Firewall Rule generation Using decision tree diagram

Dr.P.Senthilkumar
Professor, Department of CSE, Shadan Women's College of Engineering and Technology
JNTUH,Hyderabad,India
psenthilmephd@gmail.com
Asha.V
Faculty, The Indian Public School,Erode,Tamilnadu
ashamecse@gmail.com
M. Muthukumar
Assistant Professor in CSE,SRM Institute of Science and Technology
Delhi-NCR campus, Modi Nagar,UP 201204

***Abstract*— A firewall is a safety measure that is put in place between two or more networks. The ordering of the filtering rules affects the firewall's functionality. The correct rule order must be determined after taking into account all rule relations. Every private network on the Internet has a firewall installed at its entry. A firewall's job is to inspect each packet that enters the system and determine whether to accept it and let it continue, or to reject it and send it on its way. There are three main issues with the existing method of explicitly constructing a firewall as a liner rule.1. When a user specifies a rule in a firewall, the firewall machine checks to see if it matches or not.2. Enter the website if the rules are met in which case the packet is accepted; otherwise, the packet is dropped.3. Use mathematics to identify the unnecessary rules. Creating a firewall decision tree diagram (FDTD) is the first step in our procedure, and a theorem can be used to verify its consistency and completeness. The current study takes into account a scenario in which packet traffic results in a dynamic access rule set, which increases the computational cost of binary conversion during comparison. Therefore, integrating traffic awareness to create dynamic access rules and converting the access rule list to binary format will improve firewall optimization. Results from 1 million packets show that using a BDD-based strategy over a list-based with promotion method results in an average reduction of 70% for most-accept packets in such comparisons. This reduction is about 32% for packages that receive the most rejections.**

Keywords*— *FDTD, Firewall Rule generation, Firewall Compactness

I. INTRODUCTION

One of the most important modern firewall design methods is packet filtering. To make the decision solely at the packet is a key design objective. Utilizing a Binary Decision Diagram (BDD) for the implementation of such a packet filter has significant benefits in terms of memory utilization and lookup time. The time it takes to decide on a packet is related to the number of rules in the list-based packet filter firewall, where rules are examined one at a time for each incoming packet. Rule promotion improves performance, but it is a slow process in and of itself. In this paper, we describe a BDD-based method that produces significantly superior results in terms of the amount of comparisons or rule list accesses.

Installing a firewall can be used to prevent internet access. At the location where the internal network links to the Internet, a firewall is established (Figure1).It blocks undesirable traffic from or to the internal network. An ordered collection of rules is the basis for the filtering decision. Filtering rules are necessary for the firewall to function properly. When deciding on the proper rule sequence, the administrator must take into account all rule relationships.

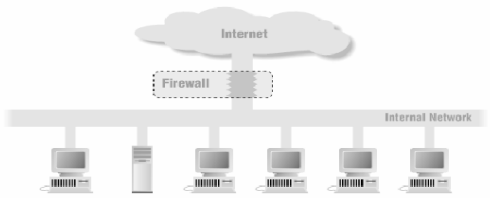

Figure1 Firewall diagram

At every point where a private network enters the Internet, a firewall is frequently installed. This firewall's job is to enable safe access to and from the private network. Particularly, the firewall situated at that point first examines any packet that attempts to enter or leave the private at some entry point, and depending on the various fields of the packet, the firewall decides whether to accept the packet and allow it to proceed in its way or to discard the packet.

A firewall is composed of a set of rules, each of which is of the form **Predicate → Decision**

Where the Decision is either "a" (for accept) or "d" and the Predicate is a Boolean expression over the various fields of a packet (for discard).The rules in the sequence are checked one by one until the first rule, whose Predicate is satisfied by the packet fields, is identified, at which point a judgment regarding a packet is made. This rule's decision is applied to the packet. The firewall rule set should be maintained as straightforward and narrowly focused as feasible. Inbound traffic should be banned by default policy unless the connections and traffic type have been specifically permitted.

