2011 International Conference on Network and Electronics Engineering
IPCSIT vol.11 (2011) © (2011) IACSIT Press, Singapore

The Theories for Analyzing Matched Packets on Cisco ACL Rules

Thawatchai Chomsiri, Preecha Noiumkar

Faculty of Informatics, Mahasarakham University, Thailand
thawatchai@msu.ac.th, preecha.n@msu.ac.th

Abstract. This paper, we proposed the theories that can be used to analyze a matching between packets and rules of Cisco ACL. This can help administrators to understand conflicts that will be occurred on ACL rules. The conflicts such as, a couple of rules that able to be swapped (or should not to be swapped), and some rules that able to be removed without changing of a policy. Our proposed theories are based on basic ideas. They are easy to use for Cisco ACL rules which are not complicate. We propose seven theories and we have already proved them with mathematics and our proposed model for showing that they are accurate. Our proposed theories can be developed or extended for analyzing other complex firewalls' rules.

Keywords: Cisco, ACL, Firewall, Rule, Network, Security.

1. Introduction

Firewall is an important device that can help improve network security. The security levels do not depend on a price of firewall but they come from the secure rules inside. Studying about firewall configuration, we should focus on creating accuracy and no conflict rules set. We should begin with a simple firewall such as Cisco ACL before studying on the other complex firewalls. This research, therefore, we begin with Cisco ACL. We created our model for explaining a matching between packets and ACL rules. We proposed our theories and proved that they are accurate. The detail of the theories is in section 3 and the proof of them is in section 4.

2. Background and Related Works

Cisco ACL is a simple firewall existed on Cisco router. An example of ACL is shown below

```
router# show access-list
  Extended IP access list 101
       permit tcp host 10.1.1.1 host 20.1.1.1 eq www
       deny   tcp host 10.1.1.2 host 20.1.1.1 eq www
       deny   tcp 10.1.1.0 0.0.0.255 host 20.1.1.1 eq www
       permit tcp host 10.1.1.3 host 20.1.1.1 eq www
       deny   tcp 10.2.2.0 0.0.0.255 host 20.2.2.5 eq www
       deny   tcp host 10.2.2.5 20.2.2.0 0.0.0.255 eq www
       permit tcp 10.3.3.0 0.0.0.255 host 20.3.3.9 eq www
       deny   tcp host 10.3.3.9 20.3.3.0 0.0.0.255 eq www
       deny   ip  any  any
```

There are many users do not feel easy with wildcard mask of ACL. Thus, they are often design the ACL rules in easier format (we called "normal form") before applying these rules on a Cisco router configuration.

Table1. Firewall Rules (in normal form) translated from Cisco ACL

No.	Protocol	Source	Destination	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

There are many researches studied about firewall rules conflictions (anomalies) that occur within rule set including Cisco ACL rules. E-hab Al Shaer [1] proposes several anomaly definitions including “shading anomaly”. He defined the “shadowed rule” as the rule that can not match with any packet. For example, rule number 4 (see Table1) is a shadowed rule. This type of rule can be removed from rule list without changing of the policy. Moreover, he has applied his definition and theories for analyzing a distributed firewall [2]. In [1] and [2], he focused on Cisco ACL. Scott Hazelhurst [3] uses Binary Decision Diagrams (BDDs) to present and analyze rule set. Pasi Eronen [4] proposed an Expert System that is based on constraint logic programming (CLP) for user to write higher-level operations for detecting common configuration mistakes and find packet match on each rules.

In this paper, we propose an alternative but simple approach using Cartesian product [5] of each field in the rule and use Relational Algebra [5] Operations to find packet matching within each rule. Moreover, we propose simple model and theories for explaining a matching between packet and firewall rule. Proposed theories are suitable for Cisco ACL because Cisco ACL is very simple. We used mathematics and proposed model to prove that our theories are accurate.

3. Theories

We would like to propose our theories that can be used for explaining a matching between packet and ACL. They consist of following 7 theories.

Notation:

Rule-i denotes Rule number i.

Ri denotes relation that computed from Rule-i.

R denotes sample relation.

Note: Relation is subset of Cartesian product of domain. For example,

Suppose Rule-x is

No.	Protocol	Source	Destination	Dest Port	Action
x	TCP	10.1.1.1	20.2.2.0/30	80-81	Accept

Therefore, Rx is

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

Theory 1 The incoming packets $p \in R$ that will match with Rule-1 are in $R_1 \bigcap R$

This theory informs us to known that which packets are matched with Rule-1.

