with an intuitive cognitive sense about policy anomaly. We also discuss the concept of implementation part of a visualization-based firewall policy analysis tool called Firewall Anomaly Management Environment (FAME). In addition, we have shown an experimental result how efficiently our approach can discover and resolve anomalies in firewall policies.

Keywords

Anomaly, Segmentation, Firewall, FIREMAN

1. Introduction

A firewall can either be software-based or hardware-based and is used to help keep a network secure. Its primary objective is to control the incoming and outgoing network traffic by analyzing the data packets and determining whether it should be allowed through or not, based on a predetermined rule set. A network's firewall builds a bridge between an internal network that is assumed to be secure and trusted, and another network, usually an external (inter)network, such as the Internet, that is not assumed to be secure and trusted.

Being a mediator between a private network and the public Internet, a firewall examines all incoming and outgoing packets based on security rules to monitor suspicious traffic and unauthorized access to Internet-based enterprises. To implement a security policy in a firewall, system administrators define a set of filtering rules which is depending on the organizations and institutions.

| Rule | Protocol | Source IP | Source Port | Destination IP | Destination Port | Action |
|----------------|----------|-----------|-------------|----------------|------------------|--------|
| r ₁ | UDP | 10.1.2.* | * | 172.32.1.* | 53 | deny |
| r ₂ | UDP | 10.1.*.* | * | 172.32.1.* | 53 | deny |
| r ₃ | TCP | 10.1.*.* | * | 192.168.*.* | 25 | allow |
| r ₄ | TCP | 10.1.1.* | * | 192.168.1.* | 25 | deny |
| r ₅ | * | 10.1.1.* | * | * | * | allow |

Fig. 1: An Example of Firewall Policies

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 environments.

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 [1-2]. Corresponding policy analysis tools, such as Firewall Policy Advisor [3] and

FIREMAN [4-5], 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 [6]. For each firewall rule, FIREMAN only examines all preceding rules but ignores all subsequent rules when performing anomaly analysis.

The anomaly detection procedures of FIREMAN are thus incomplete. 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 typically 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. Besides, firewall policies deployed on a network are often maintained by more than one administrator, and an enterprise firewall may contain legacy rules that are designed by different administrators.

Thus, without a priori knowledge on the administrators' intentions, changing rules will affect the rules' semantics and may not resolve conflicts correctly. Furthermore, in existing cases, a system administrator may intentionally introduce certain overlaps in firewall rules knowing that only the first rule is important. In reality, this is a commonly used technique to exclude specific parts from a certain action, and the proper use of this technique could result in a fewer number of rules [7-8]. In this case, conflicts are not an error, but intended, which would not be necessary to be changed.

In this paper We introduced a grid-based representation technique, with an intuitive cognitive sense about policy anomaly which can directly reflect on the rules on the firewall.

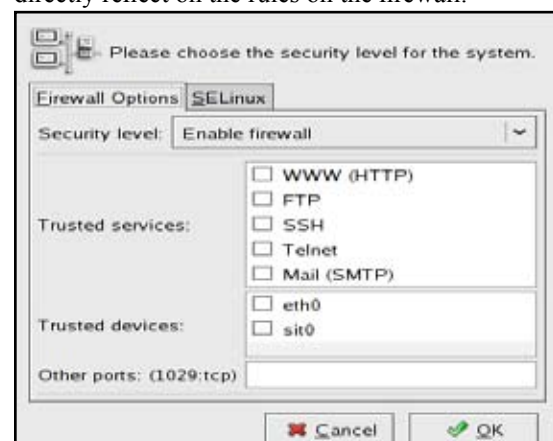


Fig. 2: Basic Firewall Configuration

II. Firewall Policies and Anomalies

Two rules in a firewall policy may overlap, which means one packet may match both rules. Moreover, two rules in a firewall may conflict, implying that those two rules not only overlap each other but also take different actions. Policy conflicts may lead to both security problems (e.g. allowing malicious traffic) and availability problems (e.g. denying legitimate traffic), and policy redundancies will affect the performance of a firewall. A comprehensive classification of policy anomalies (miss configurations) has been articulated by several related work [9-10]. Following existing classification, we summarize policy anomalies as follows:

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_1 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_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_1 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. **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_1 , 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_5 is a preceding rule of r_5 , every packet within the intersection of these rules is denied by r_1 . However, if their positions are swapped, the same packets will be allowed. **24. 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_5 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_5 can match r_1 as well with the same action.

Anomaly detection algorithms and corresponding tools were also introduced by [11,12]. However, prior work only treated 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 rule set involved in a conflict is critical for effectively resolving the conflict.

