A Novel and Feasible System for Rule Anomaly and Behavior Mismatching Diagnosis Among Firewalls

Chi-Shih Chao

Abstract While configuring firewalls, firewall rule ordering and distribution must be done cautiously on each of cooperative firewalls, especially in a large-scale network. However, network operators are prone to incorrectly configuring firewalls because there are typically hundreds of thousands of filtering rules (i.e., rules in the access control list file, or ACL for short) which could be set up in a firewall, not to mention these rules among firewalls could affect mutually. To speed up the crucial but laboring inspection of rule configuration on firewalls, this chapter describes our developed diagnosis system which can not only figure out anomalies among firewall rules effectively but also infer/correlate the main reasons from the diagnosed anomalies for filtering (behavior) mismatching between firewalls. At the end of this chapter, the system prototype is shown as a demonstration of our system implementation.

Keywords Firewall rule anomaly • Firewall behavior mismatching • Anomaly correlation • Diagnosis results reuse

1 Introduction

In the Internet, firewalls and their associated filtering rules should be discreetly deployed and configured for cooperative, integrated, and in-depth network security protection. Yet, in a large and complex enterprise network equipped with numbers of firewalls, it is very possible for a network manager to make mistakes while setting the firewall rules (i.e., ACL rules) since maintaining the security consistency between firewalls' rule configuration and the demands of network security policies is always time-consuming, laboring, and error-prone. Sometimes, the matter can go

C.-S. Chao (✉)
Department of Communications Engineering, Feng Chia University, Taichung 40724, Taiwan
e-mail: cschao@fcu.edu.tw

J. Juang and Y.-C. Huang (eds.), *Intelligent Technologies and Engineering Systems*,
Lecture Notes in Electrical Engineering 234, DOI 10.1007/978-1-4614-6747-2_7,

even worse when several managers with different levels of professional knowledge are assigned to do this job collectively.

The security inconsistency typically can be revealed by either the occurrence of anomalies between the firewall rules or demand mismatching of network security policies [1]. E. Al-Shaer and H. Hamed formally define an anomaly as a duplicate or multiple rule matching for a packet in a rule set. Based on the concept, they further define several different intra-/inter-ACL anomalies among the firewall rules [2–5]. However, because a finite-state-machine (or FSM)-based comparison between each pair of rules should be conducted for anomaly checking, their anomaly diagnosis will meet an inefficiency when the number of rules or firewalls get large.

To lower the comparison times between firewall rules needed in [4], Y. Yin et al. [6, 7] segment the IP address space formed by the source and destination networks into blocks where each block is precisely cut out by the IP addresses in the conditional field of each firewall rule. Utilizing these varying-sized blocks, a SIERRA tree is built and two conflict rules would be hanged on the same branch [8]. The network manager (or system) just needs to do the anomaly inspections/checking on rules in the same spatial block(s), instead of wasting enormous time to conduct a comprehensive pair-wise rule comparisons. Yet, this approach would lead to a fatal drawback in a networking environment with frequent rule updates. A clean-slate reconstruction of the SIERRA tree is very possibly unavoidable if a simple rule deletion or insertion is performed. It is because space blocks are exactly sliced according to the IP addresses of each rule. So, once one rule changes, a change for the whole spatial rule relationship would occur, and the corresponding data structures could be reconstructed. This drawback also means the local diagnosis results, that is, the intra-ACL rule diagnosis results, can hardly be reutilized for the diagnosis of inter-ACL rule anomalies.

