

# Firewall Rules Analysis

**Thawatchai Chomsiri**  
**Faculty of Informatics**  
**Maharakham University**  
**Maharakham, Thailand**

**Chotipat Pornavalai**  
**Faculty of Information Technology**  
**King Mongkut's Institute of Technology Ladkrabang**  
**Bangkok, Thailand**

*Abstract - In this paper, we propose a method to analyze the firewall policy or rule-set using Relational Algebra and Raining 2D-Box Model. It can discover all the anomalies in the firewall rule-set in the format that is usually used by many firewall products such as Cisco Access Control List, IPTABLES, IPCHAINS and Check Point Firewall-1. While the existing analyzing methods consider the anomalies between any two rules in the firewall rule-set, we consider more than two rules together at the same time to discover the anomaly. Therefore we can find all the hidden anomalies in the firewall rule-set. Results from analyzing can be used with the proposed rules-combination method presented in this paper, to minimize the firewall rule without changing the policy. Finally, we have developed an application based on the proposed analyzing method. This application could help administrator to analyze and modify a complex firewall policy with less error.*

**Keywords:** firewall, policy, rules, relational algebra

## 1.0 Introduction

Conflicts and incorrect order within firewall rules can make system work improperly. Writing a rule-set usually contain many hidden conflicts. Pasi Eronen [2] proposed an Expert System that is based on constraint logic programming (CLP) for user to write higher-level operations for detecting common configuration mistakes. Scott Hazelhurst [3] using Binary Decision Diagrams (BDDs) to present and analyze rule-set. Ehab Al-Shader et.al [1] presents an anomaly discovery algorithm using SET theory. They presented a method for finding some error within rule-set that called “anomaly”. But their research can not discover all anomalies when considering more than two rules at the same time.

In this paper, we propose an alternative approach using Relational Algebra (RA) technique and Raining 2D-Box Model for finding anomaly within the rule-set. It can discover all the hidden anomaly when considering more two rules together. This paper is organized as follows. In section 2, we present how to map the firewall rules into Relation [4] using Cartesian product. In section 3, we classify and define firewall policy anomaly. In section 4, we present how to remove anomaly and how to reduce the rule-set’s size by combining many rules together. Application of the proposed method is also presented in section 5. We conclude this paper in section 6.

## 2.0 Firewall and Relational Algebra Background

Firewall can allow or deny any packets by considering the specified rule-set. Example of rule-set is shown in Fig. 1.

Order	src_ip	dst_ip	dst_port	Action
1	10.0.0.5	20.0.0.1	20-21	accept
2	10.0.0.5	20.0.0.1	21-22	accept
3	10.0.0.5	20.0.0.1	23-25	deny
4	10.0.0.5	20.0.0.1	22-23	accept
5	10.0.0.0/30	20.0.0.1	80	deny
6	10.0.0.1	20.0.0.0/30	80	accept
7	10.0.0.0/30	20.0.0.0/30	80	deny
8	10.0.0.1	20.0.0.1	80-143	accept
9	10.0.0.1, 10.0.0.4	20.0.0.1	80	deny
10	10.0.0.0/24	20.0.0.0/24	any	accept
11	10.0.0.0/24	20.0.0.0/24	80	deny
12	any	any	any	deny

Figure 1. example of firewall rule-set

Relation is a subset of Cartesian product of domain [4]. Relation Algebra is a procedural query language that consists of a set of operation on the Relation(s). Example of operation in Relational Algebra are SELECT, PROJECT, UNION, and DIFFERENCE (See Fig. 2). Example; when src\_ip = 30.0.0.0/30, dst\_ip = 40.0.0.1, dst\_port = 80, and action = deny, it can be mapped to Relation 1 (R1) as shown in Fig. 2, while R2 is the relation that is mapped from rule; src\_ip = 30.0.0.0/31, dst\_ip = 40.0.0.1, dst\_port = 80-81, action = accept.

