Limitation of Listed-Rule Firewall and the Design of Tree-Rule Firewall

Thawatchai Chomsiri, Xiangjian He, and Priyadarsi Nanda

School of Computing and Communications,
Faculty of Engineering and Information Technology,
University of Technology, Sydney, Australia
Thawatchai.Chomsiri@student.uts.edu.au,
{Xiangjian.He,Priyadarsi.Nanda}@uts.edu.au

Abstract. This research will illustrate that firewalls today (Listed-Rule Firewall) have five important limitations which may lead to security problem, speed problem, and "difficult to use" problem. These limitations consist of, firstly, limitation about "Shadowed rules" (the rule that cannot match with any packet because a packet will be matched with other rules above) which can lead to security and speed problem. Secondly, limitation about swapping position between rules can bring a change in firewall policy and cause security problem. The third limitation is about "Redundant rules" which can cause speed problem. Next, limitation of rule design; firewall administrators have to put "Bigger Rules" only at the bottom or lower positions that can result in a "difficult to use" problem. Lastly, limitation from sequential computation can lead to speed problem. Moreover, we also propose design of the new firewall named "Tree-Rule Firewall" which does not have above limitations.

Keywords: firewall, rule list, rule conflict, tree rule, network security.

1 Background and Related Works

Firewalls are important devices that can improve network security. A firewall's security level does not depend on its cost, but rather comes from the secure rules inside it. While configuring the firewall, we should focus on creating accuracy and non-conflicting rule sets. There are many studies about firewall rule conflicts (anomalies) that occur within rule sets. E-hab Al Shaer et al [1] propose several anomaly definitions including “Shadowing anomaly”. He defined the “Shadowed Rule” as the rule that cannot match with any packet. For example, rule number 4 (see Table1) is a shadowed rule. This type of rule can be removed from the rule list without any change of policy. Moreover, they have applied their definitions and theories for analyzing a distributed firewall [2]. In [1-3], authors focused on mathematics for analyzing firewall rules. Scott Hazelhurst [4] uses Binary Decision Diagrams (BDDs) to present and analyze rule sets. Pasi Eronen[5] proposed an Expert System that is based on Constraint Logic Programming (CLP) for users to write higher-level operations to detect common configuration mistakes and find packet matched on each rule.

Y. Xiang et al. (Eds.): IDCS 2012, LNCS 7646, pp. 275–287, 2012.

Table 1. An example of rules on the Listed-Rule Firewall

No.	Protocol	Source IP	Destination IP	Dest. Port	Action
1	TCP	10.1.1.1	20.1.1.1	80	Accept
2	TCP	10.1.1.2	20.1.1.1	80	Deny
3	TCP	10.1.1.0/24	20.1.1.1	80	Deny
4	TCP	10.1.1.3	20.1.1.1	80	Accept
5	TCP	10.2.2.0/24	20.2.2.5	80	Deny
6	TCP	10.2.2.5	20.2.2.0/24	80	Deny
7	TCP	10.3.3.0/24	20.3.3.9	80	Accept
8	TCP	10.3.3.9	20.3.3.0/24	80	Deny
9	IP	0.0.0.0/0	0.0.0.0/0	0-65535	Deny

Lihua Yuan et al. proposed Fireman Toolkit [6] which can help administrators to design and analyze firewall rules. However, their toolkit only mitigates some problems of traditional firewall, and Fireman Toolkit is not a new type of firewall. Our research presents limitations of the traditional firewall and also propose a new type of firewall which has different mechanism inside. We use the tree-shape rule which is also hierarchical. Although the phrase "hierarchical rule set" [7] have been appeared in manuals of CSS (Cisco Services Switch), those are relevant to load-balancing devices in the network but not in firewall (describes which content (for example, .html files) is accessible by visitors to the Web site). Alex Liu and Mohamed Gouda proposed "Diverse Firewall Design" [8] using tree structure rule translated from rule list to discover and eradicate some of the rule conflictions. However their work still focuses on the traditional firewall, and they did not propose a new type of firewall.