As to the demand mismatching of network security policies for the security inconsistency, many research results can be found for the past several years. Among them, some are noteworthy. Chao (this chapter's author) [9] first categorizes the filtering behavior (or effect) mismatchings between two firewalls into two different types: incorrectly blocking error and incorrectly admitting error. A formal high-level specification language is also designed to let managers describe the demand of the security policies [10]. The developed system can compare the high-level specification description file with the ACL file of any specific firewall for behavior mismatching check. Still, due to lack of sound linguistic validation for its high-level specification, by far, the system is not ready on the stage of practicability. In 2008, Alex Liu et al. [11, 12] propose firewall decision diagram (FDD) data structure as well as a similar high-level specification language called property rule. With their associated algorithm, the ACL file of filtering rules and the high-level specification file representing the demand of network security policies would be transformed to their FDD counterparts separately and then find the differences of their filtering effects. However, their implementations are viewed as far from being practicable either due to the lack of proof of solidness and completeness of the high-level logics they use to specify the firewall behavior.

The rest of our chapter is organized as follows: Sect. 2 describes the novel data structure we use to achieve feasible and efficient rule anomaly diagnosis. In Sect. 3, we illustrate how the diagnosed rule anomalies are correlated and how we filter out the anomalies which actually lead to mismatching filtering effects among firewalls. Lastly, Sect. 4 gives a brief summary with a glance at our future work.

2 Firewall Rule Anomaly Diagnosis

For the anomalies between firewall rules, they are defined completely by E. Al-Shaer et al. and classified broadly into two types: anomalies within one single ACL (or called intra-ACL rule anomalies) and anomalies among different ACLs (or called inter-ACL rule anomalies). In this section, our RAR tree-based diagnosis approach is introduced and we will show how it can facilitate the diagnosis of firewall rule anomalies. To avoid the typical time-consuming pair-wise rule comparisons for anomalies checking [3], a 2-dimensional address space matrix is designed as a structural basis of our RAR tree to prune out those unnecessary comparisons in which there is no intersection between the IP address spaces of two rules. To do so, in our system, the IP address ranges of the source network domain and destination network domain are employed as two axes to form a rectangle plane which is further divided into a matrix containing fixed-sized blocks. Later, with the fields of <source_IP> and <destination_IP>, the IP address space of each ACL filtering rule can be represented as a smaller rectangle and drawn on some proper place of this matrix (see Fig. 1).

After that, the address space of a rule can be recorded in our RAR tree in the form of, □—○—△ where □ contains the values of the conditional fields of the rule, ○ is used to indicate the matrix block(s) spanned by the address space of the rule, △ and shows the label (or the order) of the rule. By dealing with each rule in this fashion, the RAR tree depicting the structural configuration of Fig. 1 can be built as Fig. 2. From Fig. 2, it can be found that there are six branches containing more than one △ leaves, which indicates only the IP address spaces of those rules in these branches could have the chance to intersect with each other and hence incur intra-ACL rule anomalies. So, we simply have to do the pair-wise rule comparisons for anomaly checking on the rules at the same branch within these six branches. Comparing to [2], if the RAR tree is not introduced, then around three times rule pair-wise comparisons are required for anomaly checking.

To isolate the inter-ACL (or even inter-firewall) rule anomalies, in our approach, it can easily be achieved by simply reutilizing the RAR trees built for the diagnosis of intra-ACL (or intra-firewall) rule anomalies. As network managers often do, for instance, we can first do the intra-ACL anomaly diagnosis for rules inside two designated firewalls individually, which will lead to the construction of two RAR trees separately for the diagnosis of intra-ACL rule anomalies. Later, to obtain the diagnosis of inter-firewall rule anomalies between these two firewalls, a tree integration can be made by simply collecting the leave △ nodes belonging to the

