International Journal of Advances in Engineering Research http://www.ijaer.com

(IJAER) 2013, Vol. No. 4, Issue No. I, May **ISSN: 2231-5152**

How to Optimize a Firewall

E. Gajendran[1] **Dr.** K P Yadav[2]

[1]Research Scholar, Sunrise University, Alwar, Rajasthan

[2]Director, MIET, Greater Noida(UP)

ABSTR ACT

This pape represents a general frame work for rule based firewall optimization . We give a pr ecise formulation of firewall op timization as an integer pro gramming pro blem and show that our fram ework produc es optimal reordered rule sets th at are semantically equivalent to the origi nal rule set. Our framework considers the complex intera ctions among the rules in firew all configurations and relies on a novel pa rtitioning of the packet space defined by th e rules themselves . For validation, we emplo y this framew ork on real fir ewall rule se ts for a quantitative evalua ion of existing heuristic approaches.

Key Word s- Firewall optimization, AC L optimization, ACL partiti oning**.**

Introdu ction

A firewall is a security guard placed at the point o entry bet een a private network an d the outside Internet su ch that all incoming and outgoing packet have to pa ss through it.

A packet can be viewed as a tuple with a finit number of fields such as IP address, destination number IP address, source portnumber, destination port number, and protocol type.

A general framework for evaluatin optimization techniques for rule-based firewalls fi rst divides the packet spa ce into partitions where all the packets in any given partition match the same s et of firewall rules. For each partition, the framewor calculates the cost for the firewall to p rocess all the packets in the partition b ased on traffic profile. Then, using thes partitions, the framework generates t e dependency of all the rules in the firewall.

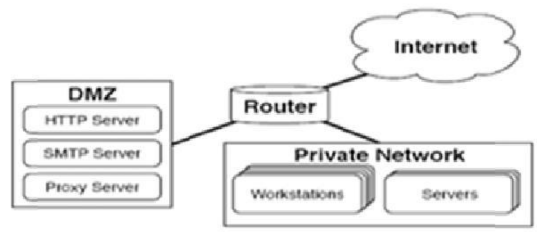
Upon receiving a pac ket, a firewall checks th packet's header against a set of user - specified rule (inspectio) and forwards/drops the packet if it i desired/undesired (filtering).

The key component of a firewall configuration is the access co trol list (AC L). An ACL consists of an ordered list of rules, each with a predicate that describes which packets are matched by this rule and the action to be taken on matched pack ets.

A packet is compared with each rule uccessively in the sequen ce until the first matching rule is found, and the action for this ru e is taken on the packet.

Let’s con sider a simple though realistic example which will explain th application of the given framework. The ro uter shown in the figure m diates all communication among the private networ k, the demilitarized zone (DMZ), and t he Internet. For this network, we assume the following required poli cy:

(R1) Communication sessions initiating fro m the Internet are onl y allowed f or http and smtp connections to the DMZ servers;



1: **allow** dnet *DMZ* dport *http*
2: **allow** dnet *DMZ* dport *mail*
3: **deny** dport *sql*
4: **allow** snet *DMZ* dnet *Internet*
5: **allow** snet *Private* dnet *DMZ*
6: **deny** all

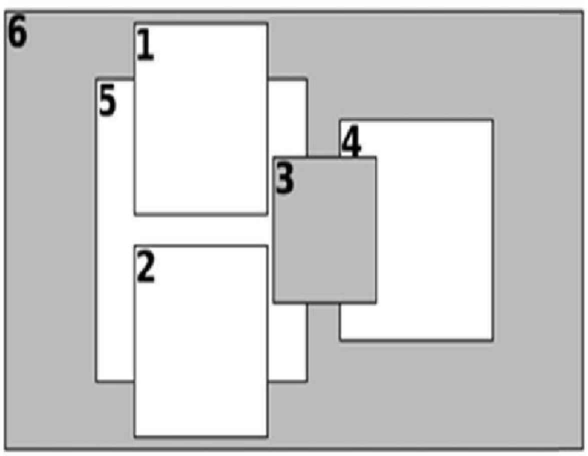
Figure 1 ACL to enforce the req ired policy for n/w