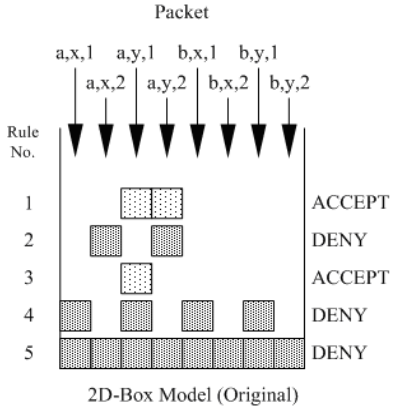
2 Limitations of Listed-Rule Firewall

In this section, we propose our solution to the five limitations of Listed-Rule Firewall (today's firewall). We begin with the model explaining the Listed-Rule Firewall. We designed a model, namely the 2D-Box Model for explaining a matching between packets and firewall rules as shown in Figure 1. Suppose there are two source IP addresses ('a' and 'b'), two destination IP addresses ('x' and 'y'), and two port numbers ('1' and '2') in the system. For understanding this model easily, we did not mentioned other attributes yet (such as source port and protocol types).

Note:

SIP: Source IP Address
DIP: Destination IP Address
DPT: Destination Port

Considering the 2D-Box Model (Figure 1- left), incoming packets will be matched with Rule-1 first. In this case, Rule-1 will ‘accept’ two packets. The remaining packets will continue to fall down to Rule-2 that has a ‘deny’ action. After that, the remaining packets will continue to fall down to other rules below until they reach the last rule or match with some rules.



Rule List

Order	SIP	DIP	DPT	ACTION
1	a	y	any	ACCEPT
2	a	any	2	DENY
3	a	y	1	ACCEPT
4	any	any	1	DENY
5	any	any	any	DENY

Fig. 1. The 2D-Box Model (left) and rules of Listed-Rule Firewall (right)

As we can see, matched packets of each rule are subsets of SIP × DIP × DPT (ACTION will be excluded), where "×" is an operator for computing the 'Cartesian product' [9]. The result from the Cartesian product is called a Relation [9]. For example (see Figure 1),

Rule Number 1 (Rule-1)
a × y × any = { (a,y,1), (a,y,2) }. The notation representing this relation is ' R_1 '

Rule Number 4 (Rule-4)
any × any × 1 = { (a,x,1), (a,y,1), (b,x,1), (b,y,1) }. The notation representing this relation is ' R_4 '

We can apply the 2D-Box Model to the firewall rules. For example we can define a range of IP address = {0.0.0.0 – 255.255.255.255}, and a range of port number = {0 – 65535} in IPv4. Moreover, the 2D-Box Model can also be extended and applied on IPv6.

Notation:

Rule-i denotes Rule number i.
R_i denotes the relation mapped from Rule-i (ACTION will be excluded).
R denotes sample relation.

Note: 'Relation' is subset of Cartesian product of domain. For example, suppose Rule-x is:

Rule No.	Source IP	Dest IP	Dest Port	ACTION
x	10.1.1.1	20.2.2.0/30	80-81	Accept

Therefore, R_x is

```
{ ( 10.1.1.1, 20.2.2.0, 80 ),
  ( 10.1.1.1, 20.2.2.0, 81 ),
  ( 10.1.1.1, 20.2.2.1, 80 ),
  ( 10.1.1.1, 20.2.2.1, 81 ),
  ( 10.1.1.1, 20.2.2.2, 80 ),
  ( 10.1.1.1, 20.2.2.2, 81 ),
  ( 10.1.1.1, 20.2.2.3, 80 ),
  ( 10.1.1.1, 20.2.2.3, 81 ) }
```

2.1 Limitations on Shadowed Rule

"Shadowed rule" is the rule that cannot match with any packet because a packet will be matched with other rules above. Examples of Shadowed rules are Rule-4 in Table1, and Rule-3 in Figure1 (right). This limitation can cause security and speed problem.

Security problems are likely to be occurred, especially in enterprise networks which have a large number of rules in the firewall. For example, suppose that a new worm is sending packets to attack the network. After this attack was detected, the firewall administrator will add new firewall rule for protection against such attack. If this added rule is not in the first order and be shadowed by old rules above which allow attacking packets, then the security problem is absolutely occurred.

Speed problem can be occurred because many shadowed rules can waste firewall processing time on useless rules. Because of most of packets will be matched with the last rule (the rule that deny all packets); as a result, these rules have to be compared with all shadowed rules that are located above the last rule. This can generate the low throughput to the firewall.

We analysed the theory behind such limitations and proved that Shadowed rules are not necessary for firewall and can be deleted without any change to firewall policy.