R1 (Relation 1) is

src_ip	dst_ip	dst_port	action
30.0.0.0	40.0.0.1	80	deny
30.0.0.1	40.0.0.1	80	deny
30.0.0.2	40.0.0.1	80	deny
30.0.0.3	40.0.0.1	80	deny

R2 (Relation 2) is

src_ip	dst_ip	dst_port	action
30.0.0.0	40.0.0.1	80	accept
30.0.0.0	40.0.0.1	81	accept
30.0.0.1	40.0.0.1	80	accept
30.0.0.1	40.0.0.1	81	accept

PROJECT

R3 = project (src\_ip, dst\_ip, dst\_port) R1

src_ip	dst_ip	dst_port
30.0.0.0	40.0.0.1	80
30.0.0.1	40.0.0.1	80
30.0.0.2	40.0.0.1	80
30.0.0.3	40.0.0.1	80

R4 = project (src\_ip, dst\_ip, dst\_port) R2

src_ip	dst_ip	dst_port
30.0.0.0	40.0.0.1	80
30.0.0.0	40.0.0.1	81
30.0.0.1	40.0.0.1	80
30.0.0.1	40.0.0.1	81

SELECT

R5 = select (dst\_port=80) R4

src_ip	dst_ip	dst_port
30.0.0.0	40.0.0.1	80
30.0.0.1	40.0.0.1	80

UNION

R6 = R3 union R4

src_ip	dst_ip	dst_port
30.0.0.0	40.0.0.1	80
30.0.0.1	40.0.0.1	80
30.0.0.2	40.0.0.1	80
30.0.0.3	40.0.0.1	80
30.0.0.0	40.0.0.1	81
30.0.0.1	40.0.0.1	81

INTERSECTION

R7 = R3 intersect R4

src_ip	dst_ip	dst_port
30.0.0.0	40.0.0.1	80
30.0.0.1	40.0.0.1	80

DIFFERENCE

R8 = R3 difference R4

src_ip	dst_ip	dst_port
30.0.0.2	40.0.0.1	80
30.0.0.3	40.0.0.1	80

Figure 2. Relational Algebra Operation

### 3.0 Firewall Policy Anomaly Discovery

In this section, we define and classify the types of anomaly using Raining 2D-Box Model. *Rule-x* is defined as a rule number  $x$  from the rule-set table. Therefore, if  $x < y$ , then *Rule-y* follows *Rule-x*.  $R_x$  is a relation that has been mapped from *Rule-x* using PROJECT operation to exclude the *action* attribute.

Ehab S. Al-Shaer et al [1] classified anomaly into 4 types. They are Shadowing anomaly, Correlation anomaly, Generalization anomaly and Redundancy anomaly. We also classify anomaly into 4 types but considering more than two rules at the same time. We also present many definitions and that will be used to discover anomalies. The formal proposed definitions and algorithm can be found in our recent publication [5]. There are 6 theorems that come out as the result of analyzing in these 4 anomalies types. More detail about proved of theorem 1-6 is presented in [6].

#### 3.1 Shadowing Anomaly

Shadowed rule is a rule which will never be executed because all the packets that matched this rule are already matched by the one or more rules that are written above in the rule-set. For example, *Rule-4* is shadowed because  $R_4 - (R_3 \cup R_2 \cup R_1) = \emptyset$ . No packet can be passed to *Rule-4* (see Fig. 1 or Fig. 3(a)). In Ehab S. Al-Shaer [1] research, shadowed rule is defined as a rule that is shadowed by only one previous rule which has different action. In this example, it would be *rule-11* which shadowed by *rule-10*. Thus their algorithm can not classify that *rule-4* is shadowed. But in fact, *rule-4* is shadowed by two rules (*rule-2* and *rule-3*).

We represent anomalies as a two-dimension box that contains relations that are mapped from rules in the order described in the rule-set. A rectangular is used to represent relation of the rule and specified action within each box. If action is not specified in the rectangular, it can be any actions (accept or deny). This model simulates packets that fall from the top to the bottom like raining. For example, when the part of the relation in the box is not wet, means that it is shadowed, as shown in Fig. 3 (a). When shadowed rules are discovered, we should remove them to reduce the size of rule-set.

