ShatanandPatil, B. B. Meshram / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, July-August 2012, pp.608-612

Use of Formal models for the Firewall Policy Optimization

ShatanandPatil* and B. B. Meshram**
*(Department of Computer Technology, Veermata Jijabai Technical Institute, Mumbai – 19)
*(Department of Computer Technology, Veermata Jijabai Technical Institute, Mumbai – 19)

Abstract –

Firewalls are the core elements in network security & access control. A firewall controls the flow of traffic between different areas of your network. It uses a rule set called as firewall policy for this purpose. However as the size of rule set increases, specification and verification of the firewall rules becomes complicated and error-prone. This paper serves to provide an overview of the research efforts taken in the formalization of firewall policy specification and different formal models for firewall simulation, the verificationof firewall policies, which will help in detecting the potential problems in the firewalls and also anomaly free editing of the firewall policies. At the end an algorithm has been proposed to reduce the number of conflicting filters by introducing time field in the Policy tree representation.

***Index Terms*— Firewall; Rule Set; Firewall Decision Diagrams; Anomalies; Redundancy;**

I. INTRODUCTION

Security is the main concern when it comes to the computer networks. There are various technologies to secure computer networks including Intrusion detection systems (IDS), various antiviruses, Firewalls etc. This paper focuses on one of the most important technology of network security i.e. Firewalls.

There are various types of firewall technologies as specified by Robert Zalenski [10]. But this paper focuses only on the most basic type of firewall i.e. packet filtering firewall. This paper discusses basics of packet filtering, followed by various policy representation techniques. Then it addresses the research efforts in the area of policy optimization and detection of configuration errors in firewalls.

The rest of the paper is organized as follows. Section II briefly explains the basic definitions regarding firewall rule setand firewall languages and alsogives an overview of different models for firewall policy representation. Section IIIis focused on the comparison of these models by considering firewall performance. Section IV addsconcluding remarks to the paper.

II. BASICS OF PACKET FILTERING

A. Firewall Rule Set

A packet filtering firewall performs its function based on a specifiedsequence of rules. Each rule is of the form

<predicate>→<decision>

Where < predicate > is a condition consisting of various variables that assigns to each packet a Boolean value, true or false, and < decision > specifies the action to be taken which is either ”accept” or ”deny”.

Accept means the packet is permitted to pass through

Deny means the packet is dropped silently.

Also activities like logging can be attached with the above actions. For a packet to match a condition, all tests must be satisfied. A firewall F uses first match criterion to decide which rule should be applied to which packet.

While specifying filtering rule format, Ehab and Hazem [6] mention that the most commonly used filtering fields are: protocol type, source IP address, source port, destination IP address, and destination port i.e. a 5-tuple rule. But in Time-Based Firewall Policies [11], additionally time field which specifies Active period is also specified.

A firewall security policy is a list of ordered rules that define the actions performed on network packets based on the specific filtering conditions.Because the last rule in a firewall is either an accept-all rule or a discard-all rule, it is straightforward to show that for every packet and every firewall F, either the packet is accepted by F or the packet is discarded by F. Two firewalls F and G are said to be equivalent iff F andG accept the same set of packets (and discard the same set of packets).

Rule	Protocol	Source		Destination		Action
		Address	Port	Address	Port	
1	tcp	130.192.37.20	any	*.*.*.*	80	deny
2	tcp	130.192.37.*	any	*.*.*.*	80	accept
3	tcp	*.*.*.*	any	141.120.33.40	80	accept
4	tcp	130.192.37.*	any	141.120.33.40	80	deny
5	tcp	130.192.37.30	any	*.*.*.*	21	deny
6	tcp	130.192.37.*	any	*.*.*.*	21	accept
7	tcp	130.192.37.*	any	141.120.33.40	21	accept
8	tcp	*.*.*.*	any	*.*.*.*	any	deny
9	udp	130.192.37.*	any	141.120.33.40	53	accept
10	udp	*.*.*.*	any	141.120.33.40	53	accept
11	udp	130.192.38.*	any	141.120.35.*	any	accept
12	udp	*.*.*.*	any	*.*.*.*	any	deny

Figure 1: Example of firewall rule set

B. *Firewall Languages*

There are various firewall languages proposed and used such as high level firewall language [16], Firewall builder. High Level Firewall Language translates high level language firewalling rules into usable rules for IPChains, NetFilter, IPFilter, Cisco, and many other firewalls. These languages allow us to describe the policy of a firewall.