Theorem 1. If Rule-i has no chance to match with any packet (due to all packets being already matched with other rules above it) then we can remove Rule-i without any change of policy. In this theorem, we call Rule-i a “Shadowed Rule”.

For example, we can remove Rule-4 (in Table 1) from rule set without any change of policy.

Proof of Theorem 1. We can prove Theorem 1 using the 2D-Box Model (see Figure 1). However, we will prove it using SET and Relational operations as be shown below:

Denote "-" is a Difference operation of SET theory (this operation can be used for manipulating Relations too).

If p is a packet matched with Rule-1,

we find that $p \in R_1$ …… (can be proved by using 2D-Box Model)

If p is a packet matched with Rule-2,

we find that $p \in R_2 - R_1$ ... (can be proved by using 2D-Box Model)

If p is a packet matched with Rule-3, we find that $p \in R_3 - R_2 - R_1$

If p is a packet matched with Rule-i, we find that $p \in R_i - R_{i-1} - R_{i-2} - \ldots - R_1$

From the properties of SET, thus $p \in R_i - (R_{i-1} \bigcup R_{i-2} \bigcup \ldots \bigcup R_1)$

Before deleting Rule-i,

Packet that fell on rules above Rule-i was $p \in R_{i-1} \bigcup R_{i-2} \bigcup \ldots \bigcup R_1$

Packet that fell on Rule-i was $p \in \phi$ (because Rile-i is a Shadowed rule)

After deleting Rule-i,

Packet that fell on rules above Rule-i was

$$p \in R_{i-1} \bigcup R_{i-2} \bigcup \ldots \bigcup R_1 \quad \text{..... (same value)}$$

Packet that used to match with Rule-i ($p \in \phi$) and switched to match with the other rules below was $p \in \phi$

Therefore, deleting Rule-i cannot caused on changes of the policy.

From Theorem 1, we found that Shadowed rules can be occurred on the Listed-Rule Firewall.

2.2 Limitation about Swapping Position between Rules

Swapping position of two rules can cause policy changing on firewall if the two rules are in different actions, and both of them can be matched with the same packet. For example, swapping between Rule-7 and Rule-8 (see Table 1) will change the packet (with Source IP=10.3.3.9, Destination IP=20.3.3.9, Destination Port=80) from being accepted to be denied, and can cause a Shadowed rule.

In this limitation, security problems are likely to be occurred if denied dangerous packets change to be accepted. Moreover, changing the packets from being accepted to be denied can also be considered as security problem. For example, if packets which send / receive between clients and servers are blocked, it can be referred as security problem because of the lack of "Availability" (ready to use). Above limitations are proved as following.

Theorem 2. If $(R_i \bigcap R_{i+1}) \neq \phi$ and both rules (Rule-i and Rule-i+1) have different actions, swapping the positions of Rule-i and Rule-i+1 may cause some changes of policy.

For example, if we swap positions of Rule-7 and Rule-8 (in Table 1) the policy may be changed.

Proof of Theorem 2

Let

R_u be R_x before swapping position

R_v be R_y before swapping position

K be $(R_{x-1} \bigcup R_{x-2} \bigcup ... \bigcup R_1)$ (it is the group of packets that already match with previous rules above)

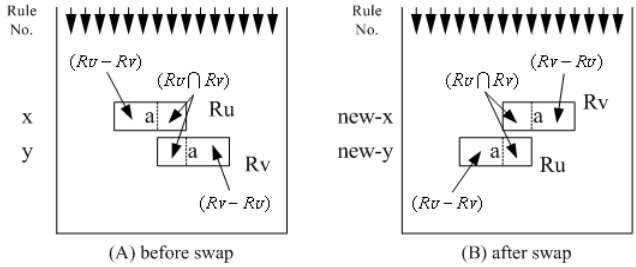


Fig. 2. The results from Difference and Intersection operation of Relations

Note: Operation without brackets means operating from the left to right hand. For example, A-B-C means (A-B)-C.

Consider packet (p) that will fall down to Rule-x

Before swapping (see Figure 2-A)

$$p \in R_u - K$$
$$p \in (R_u - R_v) \bigcup (R_u \bigcap R_v) - K \quad (1)$$

After swapping, consider the Rule-x (old) that was changed to the 'Rule-new-y' (Figure 2-B).

