Abnormal Policy Detection and Correction Using Overlapping Transition

Sunghyun Kim and Heejo Lee [?]

Korea University, Seoul 136-713, South KOREA
fafshkim, heejog@korea.ac.kr

Ordered Binary Decision Diagram(OBDD) [7] to present and to manipulate the policy expressions, for automatic discovery of security policy conflicts among IPsec as well as firewalls.

Another useful application for security policy, Firewall Policy Adviser(FPA), was proposed by Al-Shaer et. al [8–10]. They presented a set of techniques and algorithms that provide automatic discovery of firewall policy anomalies to reveal rule conflicts and potential problems in legacy firewalls, and anomaly free policy editing for rule insertion, removal, and modification. FPA constructs a policy tree for firewall rules and state diagram for anomaly discovery for rules. Traversing them, FPA searches the misconfigured rules.

3.1 Intra-policy anomaly

Intra-policy anomalies occur among rules in a single security device. Table 1 shows all anomalies in a single firewall. Overlaps among rules make abnormal relations. That is, if there was no overlap among rules, anomalies cannot occur except for irrelevance that

4 Resolving Policy Anomaly and Overlaps

We propose a new method to solve the policy anomaly based on set theory. As de-

[redacted]

Algorithm 1 CreateIntraRPA(f_i, R_{out})

```
Require: $R_{out} = f r_1; r_2; ::: r_n g$
Ensure: $RPA_{f_i}$
  Extract $V_{f_i} = f v^i_{f_1}; v^j_{f_2}; :::v^k_{f_m} g$ from $R_{out}$
  $k \tilde{A} jV_{f_i}j$
  Create $(2k + 1)$'s size of $RPA$ for $f_i$
  $v^0_{f_i} ( 0$
  for $j = 0$ to $k$ do
      if $v^j_{f_i}$ 6= $v^{j\ i\ 1}_{f_i} + 1$ then
          $RPA_{f_i}[2][2[2$
```

Table 5: Final rules without anomaly.

ID	SIP	SP	DIP	DP	Act.
r_1	1.1.1.100	[1-1024]	2.1.1.[1-3]	*	D
r_2	1.1.1.100	[1-1024]	2.1.1.[5-255]	*	D
r_3	1.1.1.[1-255]	*	2.1.1.[1-255]	*	A

4.3 Inter-policy Detection and Correction

Anomaly detection and correction for multiple devices is a little more complicated than for a single device. As you can see in Table 2, we have to consider not only rules'

Fig. 3: Example of firewall deployment

sliced into a minimum overlap in RPAs as Eq. (3) describes. However, since RPAs are created by two rule sets, R_{in} and W_{out}, RPAs for F_{z_i} have two domain bitmaps in each

Algorithm 3 CreateInterRPA(R_{in}, W_{out})

```
Require: $R_{in} = fr_1; r_2; ::r$
```

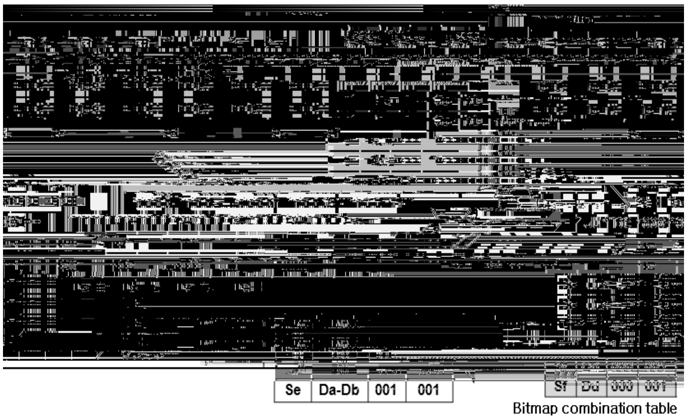

Fig. 4: Detection and correction of abnormal rules among one zone firewall and the other firewalls. (Colored rows have anomaly.)

Table 7: The number of rules and distinct values of protocol fields in two ACL files

ID	

have only a constant IP address and port number, the combined result has only one. But if a predicate of a rule has wide IP addresses or port numbers like “any”, which covers all the domain regions of the corresponding protocol field, the combined results are generated as many as the number of split domain regions. “ACL1” includes 194 rules having one “any” predicate, 75 rules having two “any” predicates, and 6 rules having three “any” predicates. “ACL2” includes 351 rules having one “any” predicate, 35 rules having two “any” predicates. As a result, “ACL1”, which has 40% less rules than “ACL2”, needed about 3 times process time because of overlaps.

6 Conclusions

Security policy has a critical role for network protection. Policy maintenance is a com-

3.