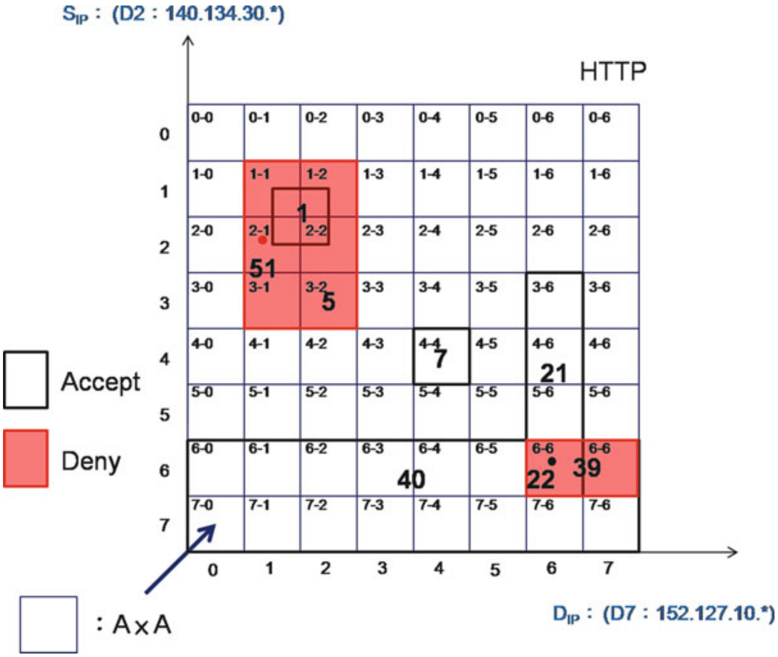


Fig. 1 2-dimensional address space matrix

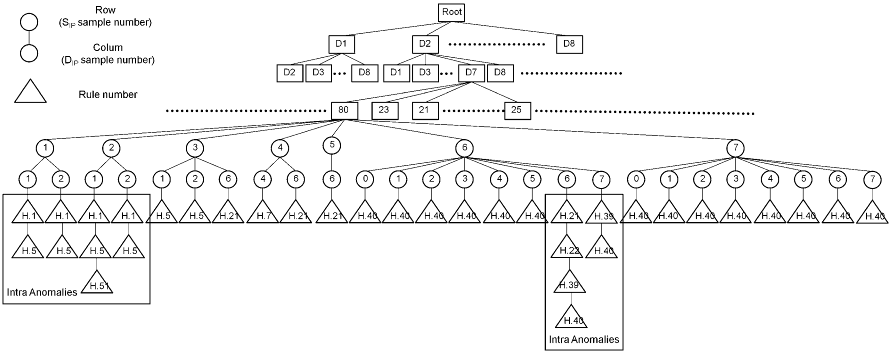


Fig. 2 The RAR tree of Fig. 3

same branch of the two individual RAR trees and putting them together under the same branch of a new RAR tree for inter-ACL rule anomaly diagnosis. And then, following the same logic (fashion) in our diagnosis for intra-ACL rule anomalies, the pair-wise comparisons for the diagnosis of inter-ACL rule anomalies would only be conducted for those rules which are under the same branch of the integrated RAR tree for inter-ACL rule anomaly diagnosis.

Two dominating advantages can be obtained by the introduction of our RAR tree:

1. Unlike the approaches in [6, 8], the local diagnosis results can be fully and easily reused. As described, the RAR trees built for intra-ACL rule anomaly diagnosis can be easily integrated for the use of the inter-ACL rule anomaly diagnosis. Comparing with the direct pair-wise-based solution in [2], our approach makes a huge saving of about 84 % time-consuming comparisons for inter-ACL rule anomaly checking.
2. By simple integration of RAR trees for intra-ACL rule anomaly diagnosis, our system can handle with ease the diagnosis of the inter-ACL rule anomalies among a large number of firewalls in an enterprise-level network, that is, our RAR tree-based diagnosis has superior capability of being up against network expansion. In fact, it can also be observed that it is quite easy for our approach to deal with the situation of network or firewall dynamic (e.g., a rule insertion or deletion). Other existing "clean-slate" approaches would do far more efforts on the rebuilding of data structures for the inter-ACL rule anomaly diagnosis. As a consequence, low system expansibility and scalability is incurred.