Apart from this, Liu and Gouda[1] introduce a simple and effective SQL-like query language, called the Structured Firewall Query Language (SFQL), for describing firewall queries. This language uses queries of the form "select . . . from . . . where . . . .", just like SQL queries.

C. *Related work onfirewall policy models*

A significant amount of work has been reported in the area of firewall. In this section, we focus our study on the related work in the area of packet filter modeling and firewall verification and optimization. The research in this area is fragmented. A single, generally accepted mathematical model describing firewall policies is yet to emerge. Below us highlightsome of the work in this area:

D. *Firewall Decision Diagram*

A Firewall Decision Diagram (FDD) with a decision set DS and over fields F1, · ·, Fd is an acyclic and directed graph that has the following five properties [3],[14]:

1) There is exactly one node that has no incoming edges. This node is called the root. The nodes that have no outgoing edges are called terminal nodes.
2) Each node v has a label, denoted F(v), such that F(v) ∈ {F1, · · · , Fd} if v is a non-terminal node, DS if v is a terminal node.

3) Each edge e:u → v is labeled with a nonempty set of integers, denoted I(e), where I(e) is a subset of the domain of u's label (i.e., I(e) ⊆ (F(u))).

4) A directed path from the root to a terminal node is called a decision path. No two nodes on a decision path have the same label.

5) The set of all outgoing edges of a node v, denoted E(v), satisfies the following two conditions:

a) Consistency: I(e) ∩ I(e) = ∅ for any two distinct edges e and e in E(v).

b) Completeness: e∈E(v) I(e) = D(F(v)).

All FDD based models first convert the firewall rule set into the Firewall Decision Diagram to satisfy the properties mentioned above. The construction of FDD is explained by Alex X. Liu, andFei Chen. [15].

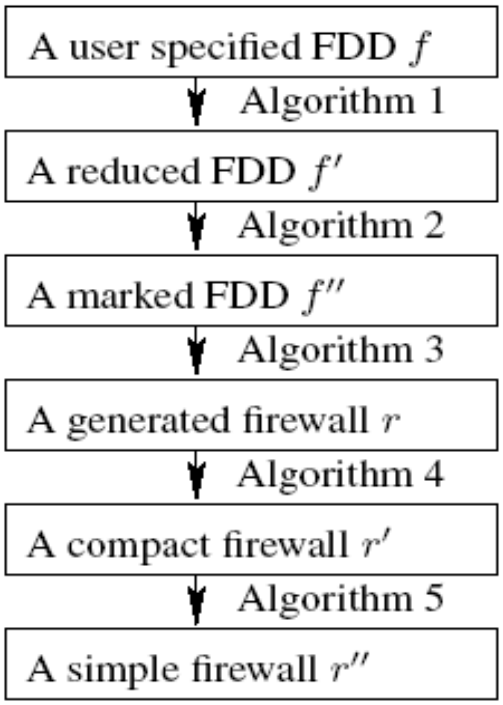

Figure 2: Steps of FDD based Firewall Design [15]

The method proposed by Gouda and Liu [14] starts by some user specifying an FDD *f* and verifies consistency and completeness properties of *f*systematically. *f*is first reduced (using Algorithm 1), and some of its edges are marked with the ALL mark (using Algorithm 2), then the firewall is generated from the marked FDD (using Algorithm 3). Algorithm 3, maintains the consistency and completeness conditions of the original FDD. Algorithm 4 is used to detect and remove all the remaining redundant rules from the generated firewall. Finally, Algorithm 5 is used to simplify the rules in the generated firewall. The marking algorithm, Algorithm 2, guarantees that the number of simple rules in the generated firewall is kept to a minimum.

E. *Policy Tree*

Ehab and Hazem[2],[6] use a firewall rule structure having 5 fields : protocol, sourceip,source port,destinationip,destination port.