Theory 2 The incoming packets $p \in R$ that will match with Rule-i are in $(R_i \bigcap R) - (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1)$

For example, packets that will match with Rule-3 (see Table 1) are the remaining packets from matching with Rule-1 and Rule-2.

Theory 3 If Rule-i has no chance to match with any packet (due to all packets are already matched with other rules above), we can remove Rule-i without any change of a policy.

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

Theory 4

If $(R_i \bigcap R_{i+1}) = \phi$ then swapping a position between Rule-i and Rule-i+1 can be done without any change of a policy.

For example, we are able to swap a position between Rule-1 and Rule-2 (in Table 1). Likewise, Rule-4 and Rule-5 can be swapped as well.

Theory 5

If $(R_i \bigcap R_{i+1}) \neq \phi$ and both rules (Rule-i and Rule-i+1) are in the same action, swapping a position between Rule-i and Rule-i+1 can be done without any change of a policy.

For example, we are able to swap a position between Rule-5 and Rule-6 (in Table 1).

Theory 6

If $(R_i \bigcap R_{i+1}) \neq \phi$ and both rules (Rule-i and Rule-i+1) are in different action, swapping a position between Rule-i and Rule-i+1 may cause some changing of a policy.

For example, if we swap a position between Rule-7 and Rule-8 (in Table 1) a policy may be changed.

Theory 7

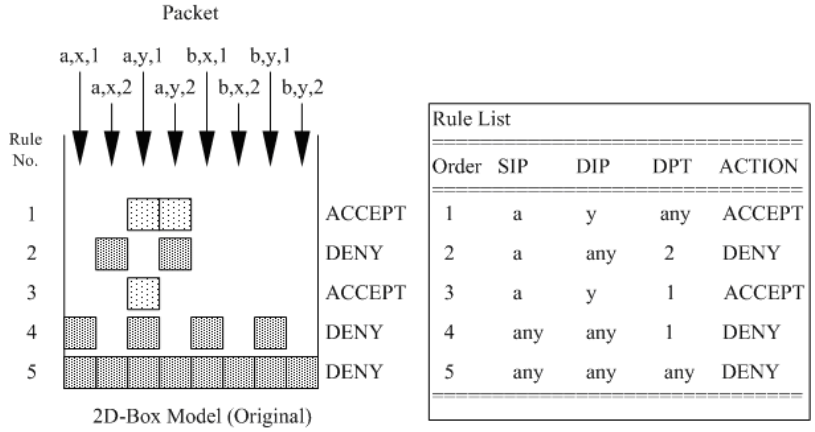
If $R_i \subseteq R_{i+1}$ and both rules (Rule-i and Rule-i+1) are in the same action, removing Rule-i can be done without any change of a policy.

For example, we are able to remove Rule-8 (in Table 1) without any change of a policy.

4. Proof of Theories

We have designed a model namely 2D-Box Model [6],[7] for explaining a matching between packets and firewall rules (Cisco ACL) as shown in Fig 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 have not mentioned other attributes yet (such as protocol types).

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



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: 2D-Box Model (left) and ACL in a normal form (right).

As we can see, matched packets of each rule are the subset of SIP × DIP × DPT when "×" is an operator for computing the Cartesian product. The result from the Cartesian product is called Relation. For example,

Rule Number 1 (Rule-1)

a × y × any = { (a,y,1), (a,y,2) } the notation represents this relation is 'R1'

Rule Number 4 (Rule-4)

any × any × 1 = { (a,x,1), (a,y,1), (b,x,1), (b,y,1) } the notation represent to this relation is 'R4'

We can apply 2D-Box Model to the ACL rule by using a normal form. 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}.

4.1. Proof of Theory 1-3

As you can see in Fig 1 (left), all packets are subset of S when S = any × any × any. The packets that will match with Rule-1 are in $S \bigcap R_1$. Therefore, if $R \subseteq S$ then the packets ($p \in R$) that will match with Rule-1 will be in $R \bigcap R_1$.

Likewise, we can prove theory 2 as the follows

If p is a packet matched with Rule-1, we found that $p \in R_1$ …… (Prove by using 2D-Box Model)

If p is a packet matched with Rule-2, we found that $p \in R_2 - R_1$ …… (Prove by using 2D-Box Model)

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

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

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

Proof of theory 3

Suppose, there is no packet that matches with Rule-i

Thus, packet that matches with Rule-i is $p \in R_i - (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1) = \phi$ …… (b)

Thus $R_i \subseteq (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1)$ …… (c)

If we remove Rule-i, packet will fall down to Rule-i+1.