3 Behavior Mismatching Diagnosis

Besides anomalies among firewall rules, the other most noticeable problem with the security inconsistency while setting firewalls is the difference of filtering effects, say behavior mismatching, between two firewalls. In SOC, network managers often want to know if two of their equipped firewalls have the same filtering effect for unified in-depth protection [9]. To achieve the objective, a 3-dimensional Service Flow Space is devised, which is formed on the basis of the fields <order>, <Source IP>, <Destination IP>, and <Action> of each of filtering rules within the designated ACL of a firewall. Figure 3 shows two examples of 3-dimensional Service Flow Space for firewalls A and B. And their corresponding 2-dimensional counterparts can be drawn like those in Fig. 4, which reveal their actual filtering effects (or behavior), separately. As an example, in Fig. 4, we can indicate two of conflicting filtering regions: one marked as M_1 in (S_{IP}, D_{IP}) = (192.168.0.0/27, 192.168.1.64/26) with incorrect blocking error and the other one M_2 located in (S_{IP}, D_{IP}) = (192.168.0.128/27, 192.168.1.64/26) with incorrect admitting error, where firewall A is set as being upstream to firewall B.

To reason these two problematic regions, one thing should be highlighted first. According to the first matching scheme of firewalls [1], the actual filtering effect of an ACL can be formed by those rules which are not fully covered by any other firewall rules in the 3-dimensional Service Flow Space. Those rules in our work are termed as "significantly effective rules." Once the problematic regions fall in (or interest with) the IP addresses of the effective rules, it can be found that some