Consider packet (p) that will fall down to 'Rule-new-y'

$$p \in R_u - R_v - K$$
$$p \in (R_u - R_v) - (R_u \bigcap R_v) - K \quad .(2)$$

Consider packet (p) that will fall to Rule-y (Figure 2)

Before swapping (Figure 2-A)

$$p \in R_v - R_u - K$$
$$p \in (R_v - R_u) - (R_u \bigcap R_v) - K \quad (3)$$

After swapping, consider the Rule-y (old) that was changed to the 'Rule-new-x' (Figure 2-B)

$$p \in R_v - K$$
$$p \in (R_v - R_u) \bigcup (R_u \bigcap R_v) - K \quad (4)$$

These are the assumptions that will be used for proving the theorems 2 in the next step.

In this case, two rules have different action and $R_u \bigcap R_v \neq \phi$

Consider packet (p) that will fall to Rule-x (see Figure 2)

Before swapping: from (1) $p \in (R_u - R_v) \bigcup (R_u \bigcap R_v) - K$ (Figure 2-A)

From the properties of sets, $(B \bigcup C) - A = (B - A) \bigcup (C - A)$

Thus $$p \in ((R_u - R_v) - K) \bigcup ((R_u \bigcap R_v) - K) \quad (5)$$

After swapping: from (2) $p \in (R_u - R_v) - (R_u \bigcap R_v) - K$

Because $(R_u - R_v)$ does not overlap with $(R_u \bigcap R_v)$

Thus $$p \in (R_u - R_v) - K \quad (6)$$

From (5) and (6), we find that the decreased packets (matched with Rule-x) are

$$p \in (R_u \bigcap R_v) - K \quad (7)$$

Consider packet (p) that will fall to Rule-y (see Figure 2)

Before swapping: from (3) $p \in (R_v - R_u) - (R_u \bigcap R_v) - K$

Because $(R_u - R_v)$ does not overlap with $(R_u \bigcap R_v)$

Thus $$p \in (R_v - R_u) - K \quad (8)$$

After swapping: from (4) $p \in (R_v - R_u) \bigcup (R_u \bigcap R_v) - K$

From the properties of sets, $(B \bigcup C) - A = (B - A) \bigcup (C - A)$

Thus $$p \in ((R_v - R_u) - K) \bigcup ((R_u \bigcap R_v) - K) \quad (9)$$

From (8) and (9) we find that the increased packets (matched with Rule-y) are

$$p \in (R_u \bigcap R_v) - K \quad (10)$$

From (7) and (10), $p \in (R_u \bigcap R_v) - K$ are the packets which changed from being matched with Rule-x to be matched with Rule-y, while both rules have the different action. Therefore, if we swap the positions of the both rules, it causes a change in the policy.

2.3 Limitation about Redundant Rules

Redundant rules mean the rule that is redundant to other rule below with same action. For example, Rule-8 (Table1) is redundant to Rule-9. Another example, Rule-4

(Figure1) is redundant to Rule-5. If we remove the redundant rule, a firewall policy will not be changed. This is because packets which use to match with the redundant rule will change to match with next rules below which have same action.

In this limitation, a speed problem can be occurred because many redundant rules can waste firewall processing time. We prove that redundant rules are not necessary and can be deleted without any change to firewall policy.

Theorem 3. Suppose Rule-i, Rule-(i+1), Rule-(i+2), …, Rule-(i+n) all have the same action (where 'n' is a positive integer). If $R_i \subset R_{i+1} \bigcup R_{i+2} \bigcup ... \bigcup R_{i+n}$, then removing Rule-i can be done without any change of policy.

Proof of Theorem 3. It was defined that $R_A = R_i \bigcup R_{i+1} \bigcup R_{i+2} \bigcup ... \bigcup R_{i+n}$
Before removing Rule-i, the packets that will fall to Rule-i are

$$p \in R_i - (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1) \tag{11}$$

After removing Rule-i, these packets will fall to the next rules below such as Rule-(i+1), Rule-(i+2), ..., and Rule-(i+n).
But $R_i \subset R_A$, thus

$$R_i - (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1) \subset R_i \subset R_A$$
$$R_i - (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1) \subset R_A \tag{12}$$

From (11) and (12),

$$p \in R_i - (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1) \subset R_A$$

After removing Rule-i, therefore, packets that used to match with Rule-i will fall down to match with Rule-(i+1) or Rule-(i+2), ..., or Rule-(i+n). Due to Rule-i, Rule-(i+1), ..., Rule-(i+n) having the same action, there are no changes to the policy.