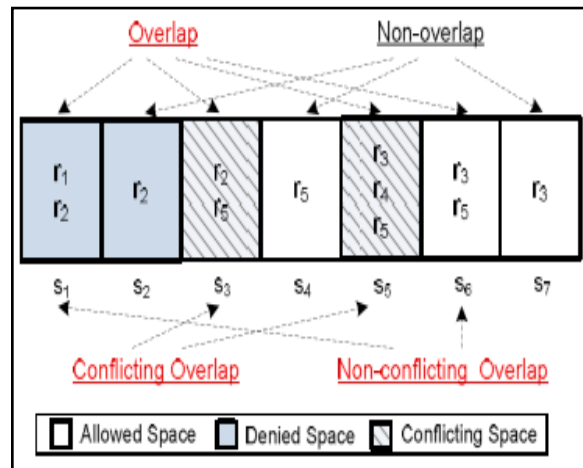


Fig. 3: Uniform Representation

Segment Generation Algorithm

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 [11], firewall testing [12] and optimization [13].

Algorithm-1 Segmentation algorithm

Input: A set of rules, R .

Output: A set of packet space segments, S .

```

foreach  $r \in R$  do
     $s_r \leftarrow PacketSpace(r)$ ;
    foreach  $s \in S$  do
        /*  $s_r$  is a subset of  $s$  */
        if  $s_r \subset s$  then
             $S.Append(s \setminus s_r)$ ;
             $s \leftarrow s_r$ ;
            Break;

        /*  $s_r$  is a superset of  $s$  */
        else if  $s_r \supset s$  then
             $s_r \leftarrow s_r \setminus s$ ;

        /*  $s_r$  partially matches  $s$  */
        else if  $s_r \cap s \neq \emptyset$  then
             $S.Append(s \setminus s_r)$ ;
             $s \leftarrow s_r \cap s$ ;
             $s_r \leftarrow s_r \setminus s$ ;

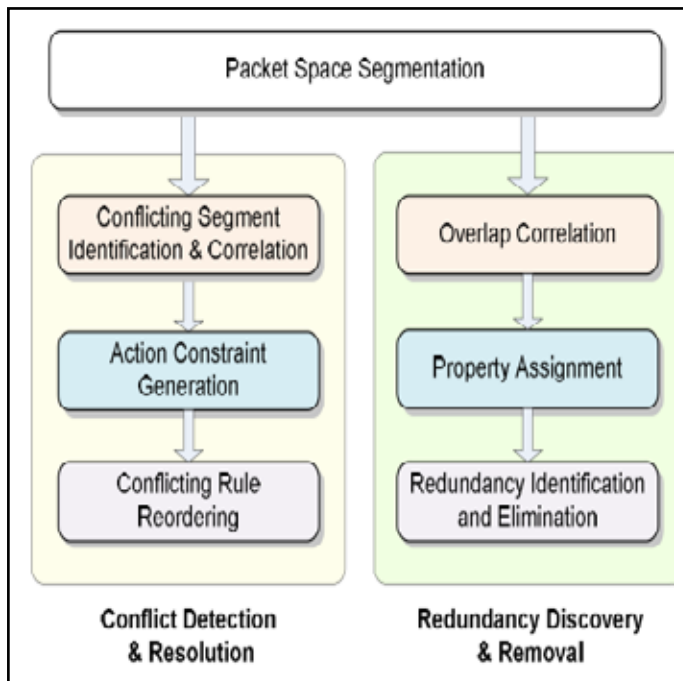
     $S.Append(s_r)$ ;

return  $S$ ;
  
```

III. Anomaly Representation

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. 1(c)). 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.



Modules

1. Correlation of Packet Space Segment
2. Action Constraint Generation
3. Rule Reordering
4. Data Package

1. Correlation of Packet Space Segment

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.

2. Action Constraint Generation

In a firewall policy 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

3. 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.

4. 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.

IV. Anomaly Discovery and Resolution Analysis

We provide two algorithms for resolving these anomalies.

A. Detection Algorithms

For detection of each anomaly, we must prove the correctness of the appropriate theorem. For proving the simple anomalies, each couple of propositions is selected.

A simple anomaly is discovered as long as the appropriate theorem for the couple of propositions is satisfied. For proving a total anomaly, we will select a proposition and a set of propositions as input for appropriate theorem. Since the anomaly theorems are specified with logical formulas, we design a proof assistant to prove these theorems. This proof assistant is designed for satisfying a propositional logic formula. A successful idea for proving propositional formulas that comes from semantics of the logic is that of binary decision diagrams, or BDDs [27-28]. We might say that they are a recent invention, as the originator of BDDs as we know them today was Randall E. Bryant in 1986 [26].

In our proof assistant, we use the BDD idea. Figure 1 shows the state machine for anomaly detection in a security policy based on our definition for each anomaly. This state machine can satisfy detection of total anomalies. The state machine is started for two propositions. For example, we apply this machine for policy insertion time and analyze anomalies for new security proposition with all propositions in the policy.

At the start state of the machine, two actions are compared. We must search for redundancy anomaly as long as the actions of two propositions are equivalent; otherwise we check three anomalies shadow, generalization and correlation.

The order of detection of the three anomalies is important for optimization of analysis process. When the two propositions have the same action for analyzing redundant anomaly. In this situation, if two propositions are not redundant, they have not any anomaly and the state machine changes to "No Anomaly" state.

