Anomalies Detection and Recovery in Firewall Policies

[1]G. Ganesh Sriram, [2]B. I. Swarna Sri, [3]N.Rama Devi, [4]K. Gouthami, [5]S. Arjun Kumar
[1,2,3,4,5]Dept. of CSE, Srinivasa Institute of Engineering & Tech., NH-216, Cheyyeru, AP, India

Abstract
Firewalls are a well known and advanced security mechanisms 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

I. 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 administrator's 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 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 [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 misco nfiguration 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.

A. 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 is shadowed by r3 because r3 4 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 r2.

B. Generalization

A rule is a generalization of one or a set 4 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 r 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, r2correlates with r, 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 2 , every packet within the intersection of these rules is denied by r. 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, r1 is redundant with respect to rin Table 1, since all DP packets coming from any port of 10.1.2.* to the port 53 of 172.32.1.* matched with r12can match ras 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 ther 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.

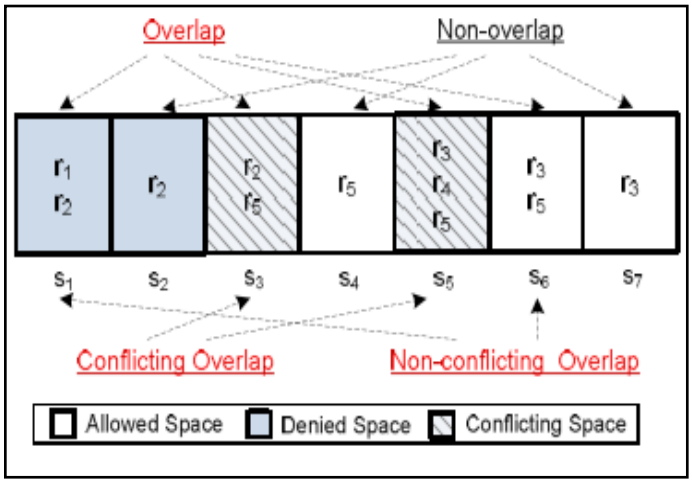

Fig. 3: Uniform Representation

1. 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].

```
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;
```

Algorithm-1 Segmentation algorithm

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.

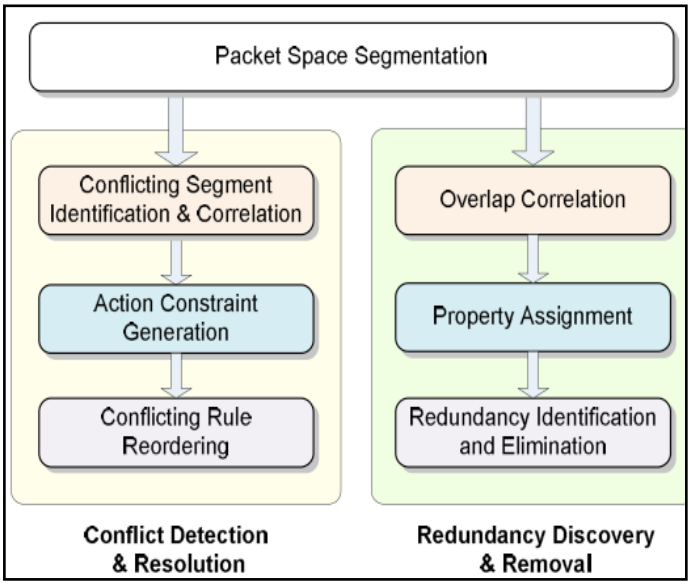

Fig. 4:

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. Fig. 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.

```
DetectionSimpleAnomaly (Px, order)
. Px translated to Cx, Ax
. for each Pi in SP
.   Pi translated to Ci, Ai
.   if orderi > order then
.     C1=Ci, C2=Cx, A1=Ai, A2=Ax
.   else
.     C1=Cx, C2=Ci, A1=Ax, A2=Ai
    if not BDD_Satisfy (A1 ? A2) then
      if BDD_Satisfy (C2 ? C1) then
          print "Shadow Anomaly (Px, Pi)", Return
      if BDD_Satisfy (C1 ? C2) then
          print "Generalization Anomaly (Px, Pi)", Return
      if BDD_Satisfy (C1 ? C2) then
          print "Correlation Anomaly (Px, Pi)", Return
    if BDD_Satisfy (C2 ? C1) then
      print "Redundancy Anomaly (Px, Pi)", Return
  print "There is no Simple Anomaly with existent propositions."
  end DetectionSimpleAnomaly
```

Algorithm2, Simple anomaly detection algorithm

```
ResolveSimpleAnomaly (SP)
  for each Pj in SP
      Pj translated to Cj, Aj
      for each Pi in SP except Pj
          Pi translated to Ci, Ai
          if orderi > orderj then
            C1=Ci, C2=Cj, A1=Ai, A2=Aj
          else
            C1=Cj, C2=Ci, A1=Aj, A2=Ai
          if not BDD_Satisfy (A1 ? A2) then
            if BDD_Satisfy (C2 ? C1) then
              Remove (R2), Return
          if BDD_Satisfy (A1 ? A2) then
            if BDD_Satisfy (C2 ? C1) then
              Remove (P2), Return
  end ResolveSimpleAnomaly
```

Algoritm 3, Anomaly resolving algorithm

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, and F. Cuppens, "Complete Analysis of Configuration Rules to Guarantee Reliable Network Security Policies", Int'l J. Information Security, Vol. 7, No. 2, pp. 103122, 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 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, 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.

Mr.G.Ganesh Sriram, well known Author and excellent teacher, received M.Tech in Software Engineering from NOVA college of Engineering and Technology, affiliated to JNT University, Kakinada. He has been working as an Assistant Professor in the Department of Computer Science & Engineering, Srinivasa Institute of Engineering & Technology, Cheyyeru. He has got couple of Publications in international journals to his credit. He is a committed teacher, whose areas of interest include C programming, Data Structures, Data Communications & Networks, Software Engineering and other advances in Computer Applications. He has been guiding many projects for Engineering Students.

Ms.B.Indu Swarna Sri is a student of Srinivasa Institute of Engineering& Technology, Cheyyeru. Presently she is pursuing her B.Tech (C.S.E) from this college.Her area of interest includes Computer Networks,Software Engineering, and other advanced Technologies in Computer science.

Ms.N.V.V.Rama Devi is a student of Srinivasa Institute of Engineering& Technology, Cheyyeru. Presently she is pursuing her B.Tech (C.S.E) from this college.Her area of interest includes Computer Networks,Software Engineering, and other advanced Technologies in Computer science.

Ms.K.Gouthami Durga is a student of Srinivasa Institute of Engineering& Technology, Cheyyeru. Presently she is pursuing her B.Tech (C.S.E) from this college.Her area of interest includes Computer Networks,Software Engineering, and other advanced Technologies in Computer science.

Mr.S.Arjun Kumar is a student of Srinivasa Institute of Engineering &Technology, Cheyyeru. Presently she is pursuing her B.Tech (C.S.E) from this college.Her area of interest includes Computer Networks,Software Engineering, and other advanced Technologies in Computer science.