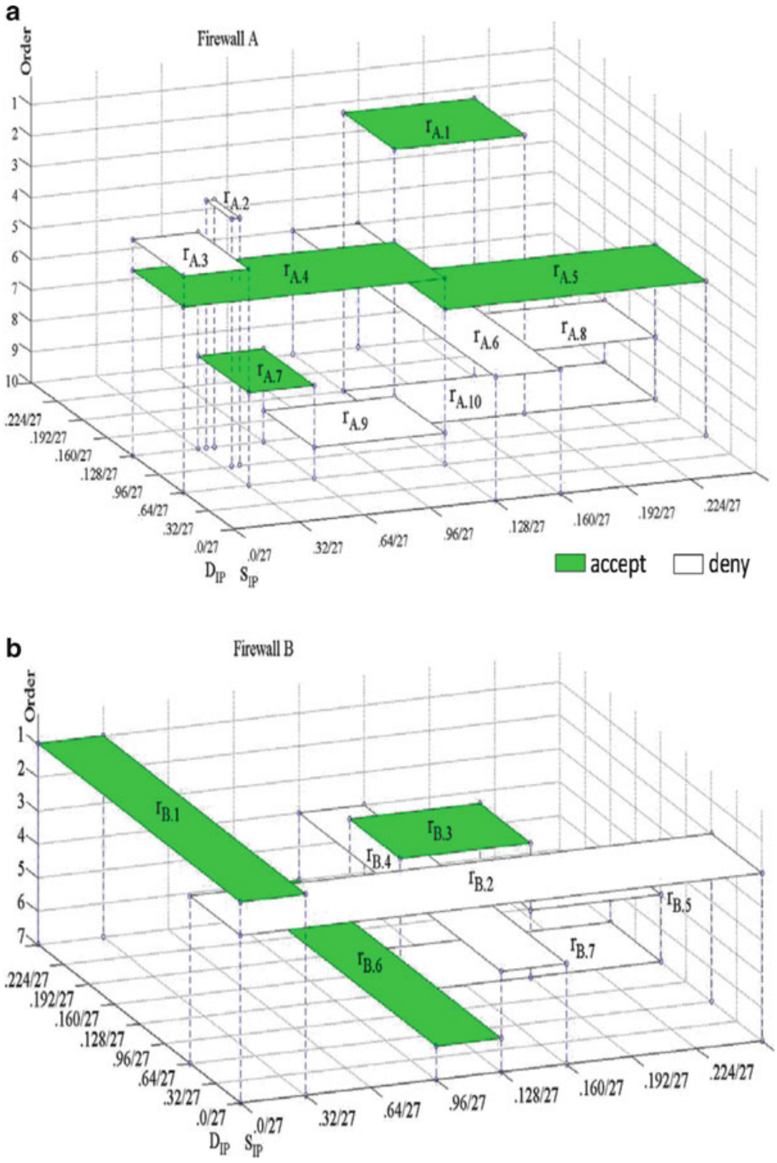


Fig. 3 3-dimensional service flow space

specific inter-ACL/inter-firewall anomalies will accompany. For example, for the region M_1 in Fig. 4, the filtering action of firewall A is to deny the traffic, but it is to accept the traffic for firewall B while the rule $r_{A.3}$ (the third rule of firewall A) and the rule $r_{B.1}$ (the first rule of firewall B) make an inter-ACL shadowing anomaly. Still, for rule $r_{A.4}$, although its IP address space also falls in M_1, its filtering effect on

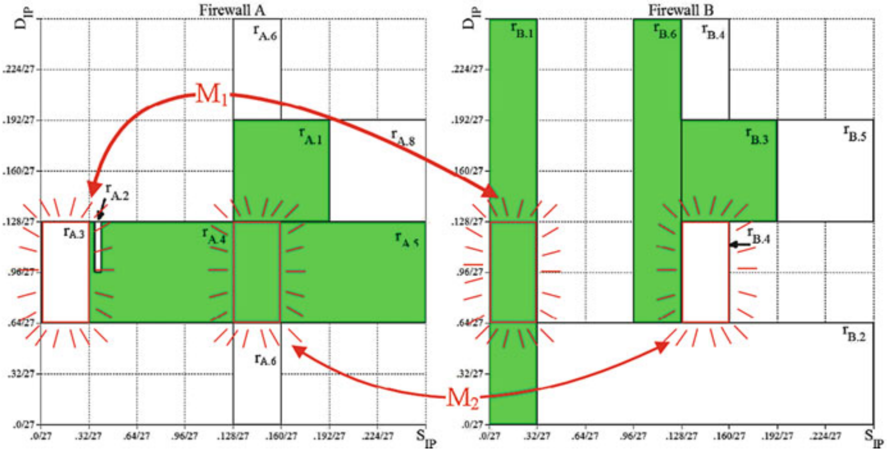
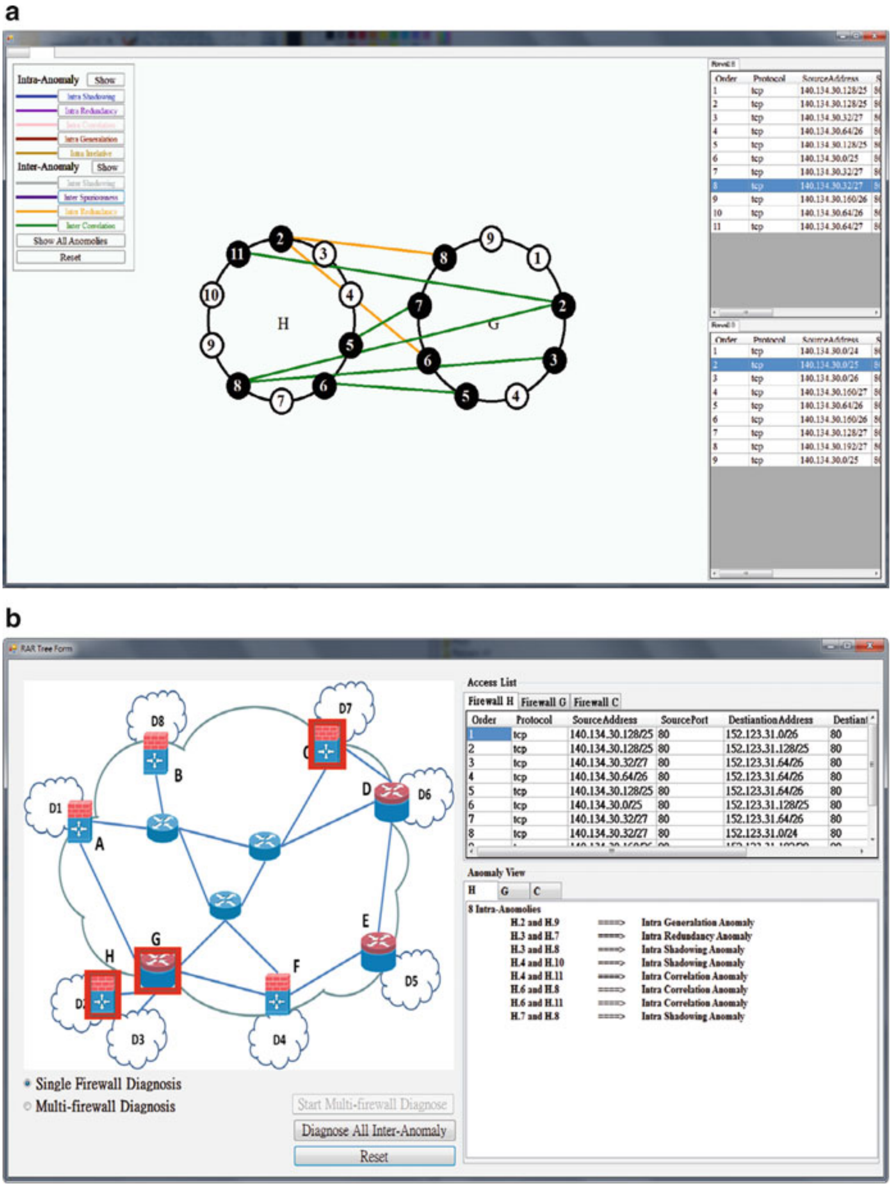