In order to build a useful model for filtering rules, these researchers determined all the relations that may relate packet filters and also showed that no other relation exists.

Then they have defined the following relations between two rules: "completely disjoined", "exactly

matched", "inclusively matched", "partially disjoint", "correlated".

Next Al-Shaer and Hamed prove that these relationships are distinct and proved that no other relation exists between any two*k*-tuple filters in a firewall policy.

The policy is represented as a single-rooted tree called as policy tree where each node represents field of a filtering rule and each branch at this node represents a possible value of the associated field. Every tree path starting at the root and ending at a leaf represents a rule in the policy and vice versa. An example of such a tree taken from [2] is shown at Figure 1

The basic idea for building the policy tree is to insert the filtering rule in the correct tree path. When a rule field is inserted at any tree node, the rule branch is determined based on matching the field value with the existing branches. If a branch exactly matches the field value, the rule is inserted in this branch; otherwise a new branch is created. The rule also propagates in subset or superset branches to preserve the relations between the policy rules. [6]

In [6] the authors first identified following firewall policy anomalies: shadowing anomaly, correlation anomaly, generalization anomaly, redundancy anomaly and Irrelevance anomaly along with the algorithm to detect any of these anomalies. The basic idea for discovering anomalies is to determine if any two rules coincide in their policy tree paths. If the path of a rule coincides with the path of another rule, there is a potential anomaly that can be determined based on the firewall anomaly definitions. If rule paths do not coincide, then these rules are disjoint and they have no anomalies. The authors in [6] also provide algorithms for anomaly free insertion of firewall rules and rule removal from a firewall rule set.

In [2] they extend their anomaly-detection algorithms to distributed firewall configuration, consisting of multiple firewalls. They provide format definition of various Inter-Firewall Anomalies and propose algorithms for their detection. The Firewall Policy Advisor [2] presented in this paper provides a number of techniques for purifying and protecting the firewall policy from rule anomalies.