*Theorem 1: The firewall policy does not change even if we remove Rule-x, when Rule-x is shadowed.*

#### 3.2 Correlation Anomaly

Two rules (with different actions) are correlated if the first rule in order matches some packets that matched the second rule and the second rule matches some packets that match the first rule. This definition is similar to [1]. But in their definition, two rules (*Rule-x* and *Rule-y*) are correlated if some fields in *Rule-x* are sub-sets or equal to the corresponding field in *Rule-y*, and the rest of the fields in *Rule-x* are super-sets of the corresponding fields in *Rule-y*. By using their definition, *Rule-5* and *Rule-6* are considered to be correlated.

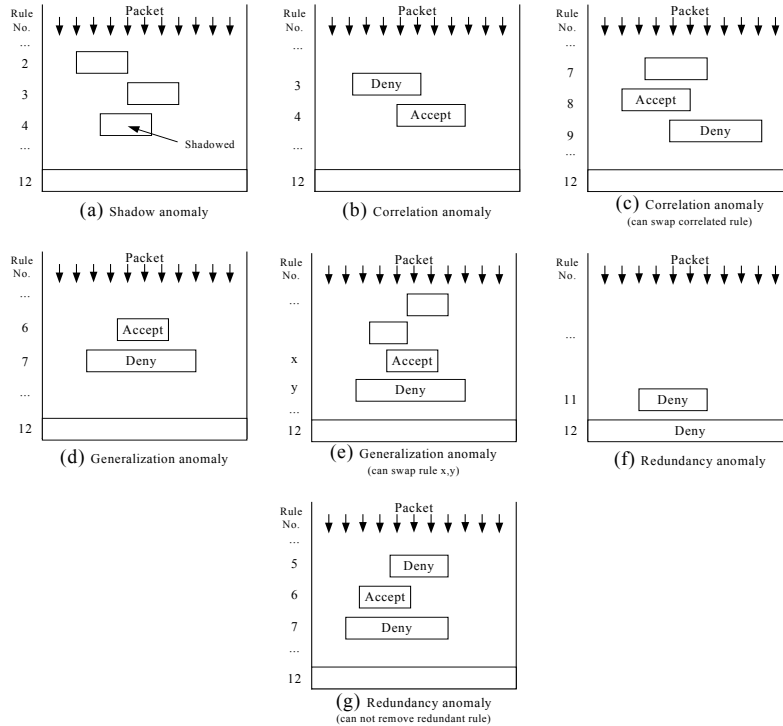


Figure 3. Raining 2D-Box Model

Another example of correlation anomaly is *Rule-3* and *Rule-4* (see Fig. 1. and Fig. 3(b)). However, research [1] will not be discovered by their algorithm because both *Rule-3* and *Rule-4* have one attribute (*dst\_port*) that is partially correlated, i.e this field is not the subset. There are many firewall products available in the market that allows this kind of correlation to be occurred, such as when using port-range attribute (field) in IPTABLES and multi-address in Check Point Firewall-1. Therefore using our definition, we can discover this kind of correlation.

However, even though *Rule-x* and *Rule-y* are correlated, we may swap the order of these two rules without policy-change in some cases, for example, *Rule-8* and *Rule-9*. The reason for this example comes from the fact that correlated part between *Rule-8* and *Rule-9* is shadowed by previous rules (in this case *Rule-7*). It can also be explained using Raining 2D-Box model in Fig. 3(c). There are 6 theorems (theorem 1-6) that we proof in [6] and publish in [5]. In this, we proof 7-8.

*Theorem 2: The firewall policy does not change even if we swap Rule-x and Rule-y, when Rule-x and Rule-y are consecutively non-correlated.*

*Theorem 3: The firewall policy does not change even if we swap Rule-x and Rule-y, where  $x < y$ , and Rule-x is consecutively non-correlated downward to Rule-y, and Rule-y is consecutively non-correlated upward to Rule-x.*

*Theorem 4: The firewall policy does not change even if we swap Rule-x and Rule-y, where Rule-x and Rule-y are correlated and Rule-xy is shadowed, and Rule-x is consecutively non-correlated downward to Rule-(y-1), and Rule-y is consecutively non-correlated upward to Rule-(x+1). Note: Rule-xy is rules that unmapped from  $R_x \cap R_y$*

### 3.3 Generalization Anomaly

A rule is said to be a generalization of the previous rule if it matches all the packets that matched previous rule, when actions are different. For example, *Rule-7* is a generalization of *Rule-6*, see Fig. 3(d).