Packets that will match with Rule-i+1 is $p \in R_{i+1} - (R_i \bigcup R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1) = \phi$

$p \in R_{i+1} - (R_i \bigcup (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1)) = \phi$

From (c), thus $p \in R_{i+1} - (R_{i-1} \bigcup R_{i-2} \bigcup ... \bigcup R_1) = \phi$ …… (d)

There is no term of Ri in (d), therefore, deleting Rule-i can not causes any change of a policy.

4.2. Proof of Theory 4-7.

4.2.1 Assumptions

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

Let

R_u is Rx before swapping a position

R_v is Ry before swapping a position

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

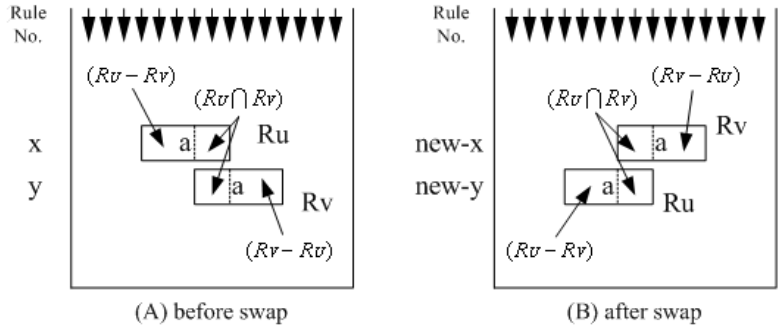


Fig. 2: Results from Difference and Intersection of Relations.

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

Before swapping (see Fig 2-A)

$p \in R_u - K$

$p \in (R_u - R_v) \bigcup (R_u \bigcap R_v) - K$ …….…….. (i)

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

$p \in R_u - R_v - K$

$p \in (R_u - R_v) - (R_u \bigcap R_v) - K$ ……..……..……. (ii)

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

Before swapping (see Fig 2-A)

$p \in R_v - R_u - K$

$p \in (R_v - R_u) - (R_u \bigcap R_v) - K$ ……...…. (iii)

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

$p \in R_v - K$

$p \in (R_v - R_u) \bigcup (R_u \bigcap R_v) - K$ ……...…. (iv)

These are the assumption that will be used for proving the theory 4-7 on the next step.

4.2.2 Proof of Theory 4

In this case, the two rules are possibly to associate any action (accept or deny, see Fig 3).

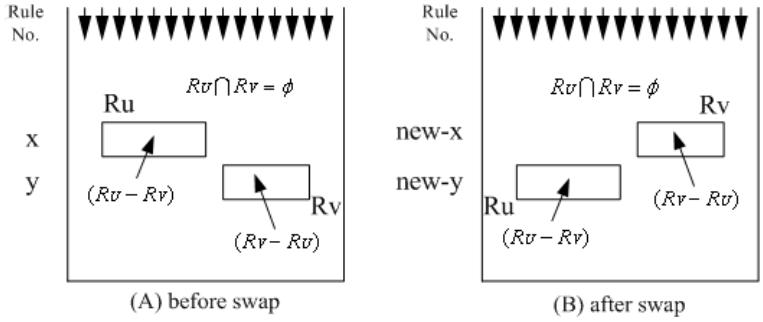


Fig. 3: Results from Difference and Intersection of Relations.

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

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

In this case, $R_u \bigcap R_v = \phi$, Thus $p \in R_u \bigcup \phi - K$

$p \in R_u - K$ …….. (a)

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

In this case, $R_u \bigcap R_v = \phi$, Thus $p \in R_u - \phi - K$

$p \in R_u - K$ …….……..…… (b)

From (a) and (b), we found that packets matched with rule-x are the same. ……...……. (c)

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

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

In this case, $R_u \bigcap R_v = \phi$, Thus $p \in R_v - \phi - K$

$p \in R_v - K$ ……..…….……..…….. (d)

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

In this case, $R_u \bigcap R_v = \phi$, Thus $p \in R_v \bigcup \phi - K$

$p \in R_v - K$ ……...…….…...…… (e)

From (d) and (e), we found that packets matched with rule-y are the same. ...……..……..……. (f)

From (c) and (f), thus theory 4 has been proved.