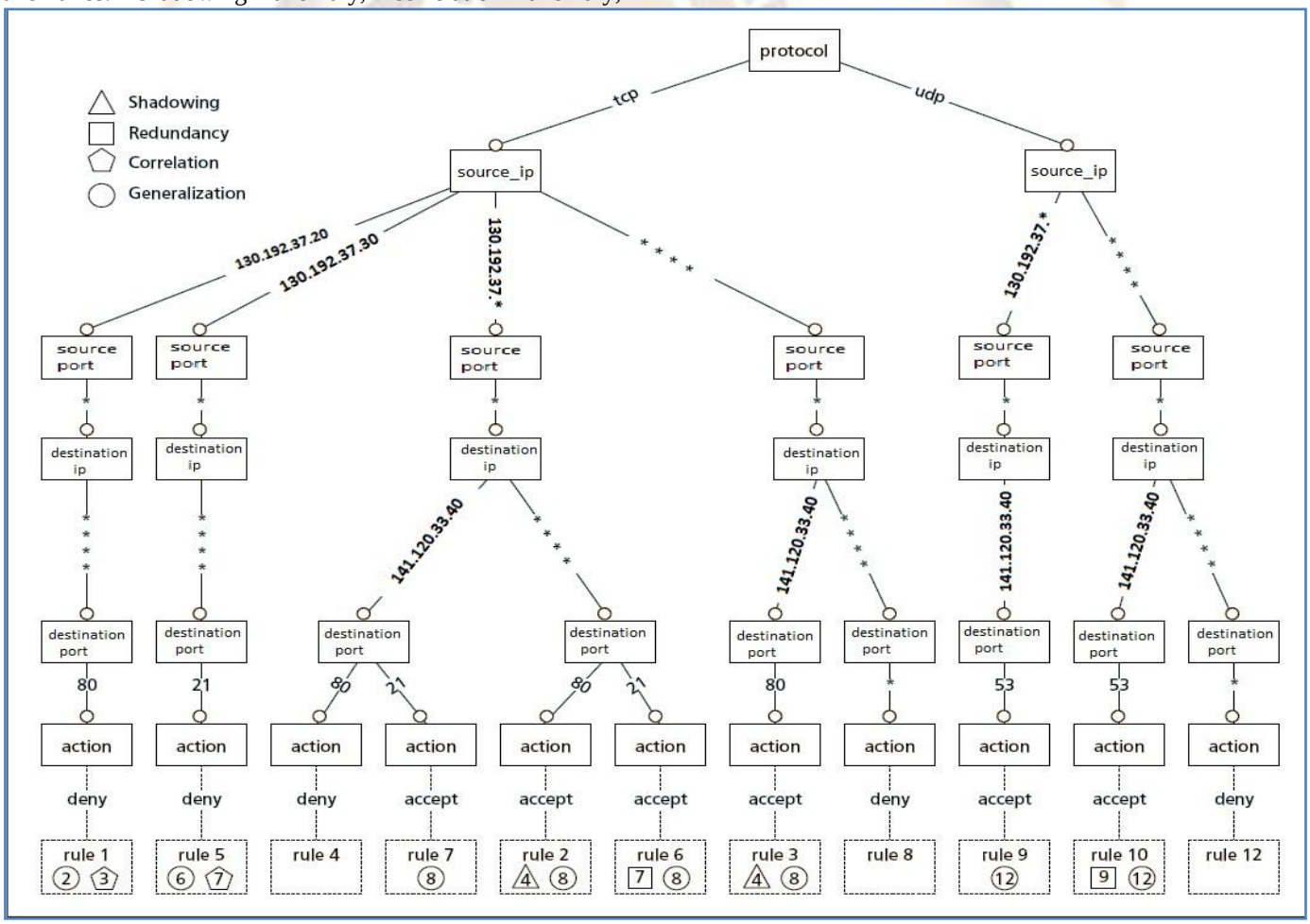

Figure 3: Policy Tree Representation

F. Time Based Firewall Policy

What we have seen in the earlier models mentioned above is that they take into account only the spatial domain. Subana, Yuichiro, Yoshiaki and Naohisa [11] have conducted experiments on time-based filters with and without considering time and found that around 50% of the conflicting filters become non-conflicting in time domain. Thus the workload of administrator is greatly reduced as the number of conflicting filters decreases rapidly in Time based Firewall Policy.

A time-based filter *fi* matches, if the packet arrival time falls on its active period. The active period is specified by two subfields called TIME and DAY.[11] So we can consider this firewall policy by adding an extra time field to the firewall rule we have seen so far.

The relations between rules of a firewall policy discussed still remain valid in Time based Firewall Policy also. Additionally due to time field, administrator will have extra information about which rules will be active simultaneously, and which rules operate in mutual exclusion. The administrator needs to check only those rule sets for which there is a collision in the time domain.

III. DISCUSSIONS UTILIZATION AND EVALUATION

In section II we have studied the different formal models for firewall specification and their use in firewall redundancy and conflict detection. In this section we will discuss the advantages and disadvantages of using the above models.

The main advantages of firewall decision diagrams are that their consistency and completeness can be checkedsystematically. Also the set of a sequence of five algorithms given in [15] can be applied to a firewall decision diagram to generate a compact sequence of firewall rules whilemaintaining the consistency and completeness of the original firewall diagram.

The tree model given by Ehab and Hazem [2],[6] has an interesting property, on which they base their algorithms:

If filter fields are prefix fields, then each field of a filter is either a strict subset of, or equal to, or a strict superset of, or completely disjoint from the corresponding field in any other filter. In other words, it is not possible to have partial overlaps of fields. Partial overlaps can only occur when the fields are arbitrary ranges, not prefixes.

Using this property, they propose to solve a filter conflict problem using a directed graph/ State diagram. The main advantage of using such a state diagram is that the time required to detect an anomaly is logarithmically reduced. Also through minor changes, the same model can be extended to distributed firewall environment. In regards to usability, the policy tree model is able to discover filtering anomalies in rules written by expert network administrators. In regards to performance, although the policy analysis algorithm is parabolicallydependent on the number of rules in the firewall policy,still the average processing time for anomaly discovery is very reasonable for practical firewall policies.