In general, two rules that are generalized can not be swapped. But in some cases, such as shown in Fig. 3(e), when *Rule-y* is generalized to *Rule-x*, but it can be swapped because *Rule-x* is shadowed by previous rules.

### 3.4 Redundancy Anomaly

In general, when two rules are considered to be redundant, we should be able to delete Redundant-Rule. For example, as shown in Fig. 3 (f), we can remove *rule-11* without any changes in firewall policy. However, we cannot remove *Rule-5*, although it is redundant to *Rule-7* (see Fig. 1 and Fig. 3 (g)). If we remove *Rule-5*, Firewall will accept packet came from 10.0.0.1 and destination to 20.0.0.1 at port 80 (the intersection part of R5 and R6).

By considering more than two rules at the same time, we can discover anomaly that cannot be explained by using SET approach. For example, Firewall Policy Advisor in [1] will recommend user to remove *Rule-5* because it is redundant by *Rule-7*, which in fact it may not be removed as explained.

*Theorem 5: The firewall rule does not change even if we remove Rule-x from the Rule List when Rule-x is consecutively redundant by Rule-y.*

*Theorem 6: The firewall rule does not change even if we remove Rule-x from the Rule List when Rule-x is redundant by Rule-y, and Rule-x is consecutively non-correlated downward to Rule-(y-1).*

## 4.0 Anomaly Removing and Rules Combination

As explained in section 3, we should remove rules that are shadowed, and rules that are redundant (with some exceptions). Also we should alert administrator when generalization anomaly or correlation anomaly are discovered. Removing anomalies (shadowing and redundancy), and “rules combination” method (will be explained below) can shorten the size of rule-set and make firewall policy easier to understand.

Reordering the rules in the rule-set may also help administrator understand the rule-set easier. It also can increase the performance of the firewall because the rules that are matched by many packets are on the top in the rule-set. We can combine many rules together by using the UNION operation to the Relations. We present theorem 7 to describe the rules combination.

*Theorem 7: Rule-x and Rule-y can be combined to Rule-z, where  $R_z = R_x \cup R_y$ , and actions are same, and  $y=x+1$ , without changing the policy.*

For proving, we will guide reader to understand something about Relation (which mapped from Rule) in 2D-Box Model.

If  $p$  is the set of packets which falling to Rule-1, we found that  $p \in R_1$  ( See 2D-Box-Model )

If  $p$  is the set of packets which falling to Rule-2, we found that  $p \in R_2 - R_1$

If  $p$  is the set of packets which falling to Rule-3, we found that  $p \in R_3 - R_2 - R_1$

If  $p$  is the set of packets which falling to Rule-i, we found that  $p \in R_i - R_{i-1} - R_{i-2} - \dots - R_1$

By using the SET Theory in Mathematic, set of packets which falling to Rule-i is  $p \in R_i - (R_{i-1} \cup R_{i-2} \cup \dots \cup R_1)$

*Proof of Theorem 7:*

Although we swap Rule-x with Rule-y or remove each rule, packets which falling to any rules above Rule-x and Rule-y will not changes. Thus we consider 2 cases according.

(1) Packets which falling to Rule-x or Rule-y

(2) Packets which falling to rules below Rule-y

Define  $A = R_{x-1} \cup R_{x-2} \cup \dots \cup R_1$

$R_z = R_x \cup R_y$

*Consider (1)*

Packets which falling to Rule-x are in set  $p$ , whenever  $p \in R_x - A$ .