The following is a typical format for a rule in a firewall policy: Order Sequence, Protocol, Source IP, Source port, Destination IP, Destination port, Decision or Action, Source IP, Destination IP, Destination Port.

The rule's place in the rule set is determined by the Order Sequence field, while the packet's transport protocol is specified by the Protocol field. The Source IP and Destination IP fields, respectively, specify the IP addresses of the source and destination. Source Port and Decision or Action fields define the port addresses of the packet's source and destination.

According to a set of rules, packets are either allowed through or blocked by a firewall. A sample packet filter firewall rule set that prevents all TCP traffic entering the network is shown in Table 1

Order Sequence	Protocols	Source IP	Source Port	Destination IP	Destination Port	Decision / Action
1	TCP	192.168.2.1	500	192.168.2.49	600	Discard
2	TCP	192.168.2.1	200	192.168.2.150	300	Accept
3	TCP	192.168.2.1	200	192.168.2.150	300	Redundant

Table 1 Packet Filter Firewall Rule Set

II. FIREWALL DECISION BINARY TREE DIAGRAMS (FDBTD)

The firewall of Fi, sometimes known as the field of Fi, is the range of nonnegative integers from which the value of a field Fi's variable is selected (Fi).An n-tuple (p0, p, pn) is a packet spanning the fields F0, , and Fn1, where each pi is drawn from the domain D(Fi) of the corresponding field Fi.A firewall decision binary tree diagram (FDBTD) is an acyclic, directed graph, complete tree, and two additional nodes in a tree that meets the following five criteria:

1. The root of f is exactly one node with no incoming edges, and the terminal nodes of f are two or more nodes with no outgoing edges.
2. A field, indicated by F(v), selected from the collection of fields F0, , and Fn1, is used to designate each nonterminal node v in the graph f.Accept (abbreviated "a" for short) or discard (abbreviated "d" for short) is the decision that is assigned to each terminal node v in the function.
3. A decision path in f is a directed path that leads from the root to a terminal node. A decision path in f has no nodes with identical labels.
4. An integer set I(e), which is a subset of the domain of field F, is assigned to each edge e that leaves a node v in the graph f. (v).
5. Let v be any of the f's terminal nodes.The set E(v) of every node v's outgoing edges complies with the following two requirements:

a.For any distinct ei and ej in E, consistency (v), I(ei) ∩ I(ej) = ∅

b.UeE(v) I(e) = D(F(v)) for completeness. Where D(F(v)) is the domain of the field F and is the empty set (v).

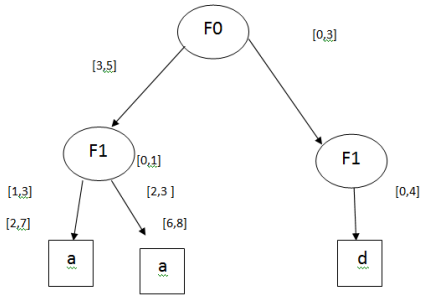

Figure 2. Firewall Decision Binary Tree Diagrams

The figure 2 displays a binary tree model FDBTD spanning the two fields F0 and F1.Each field's domain falls within the range [0, 8].A set of integers that are represented by one or more non-overlapping intervals that cover the set of numbers are assigned to each edge in this FDBTD as its label.A series of rules, each of which is of the form, can be used to express an FDBTD f over the fields F0, , and Fn−1

$$F0 \in S0 \wedge \cdot\cdot\cdot \wedge Fn-1 \in Sn-1 \rightarrow \text{Decision}$$

If a firewall of f has a rule, a packet (p0, , pn1) over the fields F0, , and Fn−1 is said to be approved by an FDBTD f over those same fields.

$$F0 \in S0 \wedge \cdot\cdot\cdot \wedge Fn-1 \in Sn-1 \rightarrow \text{Accept}$$

so that the condition is maintained. $p0 \in S0 \wedge \cdot\cdot\cdot \wedge pn-1 \in Sn-1$