If the two propositions have different actions. We run the detection process in policy insertion process. In this time, the detection algorithms will be invoked for new propositions and all propositions in database and also for new proposition and all subset of propositions in database. Thus we must create logical formulas and use BDD for satisfying them.

Algorithm2, Simple anomaly detection algorithm

```

DetectionSimpleAnomaly ( $P_x$ , order)
 $P_x$  translated to  $C_x, A_x$ 
for each  $P_i$  in SP
   $P_i$  translated to  $C_i, A_i$ 
  if  $order_i > order$  then
     $C_1 = C_i, C_2 = C_x, A_1 = A_i, A_2 = A_x$ 
  else
     $C_1 = C_x, C_2 = C_i, A_1 = A_x, A_2 = A_i$ 
  if not BDD_Satisfy ( $A_1 ? A_2$ ) then
    if BDD_Satisfy ( $C_2 ? C_1$ ) then
      print "Shadow Anomaly ( $P_x, P_i$ )", Return
    if BDD_Satisfy ( $C_1 ? C_2$ ) then
      print "Generalization Anomaly ( $P_x, P_i$ )", Return
    if BDD_Satisfy ( $C_1 ? C_2$ ) then
      print "Correlation Anomaly ( $P_x, P_i$ )", Return
    if BDD_Satisfy ( $C_2 ? C_1$ ) then
      print "Redundancy Anomaly ( $P_x, P_i$ )", Return
  print "There is no Simple Anomaly with existent propositions."
end DetectionSimpleAnomaly

```

Algorithm 3, Anomaly resolving algorithm

```

ResolveSimpleAnomaly (SP)
for each  $P_j$  in SP
   $P_j$  translated to  $C_j, A_j$ 
  for each  $P_i$  in SP except  $P_j$ 
     $P_i$  translated to  $C_i, A_i$ 
    if  $order_i > order_j$  then
       $C_1 = C_i, C_2 = C_j, A_1 = A_i, A_2 = A_j$ 
    else
       $C_1 = C_j, C_2 = C_i, A_1 = A_j, A_2 = A_i$ 
    if not BDD_Satisfy ( $A_1 ? A_2$ ) then
      if BDD_Satisfy ( $C_2 ? C_1$ ) then
        Remove ( $R_2$ ), Return
      if BDD_Satisfy ( $A_1 ? A_2$ ) then
        if BDD_Satisfy ( $C_2 ? C_1$ ) then
          Remove ( $P_2$ ), Return
  end ResolveSimpleAnomaly

```

V. Conclusion

In this paper, we have proposed a novel anomaly management framework algorithm that facilitates systematic detection and resolution of firewall policy anomalies. A rule-based segmentation mechanism is introduced to achieve the goal of effective and efficient anomaly analysis.

In addition, we have implemented 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.

References

[1] E. Al-Shaer, 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] J. Alfaro, N. Boulahia-Cuppens, 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.

[4] F. Baboescu, 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, C. Davis, "Fireman: A Toolkit for Firewall Modeling and Analysis", Proc. IEEE Symp. Security and Privacy, pp. 15, 2006.

[6] E. Lupu, 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, 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, P. Mohapatra, "ProgME: Towards Programmable Network Measurement", ACM SIGCOMM Computer Comm. Rev., Vol. 37, No. 4, pp. 108, 2007.

[10] A. El-Atawy, K. Ibrahim, H. Hamed, 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, 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, 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.



B.G. Ratna Kumari is a student of PYDAH College of Engineering & Technology, Vishakhapatnam. Presently she is pursuing her M.Tech [Computer Science Of Engineering] from this college and she completed her MSc(IS) from PYDAH College of PG Courses, affiliated to Andhra University, Vishakhapatnam in the year of 2003. Her area of interest includes Computer Networks and Object Oriented Programming languages and latest trends and techniques in Computer Science.



N.Tulasi Radha, well known author and excellent teacher. She completed B.E from GITAM college of Engineering affiliated to Andhra University and M.Tech (CSE) from JNTU College of Engineering, Kakinada, is working as Associate Professor and HOD of Computer Science & Engineering, Kaushik College of Engineering. She has 12 years of teaching experience in various engineering colleges. To

her credit couple of publications were published both in national and international conferences /journals. Her area of interest includes Data Warehouse and Data Mining, Information Security, Unix Operating systems and Human Computer Interaction.



Sri.P.Suresh Babu, well known Author and excellent teacher. Received B.Tech(CSE) from Acharya Nagarjuna university, GUNTUR, Andhra Pradesh and M.E(CSE) from Sathyabama university, Chennai is working as Associate Professor Department of Computer Science and Engineering, Kaushik college of Engineering. He is an active member of CSI. He has 14 years of teaching experience in various

engineering colleges and 4 years of industrial experience. To his credit 7 International publications, 2 national publications, 2 International conferences and 4 work shops. His area of Interest includes Compiler Design, Neural Networks and Fuzzy logic.