2.4 Limitation of Rule Design

In the rule design process, firewall administrators have to put "Bigger Rule" at only bottom or lower positions.

Note

- "Bigger Rule" is the rules that can be mapped into a "Bigger Relation"
- "Bigger Relation" is the relation that is bigger than other relations (super set of other relations) when comparing between two rules (or relations).

An example of a Bigger rule is the last rule (Rule-18 of Table2) which is bigger than every rules above. We cannot move this rule to the first position because it can shadow all other rules.

Another example of Bigger rule, if we want to prevent normal users from attacking the servers in DMZ (see Figure 3) but allow admin to manage servers through port 22, we have to prevent normal users using Rule-14 (in Table2) but allow admin by using

Rule-1 and Rule-2. As a result, Rule-14 is a bigger rule comparing to Rule-1 (and Rule-2), and we have to put Rule-14 under Rule-1 (and Rule-2).

Thus, with this limitation, designing rule on Listed-Rule Firewall can be done hardly because rule positions of each rule are not independent. Moreover, in the Listed-Rule Firewall, the rules that will be matched almost all packets (such as rule number 18) cannot be moved upward to other positions above. This also cause speed problem.

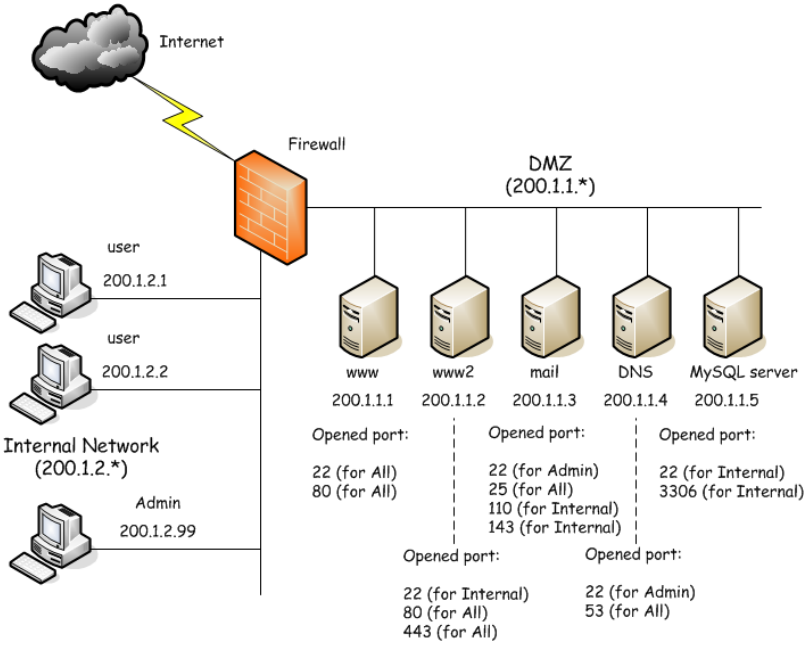


Fig. 3. A medium size network with DMZ

Table 2. An example of rules on a medium size network

No.	Source_IP	Dest_IP	Dest_Port	Action
1	200.1.2.99	200.1.1.3	22	Accept
2	200.1.2.99	200.1.1.4	22	Accept
3	200.1.2.*	200.1.1.2	22	Accept
4	200.1.2.*	200.1.1.5	22	Accept
5	200.1.2.*	200.1.1.3	110	Accept
6	200.1.2.*	200.1.1.3	143	Accept
7	200.1.2.*	200.1.1.5	3306	Accept
8	*	200.1.1.1	22	Accept
9	*	200.1.1.1	80	Accept
10	*	200.1.1.2	80	Accept
11	*	200.1.1.2	443	Accept
12	*	200.1.1.3	25	Accept
13	*	200.1.1.4	53	Accept
14	200.1.2.*	200.1.1.*	*	Deny
15	200.1.2.*	*	*	Accept
16	200.1.1.3	*	25	Accept
17	200.1.1.4	*	53	Accept
18	*	*	*	Deny

2.5 Limitation from Sequential Computation

Rule computing for packet decision on Listed-Rule Firewall is a sequential process. Consequently, it may cause speed problem. Especially, the firewall which has a large number of rules may work with a slow speed. The time used for rule computation would be depended on number of rules.