(R2) Communication sessions initiating fro m the priva te network a re not allowed to the Internet. Instead, users mus t make exter nal requests t rough the proxy server in DMZ;

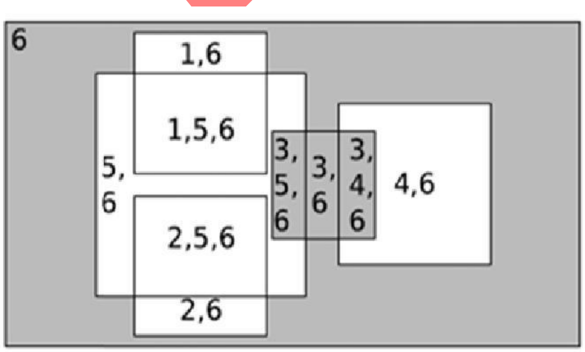
(R3) Communication sessions initiating fro m the DM an d to the private network are n ot allowed;

Due to the p ervasiveness of (R4) worms, any inter- network communication to data base is servers not allow ed. This requirement takes e priority over t first three .

Here in t his example, we have used simple ACL language. A single rule begins with th e action (allow or deny), ollowed by the predicate ag ainst which to check pac ets.



(i) Overlapping rules



(ii) The rule based partition

Fig. 1: Packet sp ce divided by ACL 1

One can replace the predicate with the keyword all, denoting that every pac ket shall be matched by thi rule.

ACL 1 e xplains how the above p olicy can b imposed. Requirement **R1** is ach ieved by the combined effect of Rule 1 (http) and Rule 2 (mail). Requireme nt **R2** is accomplished by the collective effect of Rule 5, which allows the users in the private network to initiate conne ctions to the roxy server in DMZ, and Rule 4, which allows the p roxy server to initiate connections to the Internet. Rule 3 impose **R4**, which has the utmo t priority in th e policy. Rule

6 denies all other traffic, which a lso implicitly enforces **R 3**. Rule 6 confirms ill not ACL permit

any traffic acciden tally and is t he default behaviour of most firewall pr oducts.

Optimization F ramework

Fram ework consis ts of several steps: partitioning, profiling, dependen cy generation, and optimization.

a) **Partitioning**: is used to divide the packet space into disjoint blocks accordin g to the given ACL.

b) **Profiling :** measures the weights of blocks within the partition.

c) **Dependency eneration:** ex amines the pa rtition and rules to create a set of constraints on the positions of r ules to admit only semantically equivalent rule reordering.

d) **Optimization :** step uses information from previous step to produce an integer program whose solutions yield sem antically equi valent, optimal rule reordering.

Rule-Based Partitioning O f Packet S pace

The (disjoint) blocks of the partition are created such that for any two ackets within a single bloc k, the same set of rules from the ACL 1 matches those two packets. This facil itates the correct optimization of firew all rule config urations in two key ways:

a) Since all pack ets within a bl ck will be matched by the same rule in any reord ering of the ACL, checking for correct block ac tion is sufficie nt.

b) Cost assignment can be attri uted to block rather than rul es, thus making cost calculation independent o the choice of rule ordering.

To explain rule-ba sed partitioni ng, we first p ove it on A CL 1. Figure 1(i) shows rectangles that d enote the rules of the ACL. Light rectangles denot rules that have allow act ions, while da rk rectangles denote rules that have deny actions. Notice that, rec tangle havi ng the entire figure represents Rule 6 (deny all).

Two rectangles ov erlap when the packets matc hed by

the two corresp onding rule s intersect. Rule 6 intersect with all the other rules, it mu st be positioned after all the other rules with an allow action. Rule 5 has the most inte esting relatio ships: it intersects with Rules 1, 2, 3, a nd 6. We can educe that Rules 1, 2, and 5 can be placed in any order relative to one another, while Rule 3 mu st be place d before Rules 4 and 5.

Algo rithm 1 produces a partition of the packet space wher e packets in each block have the exact same set of matchin g rules. The algorithm works by iterating over the rules in the rule sequence.