4.2.3 Proof of Theory 5

This case, two rules are in the same action and $R_u \bigcap R_v \neq \phi$

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

Before swapping: from (i) $p \in (R_u - R_v) \bigcup (R_u \bigcap R_v) - K$ …..……. (see Fig 2-A)

The properties of set, $(B \bigcup C) - A = (B - A) \bigcup (C - A)$

Thus $p \in ((R_u - R_v) - K) \bigcup ((R_u \bigcap R_v) - K)$ ……..…… (m)

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

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

Thus $p \in (R_u - R_v) - K$ …….……..…….(n)

From (m) and (n), we found that decreased packets (matched with Rule-x) are $p \in (R_u \bigcap R_v) - K$ …(o)

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

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

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

Thus $p \in (R_v - R_u) - K$ …….……...…….. (p)

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

The properties of set, $(B \bigcup C) - A = (B - A) \bigcup (C - A)$

Thus $p \in ((R_v - R_u) - K) \bigcup ((R_u \bigcap R_v) - K)$ ……...….. (q)

From (p) and (q) we found that increased packets (matched with Rule-y) are $p \in (R_u \bigcap R_v) - K$ …..(r)

From (o) and (r), $p \in (R_u \bigcap R_v) - K$ are the changed packets from matched with Rule-x to be matched with Rule-y, while both rules are in the same action. Therefore, there are no any changes of a policy.

4.2.4 Proof of Theory 6

Similar to the last paragraph of the proof of theory 5, from (o) and (r), $p \in (R_u \bigcap R_v) - K$ are the changed packets from matched with Rule-x to be matched with Rule-y, while both rules are in the different action. Therefore, if we swap the positions of the both rules, it's able to causes the changing of a policy.

4.2.5 Proof of Theory 7

Before removing Rule-x, packets that will fall to Rule-x are

$p \in R_x - (R_{x-1} \bigcup R_{x-2} \bigcup ... \bigcup R_1)$ ……... (s)

After removing Rule-x, these packets will fall to next rule below (Rule-y).

But $R_x \subset R_y$, Thus

$R_x - (R_{x-1} \bigcup R_{x-2} \bigcup ... \bigcup R_1) \subset R_x \subset R_y$

$R_x - (R_{x-1} \bigcup R_{x-2} \bigcup ... \bigcup R_1) \subset R_y$ ……..... (t)

From (s) and (t), Thus

$p \in R_x - (R_{x-1} \bigcup R_{x-2} \bigcup ... \bigcup R_1) \subset R_y$

After removing Rule-x, therefore, packets that used to match with Rule-x will fall down to match only with Rule-y. Due to both rules are in the same action, thus there are no any change of a policy.

5. Conclusion and Future Works

In This research, we have proposed seven theories for analyzing Cisco ACL rules. These theories can help administrators understand matching between packets and ACL rules. Moreover, we have proved the proposed theories with mathematics and our model (2D-Box model) ensure that our theories are accurate. In the future, we plan to develop and extend this research to analyzing more complex firewalls' rules such as IPTABLES that user able to define rules in layer-7 string data and Layer-4 TCP states.

6. References

[1] E. S. Al-Shaer and H. Hamed. Firewall Policy Advisor for anomaly Detection and Rule Editing. *Proc. of IEEE/IFIP Integrated Management IM'2003*. March 2003.

[2] E. S. Al-Shaer and H. Hamed. Discovery of Policy Anomaly in Distributed Firewall. *Proc. of INFOCOM 2004 - Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, Volume 4, March 2004.

[3] S. Hazelhusrt. Algorithms for Analyzing Firewall and Router Access Lists. *Technical Report TR-WitsCS-1999*, Department of Computer Science, University of the Witwatersrand, South Africa, July1999.

[4] P. Eronen and J. Zitting. An Expert System for Analyzing Firewall Rules. *Proc. of 6thNordic Workshop on Secure IT-Systems (NordSec 2001)*. November 2001.

[5] A. Silberschatz, H. F. Korth, S. Sudharsan. *Database System Concepts*. Tata McGraw-Hill, 1997.

[6] C. Pornavalai and T. Chomsiri. Firewall Policy Analyzing by Relational Algebra. *Proc. of the 2004 International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2004)*. JULY 2004.

[7] T. Chomsiri and C. Pornavalai. Firewall Rules Analysis. *Proc. of the 2006 International. Conference on Security & Management (SAM'06)*. Las Vegas. 2006.