The main disadvantage of the FDDs and policy tree model is that they take into account only the spatial domain and not the time domain. In real scenario, most of the time there will be different sets of firewall policies that will be active at different times. With time based model, the research indicates that *disjoint*filters increases from 1% to 48% when time is considered. If we consider the other topologies, the error causing *inside* topology is reduced from 16% to 2%. The warning causing *overlap* topology is reduced from 78% to 49% and warning causing *contains* topology is reduced from 5% to 1%.The results of time-based filters show that, when time is considered for conflict detection, many filters lie apart with each other. When the computational dimension increases from *n* to *n*+1, conflicting filters in *n*dimension becomes *disjoint* in time and as a result, the numberof conflicting filters decreases rapidly. Also with time based filtering policy, not much overhead is introduced though we have to add an extra tuple in the firewall rule.

While the policy representation problem is well studied, most models consider a simple ordered set of rules, usually with "single trigger" semantics i.e. an action of the first matching rule will be performed. For practical applications, models must be able to deal with more complex processingmodels implemented in real firewall products. This includes the "multitrigger"processing model (i.e. all rules will be matched and an action from the last matching rule will be performed), as well as more complex, branching models like thechains-based one used in NetFilter. Perhaps some of these models could betransformed to a more simple ordered set of rules model, but such transformationshave yet to be formally defined.Most policy optimization and anomaly detection algorithms described inthe literature only consider "accept" and "reject" actions. However some rulesmight have actions producing side effects and such rules (along with their triggeringorder) must be preserved during policy transformation/optimization process.

Based on the above models, we propose a modified model for conflict detection in firewalls. The goal of the proposed system is to reduce the number of conflicting filters by introducing time field in the Policy tree representation given by Liu and Gauda.[1] The main problem in this system will be how to divide the time in order to detect whether the two rules overlap in time domain. For this purpose we will use a state diagram approach which will be same as the approach used to detect anomalies in spatial domain.

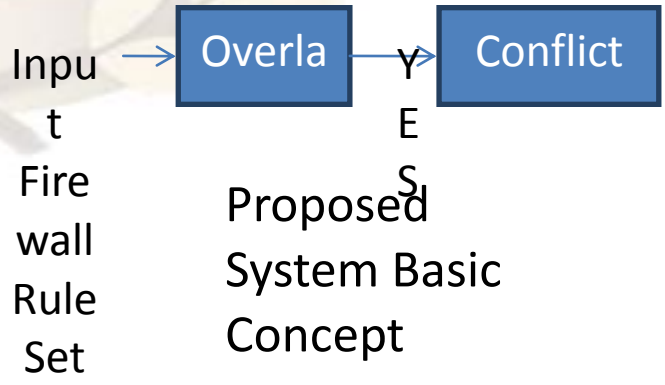

Figure 4 : Proposed System basic Concept

The block diagram for the proposed system is as given above. It accepts firewall rule set as input. It then checks if there is any overlap in time domain. If not then there is no need of anomaly discovery. If yes then it applies conflict detection using policy tree. It basically stresses upon applying conflict detection algorithm only if it is overlapping in time domain.

Algorithm for proposed system

Input: ***rule, branch***

Output: ***anomaly***

1: Flag=No
2: For each field ∈rule.fields
3: If field = period field then
4: If branch ∩period.day = ϕ then
5: flag=No
6: Else if branch ∩period.time = ϕ Then
7: flag = No
8: Else flag = Yes
9: End if
10: If flag = Yes Then
11: apply anomaly detection algorithm as proposed by Liu and Gauda [2]
12: Else anomaly ← NOANOMALY
13: End if
14: End if
15: End for

The algorithm for the proposed system is based on the algorithm proposed by Liu and Gauda for firewall anomaly detection [2]. In fact it improvises the given algorithm by considering the time domain. It starts first by considering that there is no anomaly in the system. It thus sets flag to No. it checks whether the two rules overlap in time domain or not. Considering example firewall filtering policy as given in above fig., algorithm first checks if the two rules are active on the same day. If not then there is no question of anomaly detection. If the two rules are active on the same day, then it checks if there is an overlap of time interval. If not then there is no question of anomaly detection. If there is an overlap of time interval, then this means that the two rules will be active at the same time and hence it is needed to check if there is any conflict in space domain. For anomaly detection in space domain, we apply the Firewall anomaly discovery algorithm as proposed by Liu and Gauda [2].