Similarly, if a firewall of f has a rule, it is asserted that a packet (p0, , pn−1) over the fields F0, , Fn−1 will be discarded by an FDBTD f over those same fields.

$$F0 \in S0 \wedge \cdot\cdot\cdot \wedge Fn-1 \in Sn-1 \rightarrow \text{Discard}$$

III. THEOREM OF FIREWALL DECISION BINARY TREE DIAGRAMS

Any FDBTD f over fields F0,,, and Fn−1.Let f be an FDBTD over the fields F0, Fn-1, and let represent the set of all packets over those fields.The subset of that contains all the packets that f has accepted is known as the f.accept set.In a similar vein, the subset of known as f.discard comprises all of the packets that f has discarded.

Theorem A: f.accept ∩ f.discard = ⦰
Theorem B: f.accept ∪ f.discard = Σ

Where the ∅ (empty set) is 0 and the (Σ)set of all packets spanning the fields F0, , and Fn1 is 1.

The figure 2 FDBTD as an example. The values for the accept and discard f.accept={1,3}{2,7}
f.discard={0,3}

Applying the Theorem A:f.accept ∩ f.discard = ⦰
Replace the values for f.accept and f.discard. {1, 3}{2, 7}∩ {0, 3} =3 Thus, Theorem A is demonstrated.

Applying the Theorem B: f.accept ∪ f.discard = Σ
Replace the values for f.accept and f.discard.{1,3}{2,7}∪{0,3}={0,1,2,3,7} Theorem B is demonstrated.

IV. RELATED DISCUSSION

If an FDBTD f meets all three of the following criteria, it is said to be reduced:

1. No node in the graph f has exactly one outgoing edge.

2. F does not have any edges that are both entering and leaving from the same node.

3. There aren't any two separate isomorphic nodes in f.

Using a smaller FDBTD the reduced FDBTD in Figure 3 is produced by three conditions to the FDBTD in Figure 2.

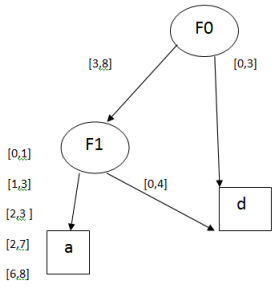

Figure 3 Reduced FDBTD

V. FIREWALL GENERATION

The firewall that is generated consists of a series of rules, each of which corresponds to a decision path in the indicated FDBTD f.

The rules r0, , rm−1 that make up a firewall over the fields F0, , and Fn−1 are each of the following forms: F0 ∈ S0 ∧ ・ ・ ・∧ Fn−1 ∈ Sn−1 → Decision

Each Si is either a nonempty set of numbers drawn from the domain of field Fi or the mark ALL (which is an interval of consecutive nonnegative integers).There are two options: a (for accept) or d. (for discard).The r1 and r2 in this instance stand for accept and discard.

When the Create firewall conditions are applied to Figure 3 (Reduced FDBTD), the resulting generate firewall is shown in table 2.

r 1 = F0 ∈ [3, 5] ∧ F1 ∈ [0, 1] ∪ [1, 3] ∪ [2, 7] ∪ [2, 3] ∪ [6, 8] → a
r 2 = F0 ∈ [3, 8] ∧ F1 ∈ [0, 1] ∪ [1, 3] → d

Table 2 Packet Filter Firewall Rule Set

FIREWALL COMPACTNESS

If a firewall has no redundant rules, it is referred to as compact. The firewall in Figure 3 can be easily argued to be small.

r 1 = F0 ∈ [3, 5] ∧ F1 ∈ [0, 8] → a

r 2 = F0 ∈ [3, 8] ∧ F1 ∈ [0, 3] → d

Theorem for Redundancy of Firewall Rules

Let (r0, r, rm-1) act as a firewall over fields F0, r, rm-1, and Fn-1.In this firewall, a rule ri is redundant if at least one of the following two requirements is true for each j, I j m-1:

1. rj's decision is the same as ri's decision.