Let

Number of rules is N
Number of rules (in average) that will be compared with packets is N/2

Thus

In Big O aspect, the time which will be used to compute each packet is $t \in O_{(N)}$
In the next section, we will propose a new type of firewall which has a better Big O.

3 The Design of Tree-Rule Firewall

We have designed the new type of firewall which is called "Tree-Rule Firewall" as shown in Figure 4.

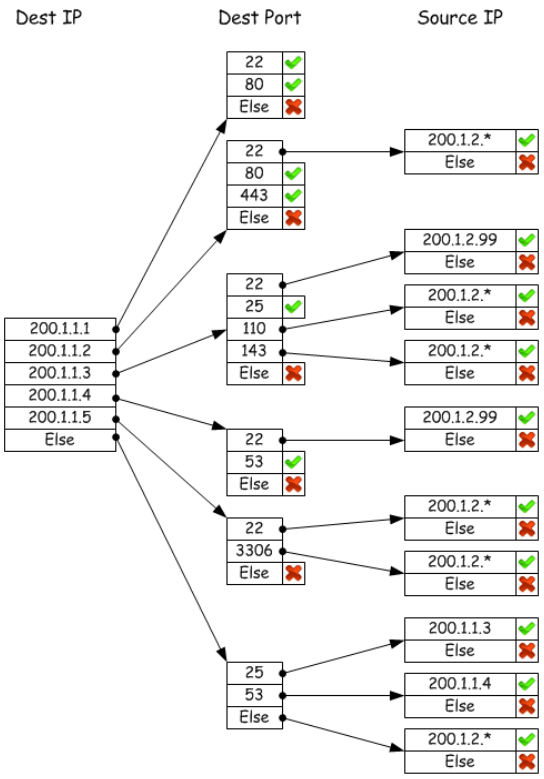


Fig. 4. The rule of Rule-Tree Firewall

This design can avoid limitations occurring on Listed-Rule Firewall. In this section, we will explain many advantages of Tree-Rule Firewall such as:

- No shadowed rule
- No swapping rule (the rule will be sorted automatically)
- No redundant rule
- Easy to design its rules (with independent rule path)
- High speed for packet decision

Rule-Tree Firewall is the new kind of firewall in which the rules are presented in a tree form (Figure4) instead of list of lines. Its processing (in a kernel level) is processed from its Tree Rule. This firewall will read attribute information from packet header and compare first packet's attribute with data in the root node of Tree Rule. After that, the firewall will check next packet's attribute by searching only on relevant node. As a result, the packet will be decided with specific action within a short time. For example, from the Figure 4, when the packets arrived at Tree-Rule Firewall, the Tree-Rule Firewall will consider Dest IP (destination IP address), Dest Port (destination port), and Source Port respectively until packets will be decided by a predefined action.

As we can see, the Tree-Rule Firewall has no security problem because the user cannot swap rule positions (Tree-Rule Firewall has no rules number, but we call each path of tree "Rule Path"). Data in each node will be sorted in ascending order.

Rule designers are not necessary to have any skill. They need only basic concept of Tree-Rule Firewall designing. This means that Tree-Rule Firewall's rules are is easy to design.

Moreover, the rules that will be matched almost all packets (such as the rule on the bottom path that show "Else->Else->Else->Deny") will take time of packet decision (Accept or Deny) equivalent to other rule paths. Each rule (each decision path) will take the same interval of time to decide packet.

With regards to performance, Big O of a decision time in Listed-Rule Firewall is O(n); where n is number of rules. While Big O the decision time in Tree-Rule Firewall is logarithm. For example, in average case of Listed-Rule Firewall, if we assume that the chance of matched packets are equal for each rule in Table 2 (N=18, compare with 3 attributes (Source IP address, Destination IP address, and Destination Port)), we found that it will takes (18/2) * 3 * C = 27C (where C is the time interval that is used for comparing between "1 attribute of packet header" and "1 attribute of rule"). While Tree-Rule Firewall in Figure 4 (Dest IP = 6 lines, Dest Port = 4 lines (in average), Source IP = 2 lines (in average)) will takes a time less than C*Log 8 + C*Log 4 + C*Log 2 = C*(3+2+1) = 6C.