Algorithm 1: Partition the Packet Space

```
Require: |r| = n, n ≥ 0
Ensure: Γ is the rule-based partition
Γ ← P
for i ← 1 to n do
   x ← packets(r_i)
   for all γ in Γ do
      if γ ∩ x = ∅ then
            continue
      else if x ⊂ γ then
            Γ.append(γ\x)
            γ ← x
            break
      else if x ⊃ γ then
            x ← x\γ
      else
            Γ.append(γ\x)
            γ ← x ∩ γ
            x ← x\γ
      end if
   end for
   Γ.append(x)
end for
```

Block	Weight	Cost(r)	Cost(r')
{6}	0.02	0.12	0.12
{1,6}	0.05	0.05	0.20
{2,6}	0.05	0.10	0.25
{3,6}	0.02	0.06	0.02
{4,6}	0.30	1.20	0.90
{5,6}	0.10	0.50	0.20
{1,5,6}	0.20	0.20	0.40
{2,5,6}	0.20	0.40	0.40
{3,4,6}	0.03	0.09	0.03
{3,5,6}	0.03	0.09	0.03
Total	1.00	2.81	2.55

Table 1 Wei ghts for the b locks in ACL

To measure the likely time, we need some typical distrib utions, *i.e.*, probability mas functions over all pack ets in the packet space. Fo mathematical expression p acket spa ce as *P*; the ACL as *r*; the packet as *X*; and the tr affic profile a profile, a apping from packets to pro abilities.

$$E[cost(r,X)] = \sum_{x \in P} p(x)\, cost(r,x)$$

Expected ost to process a packet is the sum, over all packets i P, of the p robability f the packet multiplied by the number of rules checked for that packet. Supposing a uni t cost for all rule predicates, the expect ed cost can be stated as the f llowing:

Partition Profiling an d Rule Cost

A good metric for ACL cost is the expected time to process a single packe t. Naturally, with a lowe packet pro cessing time, the firewall can accomplish higher throughput.

(IJAER) 2013, Vol. No. 4, Issue No. I, May ISSN: 2231-5152

wher e the cost to process a pack et, cost (r, s), is the num ber of rules th at are checked against s. Assuming that ri is the first m atching rule for packet s, we have cost (r, s) = i.

Noti tha wit partitions, ce t h rule-based th first matching rule ll packets block is alik e. for in This allow to re write the s us ex ected cost s the following:

$$E[cost(r, X)] = \sum_{\gamma \in \Gamma} \left(c_{r,\gamma} \times \sum_{s \in \gamma} profile(s) \right)$$

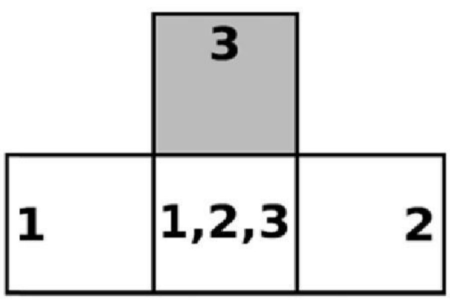


Figure 3An A CL demonst rating complex dependencies

Afte r continuing this factorizati on, we can su m the probabilities of a ll packets w ithin the blo ck to produce a single weigh t. Notice that these weight factors depend only on the traffic profile and ar independe nt of any rule rder. This lea ves us with:

$$E[cost(r,X)] = \sum_{\gamma \in \Gamma} (c_{r,\gamma} \times weight_{\gamma})$$

It is compulsory for t he traffic pro filing tool to dynamical ly monitor the traffic and update the traffic profile as needed. Assu me that the administrato selects a distribution n of the packet t pace such tha t the weights are determin ed to be as shown in Table I.

Assume th is the initial ACL. cost for
at The each
block is own in the third column of the
s table. Th
colu differe permutation of
final mn shows a nt th
AC wher reordered
L, , e the rules are a
orderin retains the
. This g same
meaning ut has a lower
, cost.

Depende ncy Gener ation

Dependen cies between the relative po sitions of two rules is that overla , but with dif ferent actions. For examp le, we noted in ACL 1 that Rules 3 and 5 intersect and have diffe ent actions, so Rule 3 must be placed before Rule 5 in any valid re ordering.