Because of  $y = x+1$ , thus packet which falling to Rule-y are in  $p \in R_y - R_x - A$

Thus packet that fall to Rule-x or Rule-y are

$$p \in (R_x - A) \cup (R_y - R_x - A)$$

$$p \in (R_x \cup (R_y - R_x)) - A$$

$$p \in (R_x \cup R_y) - A$$

$$p \in R_z - A$$

From equation, we found that  $p$  equal to packets which falling to Rule-z.

Because of Rule-x and Rule-y are same action, thus after combine rules packets which ever match Rule-x or Rule-y will be change to match Rule-z and decide with old Action.

*Consider (2)*

Define  $S$  is Sample Space of all packet.

Before we combine Rule-x with Rule-y, packets which falling pass Rule-y to another rules below (packets which not match Rule-y and rules above) are in set  $p$

$$p \in S - R_y - R_x - A$$

$$p \in S - (R_x \cup R_y) - A$$

$$p \in S - R_z - A$$

From equation,  $p$  is set of packets are can fall pass Rule-z to another rules below. (After combine rules)

From (1) and (2) we can combine rules with out policy change.

For example, in Fig. 4, by using theorem 7 above, rule can be combined one by one without changing the policy. The combined rule is the union of R1, R2, R3, and R4. In fact, it is the rule summarization of Rule-1 to Rule-4. Fig. 5 shows the rule combination using Raining 2D-Box model.

Before				
Order	src_ip	dst_ip	dst_port	Action
1	50.0.0.0/26	60.0.0.0/24	80	accept
2	50.0.0.64/26	60.0.0.0/24	80	accept
3	50.0.0.128/26	60.0.0.0/24	80	accept
4	50.0.0.192/26	60.0.0.0/24	80	accept

After				
Order	src_ip	dst_ip	dst_port	Action
1	50.0.0.0/24	60.0.0.0/24	80	accept

Figure 4. Rules Combination (1)

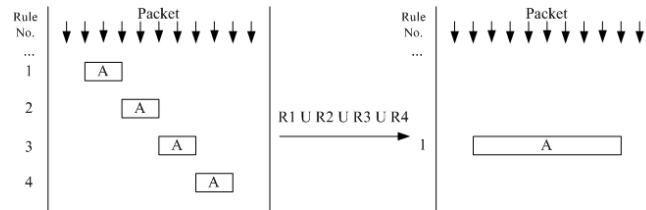


Figure 5. Rules Combination (2)

Although the Theorem 7 can works only on consecutive rules, it is used as the basis of other combinations. The rules may not be consecutively ordered in the rule-set. So we need to re-order them, but we also need to ensure that it will not change the firewall policy using our proposed algorithm in [5]. Fig. 6(a) shows an example of the rule-set. After we move rule-3 to the top and swapping rule-5 with rule-6, firewall policy does not change, as shown in Figure 6(b). Then rules 2-5 are then combined using Theorem 7, as shown in Fig 6(c).

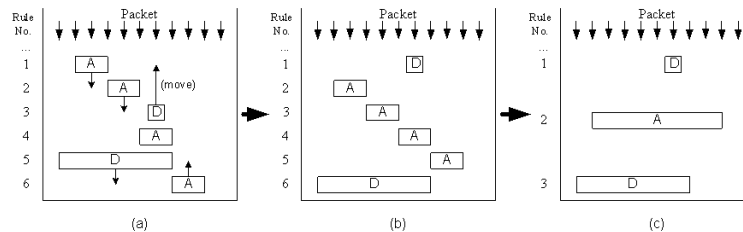


Figure 6. Rules Combination (3)

In [5], we have defined “consecutively redundant rule” as a rule that is redundant to the next rule in the rule-set. We also presented a theorem and definition in [5] showing that policy will not be changed if we remove consecutively redundant rule. Reverse to this theorem, we create theorem 8 below to show that inserting consecutively redundant rule does not change the policy as well. This theorem can be used to combine the rules. For example in Fig. 7(a), a new rule (rule-6 in Fig. 7(b)) can be inserted, without changing the policy. Then we can use the same techniques to move rules up or down, as shown in Fig. 7(c). The result after we combined rules 1-4 using theorem 7 is shown in Fig. 7(d).