IV. CONCLUSION

Firewalls provide proper security services only if they are correctly configured. Firewall policies used in real time scenarios are getting more complex as the number of firewall rules and devices becomes larger. As a result, there is a high demand for an effective policy management tool which significantly helps user in discovering firewall policy's properties and finding rule anomalies in both single and distributed firewalls.

In this paper, we described different formal models for firewall simulation, the verification of firewall policies, which will help in detecting the potential problems in the firewalls and also anomaly free editing of the firewall policies. We also observed that when time is considered for conflict detection, many filters lie apart with each other and hence performance of the firewall is increased.

REFERENCES

[1] Alex X. Liu, Member, and Mohamed G. Gouda, "Firewall Policy Queries", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 20, NO. 6, JUNE 2009
[2] Ehab Al-Shaer, HazemHamed, RaoufBoutaba, and MasumHasan, "Conflict Classification and Analysis of Distributed Firewall Policies",IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 23, NO. 10, OCTOBER 2005
[3] Alex X. Liu, Member and Mohamed G. Gouda, "Diverse Firewall Design", IEEE RANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 19, NO. 9, SEPTEMBER 2008
[4] GhassanMisherghi, Lihua Yuan, Zhendong Su, Chen-Nee Chuah, and Hao Chen, "A General Framework for Benchmarking Firewall Optimization Techniques", IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 5, NO. 4, DECEMBER 2008
[5] Alex X. Liu and Mohamed G. Gouda, "Complete Redundancy Removal for Packet Classifiers in TCAMs", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 21, NO. 4, APRIL 2010.
[6] Ehab S. Al-Shaer and Hazem H. Hamed, "Modeling and Management of Firewall Policies", eTransactions on Network and Service Management, Second Quarter 2004 IEEE.
[7] H. B. Acharya, Aditya Joshi and M. G. Gouda, "Firewall Modules and Modular Firewalls", 978-1-4244-8645-8/10©2010 IEEE
[8] Alex X. Liu Eric Torng Chad R. Meiners, "Firewall Compressor: An Algorithm for Minimizing Firewall Policies", 978-1-4244-2026-1/08 © 2008 IEEE
[9] Zubair A. Shaikh and FurqanAhmed,"Disarming Firewall", 978-1-4244-8003-6/10/$26.00 ©2010 **IEEE**
[10] Robert Zalenski,"Firewall Technologies", 0278-6648/02/$17.00 © 2002 IEEE, IEEE POTENTIALS
[11] SubanaThanasegaran, YuichiroTateiwa, Yoshiaki Katayama, Naohisa Takahashi, "Simultaneous Analysis of Time and Space for Conflict Detection in Time-Based Firewall Policies", 978-0-7695-4108-2/10 $26.00 © 2010 IEEE
[12] *AvishaiWool*,"A Quantitative Study of Firewall Configuration Errors", **0018-9162/04/$20.00 © 2004 IEEE,** IEEE Computer Society
[13] Bilal Khan, Muhammad Khurram Khan, Maqsood Mahmud, Khaled S. Alghathbar "Security Analysis of Firewall Rule Sets in Computer Networks", 978-0-7695-4095-5/10 $26.00 © 2010 IEEE
[14] Mohamed G. Goudo, Xiang-Yang Alex Liu, "Firewall Design : Consistency, Completeness and Compactness", Proceedings of the 24th International Conference on Distributed Computing Systems(ICDCS'04),1063-6927/04 2004 IEEE
[15] Alex X. Liu, Member, Fei Chen, "Privacy Preserving Collaborative Enforcement of Firewall Policies in Virtual Private Networks", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 22, NO. 5, MAY 2011
[16] http://www.hlfl.org/
[17] Alex X. Liu, "Formal Verification of Firewall Policies", IEEE Communications Society 2008.