Fig. 4 Two conflicting filtering regions

M_1 can be neglected due to its lower priority in firewall A. It also means that network managers can ignore the inter-ACL generalization anomaly generated by $r_{A.4}$ and $r_{A.3}$. The same holds true for the case of region M_2. In the same fashion, rules $r_{A.5}$ and $r_{B.4}$ are the significantly effective rules for these two firewalls and also built the region M_2 with an inter-ACL correlation anomaly in which the <action> of $r_{A.5}$ is accept but $r_{B.4}$ is deny. In spite of the occurrence of another inter-ACL redundancy anomaly between $r_{A.6}$ and $r_{B.4}$ for region M_2, $r_{A.6}$ has a lower order than $r_{A.5}$ and will be not affect the M_2 at all. Thus, its effect is overlooked. Thus, in the above manner, we can effectively filter out the rule anomalies truly causing firewall behavior mismatching.

4 System Implementation and Future Work

Figure 5 shows the rule anomalies inferred by our developed diagnosis system with a well-designed GUI while those rule anomalies causing behavior mismatching are listed and indicated on Fig. 5b. In the near future, a 3D behavior mismatching diagnosis system will be developed and network managers can use it to make recommendations directly on the 3D visualized GUI by simply changing the shape of the problematic objects, regions, or icons. Additionally, we will also go deeper in exploration on the filtering effect relationships among multiple firewalls to let our developed system be more suited for the real challenges.



Order	Protocol	SourceAddress
1	tcp	140.134.30.128/25
2	tcp	140.134.30.128/25
3	tcp	140.134.30.32/27
4	tcp	140.134.30.64/26
5	tcp	140.134.30.128/25
6	tcp	140.134.30.0/25
7	tcp	140.134.30.32/27
8	tcp	140.134.30.32/27
9	tcp	140.134.30.160/26
10	tcp	140.134.30.64/26
11	tcp	140.134.30.64/27