**Theorem 8:** The firewall policy does not change even if we insert consecutively redundant rule.

*Proof of Theorem 8:*

Before insert consecutive rule, the (old) rule order is  $x$ . After we insert consecutive rule above line number  $x$ , order of the old rule is  $y$ ;  $y = x + 1$  ( see figure 7(a) and 7(b) )

$$\text{Define } A = R_{x-1} \cup R_{x-2} \cup \dots \cup R_1$$

Because of Rule- $x$  is Consecutive Redundant of Rule- $y$ , Thus

$$R_x \subset R_y$$

$$y = x + 1$$

Before we insert Consecutive Redundant rule, all packets which falling to Rule- $y$  are in set  $p$

$$p \in R_y - A \dots \dots \dots (a)$$

After we insert Consecutive Redundant rule, all packets which falling to Rule- $y$  are in set  $p$

$$p \in R_y - R_x - A$$

$$p \in R_y - (R_y \cap R_x) - A$$

$$p \in (R_y - A) - ((R_y \cap R_x) - A) \dots \dots \dots (b)$$

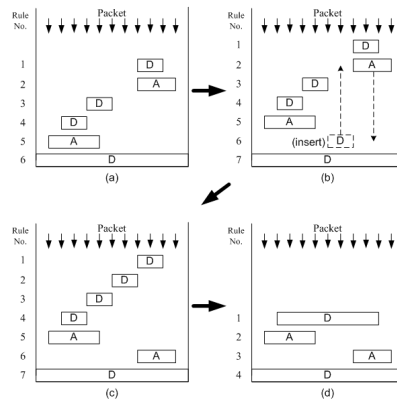


Figure 7. Rules Combination (4)

From equation (a) and (b), Packets that falling to Rule-y has loss as  $p \in (R_y \cap R_x) - A$

Before we insert Consecutive redundant rule (Rule-x), all packets which falling to Rule-x are in set  $p$

$$p \in \phi \dots\dots\dots(c)$$

( $p$  is blank set because no have Rule-x in first time)

After we insert Consecutive redundant rule (Rule-x), all packets which falling to Rule-x are in set  $p$

$$p \in R_x - A$$

Because of  $R_x \subset R_y$ , Thus  $p \in (R_y \cap R_x) - A \dots\dots\dots(d)$

From equation (c) and (d) we find that packets which falling to Rule-x is increase as  $p \in (R_y \cap R_x) - A$

After insert;  $p \in (R_y \cap R_x) - A$  that ever fall to Rule-y will falling to Rule-x.

Rule-x and Rule-y are same action, thus no have any change of policy.

## 5.0 Application and User Interface

In this paper, we proposed a technique to analyze firewall rule-set using Relational Algebra technique and Raining 2D-Box Model. It can discover anomalies that are occurred in the firewall rule-set. We also present theorems to remove and combine rules to minimize the size of firewall rule-set without changing the policy. Many related works are either complex, or can not be used to discover the anomalies presented in this paper. We have developed application using the proposed techniques. This application can help administrator analyze firewall rule-set on many commercial and open-source firewall products such as Checkpoint Firewall-1 and IPTABLES.

### 5.1 Anomaly Discovery

Fig. 8 shows shadowing anomaly from the example firewall rule-set. The result is shown in the bottom-left text box, in which rule-4 is shadowed by rules 2-3. Also rule-11 is shadowed by rules 5-7, and rules 9-10. This anomaly cannot be discovered by using application implemented in [1].

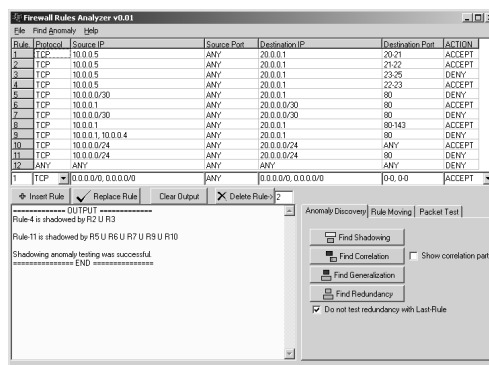


Figure 8. User Interface and Shadowing Anomaly Discovery

The application also allows administrator to search for correlation anomaly, with the option to show the overlap part of those rules. Administrator can use this information to correct the correlation anomaly. Fig. 9 shows the search result for redundancy anomaly. It shows recommendations or warnings that are generated by the application, such as “even though rule-5 is redundant with rule-7, but removing rule-5 will change the firewall policy”, as already explained in Figure 3(g). The application in [1] will recommend user to remove rule-5.