A dependency must n t be betwee n a rule pair. Instead, it should be bet ween a rule i and a block’ matching ules that have a different a tion from that of rule i. We denote th ese dependen cies using th following format: i ⊏{j,k,...,l}. Such a dependency requires that rule i must f ollow the earliest rule o f {j,k,...,l}.

Algorithm 2 shows how these constraints can b generated. It generates a dependency wheneve r there is a lock with conflicting rules (lines 2–4). Fo r these conflicting blocks, a dependency is generated for each rule with a different action from the first matching rule (lines 3–4 . The constrai nt requires that in any reo rdered ACL, the rule must be positioned after at least one of the matching rules with the same action (lin e 5).

For ACL 1 , the depende cies are given by:

$$D = \begin{Bmatrix} 4 \sqsupset \{3,6\}, 5 \sqsupset \{3,6\}, 6 \sqsupset \{1\}, 6 \sqsupset \{2\}, \\ 6 \sqsupset \{4\}, 6 \sqsupset \{5\}, 6 \sqsupset \{1,5\}, 6 \sqsupset \{2,5\} \end{Bmatrix}$$

The permu tations that satisfy the dependencies D include:

$$\langle 1,2,3,4,5,6\rangle \langle 2,1,3,4,5,6\rangle \langle 1,3,2,4,5,6\rangle$$
$$\langle 2,3,1,4,5,6\rangle \langle 3,1,2,4,5,6\rangle \langle 3,2,1,4,5,6\rangle$$
$$\langle 1,2,3,5,4,6\rangle \langle 2,1,3,5,4,6\rangle \langle 1,3,2,5,4,6\rangle$$
[illegible]

Afte r checking the cost of all permutations given the weights listed in Table I, the permutatio

is found to be an opti mal solution.

Con clusion

A general frame ork for eval uating optim zation tech iques for rul e-based firew lls first divides the packet space into p artitions where all the pac kets in any given partitio n match the s ame set of firewall rules. For each part ition, the fram ework calculates the cost for the firewall to process a ll the packets in the partition based on traffic profil . Then, using these partitions, the fra ework gener ates the depen dency of al l the rules in th e firewall.

It i worth underlining that the methods and algor ithms present ed in this paper are not restri cted to the d esign and an alysis of firew all policies. ather, they can be appli d to other rule based syste ms as well.

Ref erences

[1] Ghassan Misherghi, Lihua Yuan, Zhendo ng Su, Chen-Nee Chuah, and Hao Chen, "A G eneral Framework for Benchmarking Firewall Optimization Techniques'', IEEE TRANSACTIONS ON NETWORK AND SERVICE M ANAGEMEN T, VOL. 5, O. 4, DECEMBER 2008.

[2] Alex X. Liu, Mohamed G . Gouda, "Firewall Policy Queries'', IEEE TRANSACTIONS ON PARALLEL AND DISTRI BUTED SYS EMS, VOL. 20, NO. 6, JUNE 2009 .

[3] Alex X. Liu, Mohamed G . Gouda, "D iverse Firewall Design", IEEE RANSACTIONS ON PARALLEL AND DISTRI BUTED SYS EMS, VOL. 19, NO. 9, SEPTEMBER 2008

[4] Ehab Al-Shaer, Hazem Ham ed, Raouf Bo utaba, and Masum H asan, "Confli ct Classificati n and Analysis of D istributed Firewall Policies", IEEE JOURNAL ON SELEC TED AREA S IN COMMUNIC ATIONS, V OL. 23, NO . 10, OCTOBER 20 05.

[5] Ehab S. Al-Shaer an d Hazem H. H amed, "Discovery of Policy Anom alies in Distributed Firewalls", 0-7803-8355- 32004

9/04.CF3G Redundanc emoval for P acket Classif
IEEE. y iers in
Alex X. Liu, Mohamed G. Gouda, "Co
[6] mplete

TCAMs", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 21, NO. 4, APRIL 2010

[7] **Y**eim-Kuan Chang, "Efficient Multidimensional Packet Classification with Fast Updates", IEEE TRANSACTIONS ON COMPUTERS, VOL. 58, NO. 4, APRIL 2009.