2.No packet over fields F0, , and Fn1 satisfies the criteria.

ri.op ∧ (¬ri+1.ep ∧ ・ ・ ・ ∧ ¬rj−1.ep) ∧ rj.ep

where the original predicate of ri and rj is indicated by ri.op.ep stands for rj's demonstrated predicate.

VI. PROPOSED METHODOLOGY

We require an efficient firewall access rule set that enables early packet rejection and acceptance while minimizing computing overhead. The goal is to develop an efficient set of dynamic access rules based on packet traffic patterns, and then to convert this optimized rule set to binary in order to improve early packet rejection and lower computation overhead. The method of optimization applied to the initial firewall rule set. Figure 3 shows the three filtering settings used for this optimization.

When using our primary implementation, a user must first define an FDBTD f.It is possible to systematically verify the consistency and completeness properties of f, perhaps with the use of a computer programme.F should not be used to directly generate firewall even though it ensures that the finished firewall is consistent and complete. Regrettably, some redundant rules may still be included in the firewall that was generated (even though this firewall is generated after applying the reduction technique).

VII. CONCLUSION

Our two approaches in this study are our contribution. First, decide how to use a firewall. Early in the design process for firewalls, binary tree diagrams are used to specify firewalls. These diagrams' fundamental benefit is that their consistency and completeness can be methodically examined. The second method involves applying a number of different theorems to a firewall decision binary tree diagram in order to produce a concise list of firewall rules while preserving the consistency, completeness, compactness, and identification of redundant rules in the original firewall design. However, this design strategy may be simply expanded to let a firewall choose from a wide range of options.

REFERENCES

1. High level firewall language, http://www.hlfl.org/.
2. F. Baboescu and G. Varghese. Fast and scalable conflict detection for packet classifiers. In Proc. of the 10th IEEE International Conference on Network Protocols, 2002.
3. Y. Bartal, A. J. Mayer, K. Nissim, and A. Wool. Firmato: A novel firewall management toolkit. In Proc. of IEEE Symp.on Security and Privacy, pages 17–31, 1999.
4. A. Begel, S. McCanne, and S. L. Graham. BPF+: Exploiting global data-flow optimization in a generalized packet filter architecture. In Proc. of ACM SIGCOMM ’99, 1999.
5. R. E. Bryant. Graph-based algorithms for boolean function manipulation. IEEE Trans. on Computers, 35(8):677–691, 1986.
6. M. M. Buddhikot, S. Suri, and M.Waldvogel. Space decomposition techniques for fast Layer-4 switching. In Proc. Of PHSN, Aug. 1999.
7. D. Eppstein and S. Muthukrishnan. Internet packet filter management and rectangle geometry. In Symp. on Discrete Algorithms, pages 827–835, 2001.
8. A. Feldmann and S. Muthukrishnan. Tradeoffs for packet classification. In Proc. of 19th IEEE INFOCOM, Mar. 2000.
9. P. Gupta and N. McKeown. Algorithms for packet classification. IEEE Network, 15(2):24–32, 2001.
10. J. D. Guttman. Filtering postures: Local enforcement for global policies. In Proc. of IEEE Symp. on Security and Privacy,pages 120–129, 1997.
11. A. Hari, S. Suri, and G. M. Parulkar. Detecting and resolving packet filter conflicts. In Proc. of IEEE Infocom, pages 1203–1212, 2000.
12. V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel. Fast and scalable layer four switching. In Proc. of ACM SIGCOMM,pages 191–202, 1998.
13. K. Strehl and L. Thiele. Interval diagrams for efficient symbolic verification of process networks. IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems,19(8):939–956, 2000.
14. Mohamed G. Gouda and Xiang-Yang Alex Liu “Firewall Design: Consistency, Completeness, and Compactness” Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS’04)