Note: All "Log" values are base 2 logarithm.

When we conduct our survey on some enterprise networks that consists of approximately 100 servers, each server open about 20 ports, and has approximately 5

groups of users which need to access DMZ and Internet, we found that there are at least 200 rules on Listed-Rule Firewall which have to be defined for dealing with this condition. In average case, it has takes time about (200/2)*3*C = 300C to decide 1 packet, and requires 200*3*C = 600C in the worst case. While the time taken for 1 packet in Tree-Rule Firewall is less than C*Log 128 + C*Log 32 + C*Log 8 = C*(7+5+3) = 15C in both average case and the worst case scenarios.

Note:

128 is the number of destination IP addresses rounded up from 100.
32 is the number of destination Port rounded up from 20.
8 is the number of source IP addresses rounded up from 5.

As we can see, the Big O of Tree-Rule Firewall is obviously different from Listed-Rule Firewall's Big O. This is similar to a comparison between “Binary Search Tree” [10], and “Linear Search” [11] in an Array.

Considering the “security” aspect, Listed-Rule Firewall in enterprise networks (that has many rules) is likely to encounter with rules confliction of rules. Our scheme can address such conflictions including problems of shadowed and redundant rules. The rule path of Tree-Rule Firewall has no rule conflict within its rule base, because Tree-Rule Firewall does not have any Shadowing, Correlation or Redundancy anomaly.

In the “easy to use” aspect, it is very difficult to design rule in Listed-Rule Firewall following the policy of a corporate. Because, even we write many lines of rule, it will be difficult to check and test working of each rule. On the contrary, Tree-Rule Firewall can be designed easily, because every rule sentence (rule paths) of Tree-Rule Firewall are walk in separate paths obviously.

4 Conclusion

In this paper, we identified five important limitations on Listed-Rule Firewall which may lead to security problem, speed problem, and "difficult to use" problem. These limitations consist of (1) limitation about "Shadowed rules" which can lead to security and speed problem, (2) limitation about swapping position between rules which cause security problem, (3) limitation about "Redundant rules" which can cause speed problem, (4) limitation of rule design that can result in a "difficult to use" problem, and (5) limitation from sequential computation that can lead to speed problem. We presented various theories and their proof to validate our arguments for the above limitations. We developed the Tree-Rule Firewall which will be fast, secure, and easy to use. In our future work, we will discover equation of Big O precisely for comparing its performance with Listed-Rule Firewall. We will also study other factors that may impact the speed and security of Tree-Rule Firewall.

References

1. Al-Shaer, E., Hamed, H.: Firewall Policy Advisor for anomaly Detection and Rule Editing. In: IEEE/IFIP Integrated Management, IM 2003 (March 2003)
2. Al-Shaer, E., Hamed, H., Boutaba, R., Hasan, M.: Conflict classification and analysis of distributed firewall policies. IEEE Journal on Selected Areas in Communications (JSAC) 23(10), 2069–2084 (2005)
3. Haded, H., Al-Shaer, E.: Taxonomy of conflicts in network security policies. IEEE Communications Magazine 44(3), 134–141 (2006)
4. Hazelhusrt, S.: Algorithms for Analyzing Firewall and Router Access Lists. Technical Report TR-WitsCS-1999, Department of Computer Science, University of the Witwatersrand, South Africa (July 1999)
5. Eronen, P., Zitting, J.: An Expert System for Analyzing Firewall Rules. In: Proceedings of 6th Nordic Workshop on Secure IT-Systems (NordSec 2001), pp. 100–107 (2001)
6. Lihua, J.M., Su, Z.: FIREMAN: A toolkit for Firewall modeling and analysis. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy (2006)
7. Cisco Content Services Switch Basic Configuration Guide, `http://www.cisco.com/en/US/docs/app_ntwk_services/data_center_app_services/css11500series/v7.10/configuration/basic/guide/basicgd.pdf`
8. Liu, A., Gouda, M.: Diverse Firewall Design. IEEE Transaction on Parallel and Distributed Systems 19(9) (September 2008)
9. Pornavalai, C., Chomsiri, T.: Firewall Policy Analyzing by Relational Algebra. In: Proceeding of the 2004 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2004), pp. 214–219 (2004)
10. `http://en.wikipedia.org/wiki/Binary_search_tree`
11. `http://en.wikipedia.org/wiki/Linear_search`