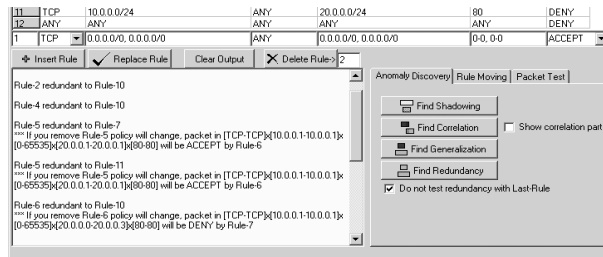


Figure 9. Redundancy Anomaly Discovery

## 5.2 Rules Reordering

On the main interface, when clicking on tab “Rule Moving”, the administrator can move or reorder the rules. The application can check whether this change will have any effects to the current firewall policy or not. This feature can help administrator moving the rules that are expected to match more frequently than the others to the top, and ensuring that it does not change the current firewall policy.

## 5.3 Analyze packet on each rule

We can test whether the packet will be matched on which rule in the firewall rule-set by selecting “Packet Test” tab. For example, we can answer question such as “Which destinations are allowed IP 10.0.0.7 to connect to HTTP port?” The result of the above question is shown in Fig. 10.

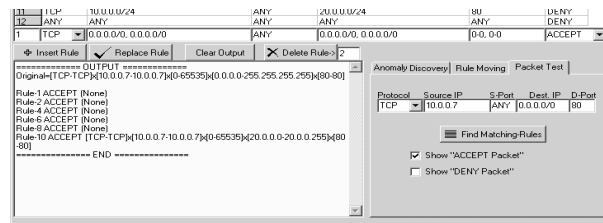


Figure 10. Destinations that allow 10.0.0.7 to connect to HTTP port are 20.0.0.0 – 20.0.0.255

## 6.0 Conclusion

In this paper, we proposed a technique to analyze firewall rule-set using Relational Algebra technique and Raining 2D-Box Model. It can discover anomalies that are occurred in the firewall rule-set. We also present theorems (and proof of theorem) to remove and combine rules to minimize the size of firewall rules without changing the policy. Many related works are either complex, or can not be used to discover the anomalies presented in this paper. We have developed application using the proposed techniques. This application can help administrator analyze firewall rule-set on many commercial and open-source firewall products such as Checkpoint Firewall-1, Cisco Access Control List, IPCHAINS and IPTABLES.

## 7.0 References

- [1] Ehab Al-Shaer and Hazem Hamed. "Firewall Policy Advisor for anomaly Detection and Rule Editing". IEEE/IFIP Integrated Management IM'2003, March 2003.
- [2] P. Eronen and J. Zitting. "An Expert System for Analyzing Firewall Rules". Proceedings of 6<sup>th</sup> Nordic Workshop on Secure IT-Systems (NordSec 2001), November 2001.
- [3] S. Hazelhurst. "Algorithms for Analyzing Firewall and Router Access Lists". Technical Report TR-WitsCS-1999 Department of Computer Science, University of the Witwatersrand, July 1999.
- [4] Abraham Silberschatz, Henry F. Korth, Sudharsan S. "Database System Concepts, 3rd Edition". Tata McGraw-Hill, 1997.
- [5] Chotipat Pornavalai and Thawatchai Chomsiri. "Firewall Policy Analyzing by Relational Algebra". The 2004 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2004), JULY 2004.
- [6] Chotipat Pornavalai and Thawatchai Chomsiri. "Firewall Policy Analyzing by Relational Algebra". Draft Technical Report, Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Thailand, January 2004.