Order	Protocol	SourceAddress
1	tcp	140.134.30.0/24
2	tcp	140.134.30.0/25
3	tcp	140.134.30.0/26
4	tcp	140.134.30.160/27
5	tcp	140.134.30.64/26
6	tcp	140.134.30.160/26
7	tcp	140.134.30.128/27
8	tcp	140.134.30.192/27
9	tcp	140.134.30.0/25

Access List

Firewall H | Firewall G | Firewall C

Order	Protocol	SourceAddress	SourcePort	DestiantionAddress	Destiant
1	tcp	140.134.30.128/25	80	152.123.31.0/26	80
2	tcp	140.134.30.128/25	80	152.123.31.128/25	80
3	tcp	140.134.30.32/27	80	152.123.31.64/26	80
4	tcp	140.134.30.64/26	80	152.123.31.64/26	80
5	tcp	140.134.30.128/25	80	152.123.31.64/26	80
6	tcp	140.134.30.0/25	80	152.123.31.128/25	80
7	tcp	140.134.30.32/27	80	152.123.31.64/26	80
8	tcp	140.134.30.32/27	80	152.123.31.0/24	80

Anomaly View

H | G | C

8 Intra-Anomolies

H.2 and H.9	====>	Intra Generalation Anomaly
H.3 and H.7	====>	Intra Redundancy Anomaly
H.3 and H.8	====>	Intra Shadowing Anomaly
H.4 and H.10	====>	Intra Shadowing Anomaly
H.4 and H.11	====>	Intra Correlation Anomaly
H.6 and H.8	====>	Intra Correlation Anomaly
H.6 and H.11	====>	Intra Correlation Anomaly
H.7 and H.8	====>	Intra Shadowing Anomaly

Fig. 5 Diagnosis results for (**a**) rule anomalies and (**b**) behavior mismatching

References

1. Hari B, Suri S, Parulkar G (2000) Detecting and resolving packet filter conflicts. Proc IEEE INFOCOM 3:1203–1212
2. Al-Shaer E, Hamed H (2004) Discovery of policy anomalies in distributed firewalls. In: Proceedings of the 23rd annual joint conference of the IEEE Computer and Communications Societies, vol 4, pp 2605–2616
3. Al-Shaer E, Hamed H (2003) Firewall policy advisor for anomaly discovery and rule editing, In: Proceedings of the 8th international symposium on integrated network management, pp 17–30
4. Al-Shaer E, Hamed H, Boutaba R, Hasan M (2005) Conflict classification and analysis of distributed firewall policies. IEEE J Selected Areas Commun 23(10):2069–2084
5. Al-Shaer E (2004) Managing firewall and network-edge security policies. In: Proceedings of network operations and management symposium, vol 1, pp 926–932
6. Yin Y, Katayama Y, Takahashi N (2008) Detection of conflicts caused by a combinations of filters based on spatial relationships. J Inf Process Soc Jpn 49:3121–3135
7. Thanasegaran S, Yin Y, Tateiwa Y, Katayama Y, Takahashi N (2009) Topological approach to detect conflicts in firewall policies. In: International workshop on security in systems and networks, proceedings of the of 23rd IEEE international parallel and distributed processing symposium, SSN-1569173665-paper-3.pdf
8. Yin Y, Bhuvaneswaran RS, Katayama Y, Takahashi N (2005) Implementation of packet filter configurations anomaly detection system with SIERRA. In: International conference on information, communication and signal processing, *LNC*S 3783, pp 467–480
9. Chao CS, Liu AC (2006) An internet firewall policy verification system. In: Proceedings of the 9th Asia-Pacific network operations and management symposium, Poster session 1, No. 4, Sept 2006
10. Chao CS (2007) An internet firewall policy validation system. In: Proceedings of the 10th Asia-Pacific network operations and management symposium, Oct 2007, pp 364–374
11. Liu A, Gouda MG (2008) Diverse firewall design. IEEE Trans Parallel Distrib Syst 19(9):1237–1251
12. Liu A (2009) Firewall policy verification and troubleshooting. Comput Netw 53(16):2800–2809