Proc. of the Seventh International Conference On Advances in Computing, Electronics and Electrical Technology - CEET 2017.
Copyright © Institute of Research Engineers and Doctors. All rights reserved.
ISBN: 978-1-63248-126-9 doi: 10.15224/ 978-1-63248-126-9-31

Tree-Based Diagnosis Mechanisms for Rule Anomalies among Internet Firewalls

Chi-Shih Chao

Abstract—**While configuring firewalls, firewall rule ordering and distribution must be done cautiously on each of cooperative firewalls. However, network operators are prone to incorrectly configuring firewalls because there are commonly 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 mention these rules among firewalls could affect mutually. To speed up the crucial but laboring inspection of rule configuration on firewalls, this paper describes our developed diagnosis mechanisms which can speedily figure out rule anomalies within/among firewalls with two innovative data structure – Adaptive Rule Anomaly Relationship tree (or ARAR tree) and Fixed-Stride Trie (FST), respectively. With the aid of these data structures and associated algorithms, significant improvements in this field have been made.**

Keywords—defense in depth, firewall rule anomalies, ARAR tree, FST, diagnosis result re-use.

I. Introduction

Network firewalls and their associated filtering rules should be properly deployed and configured for cooperative, integrated, and in-depth network security protection. However, in a large and complex network equipped with numbers of firewalls, it is very likely for a network manager to get trouble while setting the firewall rules (i.e., ACL rules) since maintaining the security consistency between firewall rule configuration and the demands of network security policies is always time-consuming, laboring, and error-prone. This is why some researcher [1] compared this configuring task to programming a distributed system in assembly language. The security inconsistency typically can be revealed by either the occurrence of anomalies between the firewall rules or demand-mismatching of network security policies. E. Al-Shaer et al. 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 [2-4]. Nevertheless, 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 increases.

To lower the comparison times between firewall rules needed in [3, 4], Y. Yin et al. [5] 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 of the tree. The network manager just needs to do the anomaly inspections/checking on rules in the same spatial block(s), instead of wasting time to conduct a comprehensive pair-wise rule comparisons like [2-4]. Yet, this approach would lead to a fatal drawback in a networking environment with the need of frequent rule updates. A clean-slate reconstruction of the SIERRA tree is very possibly unavoidable when a rule deletion or insertion is performed. 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, i.e., the intra-ACL rule diagnosis results, can hardly be re-used for the diagnosis of inter-ACL rule anomalies. By the same token, it is very likely that the modification or reconfiguration of firewall rules for new demands of network security could fail to go live in time for the system in the face of different threats. The rest of the paper is organized as following: In Section 2 and 3, the data structures we used – ARAR tree and FST with their corresponding operations are shown. Their diagnosis mechanisms for intra-ACL and inter-ACL rule anomalies are introduced also. Section 4 presents our performance evaluations and Section 5 concludes this paper and shows some of future trends of our system development.

II. Diagnosis with ARAR Tree

A. *ARAR Tree*

To have a clear overlook of our mechanism, Fig. 1 network is used and built in our lab where Fig. 2 shows those filtering rules with port 80 which are configured in firewall H, G, and C, respectively, for the routing path from network domain D2 to domain D**7** (the dotted line in Fig. 1)**.** In our work, the IP address ranges of the source network domain and destination network domain of a designated routing path are employed as two axes to form a rectangle traffic plane; later, with the fields of <source_IP> and <destination_IP>, the IP address space of each ACL filtering rule can be depicted as a smaller rectangle and put on some proper place of this traffic plane (see Fig. 3).

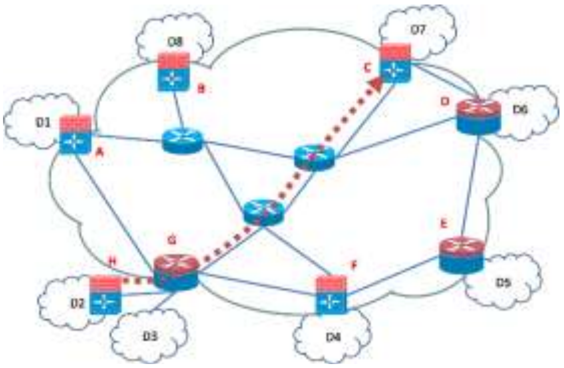
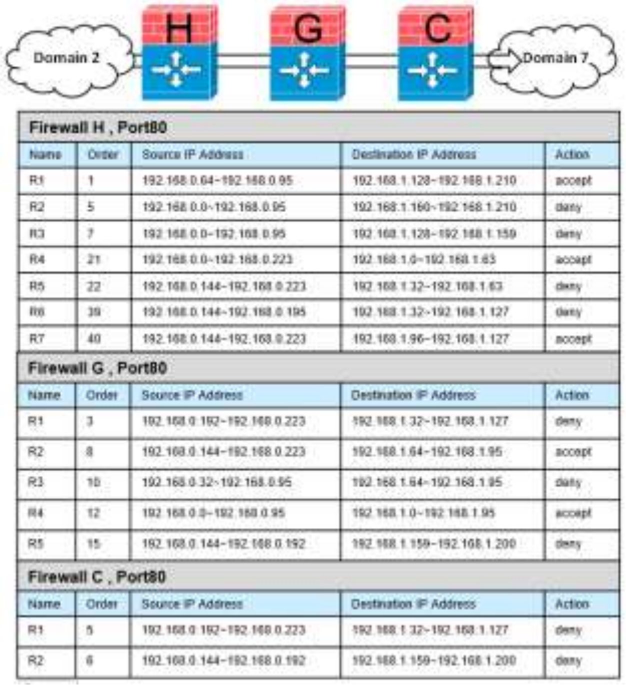


Figure 1. Example network

Chi-Shih Chao
Feng Chia University, Taiwan, ROC


H
G
C
Domain 2
Domain 7

Firewall H , Port80

Name	Order	Source IP Address	Destination IP Address	Action
R1	1	192.168.0.64~192.168.0.95	192.168.1.128~192.168.1.210	accept
R2	5	192.168.0.0~192.168.0.95	192.168.1.160~192.168.1.210	deny
R3	7	192.168.0.0~192.168.0.95	192.168.1.128~192.168.1.159	deny
R4	21	192.168.0.0~192.168.0.223	192.168.1.0~192.168.1.63	accept
R5	22	192.168.0.144~192.168.0.223	192.168.1.32~192.168.1.63	deny
R6	39	192.168.0.144~192.168.0.195	192.168.1.32~192.168.1.127	deny
R7	40	192.168.0.144~192.168.0.223	192.168.1.96~192.168.1.127	accept

Firewall G , Port80

Name	Order	Source IP Address	Destination IP Address	Action
R1	3	192.168.0.192~192.168.0.223	192.168.1.32~192.168.1.127	deny
R2	8	192.168.0.144~192.168.0.223	192.168.1.64~192.168.1.95	accept
R3	10	192.168.0.32~192.168.0.95	192.168.1.64~192.168.1.95	deny
R4	12	192.168.0.0~192.168.0.95	192.168.1.0~192.168.1.95	accept
R5	15	192.168.0.144~192.168.0.192	192.168.1.159~192.168.1.200	deny

Firewall C , Port80

Name	Order	Source IP Address	Destination IP Address	Action
R1	5	192.168.0.192~192.168.0.223	192.168.1.32~192.168.1.127	deny
R2	6	192.168.0.144~192.168.0.192	192.168.1.159~192.168.1.200	deny

Figure 2. Filtering rules with port 80 for the routing path from domain D2 to domain D7

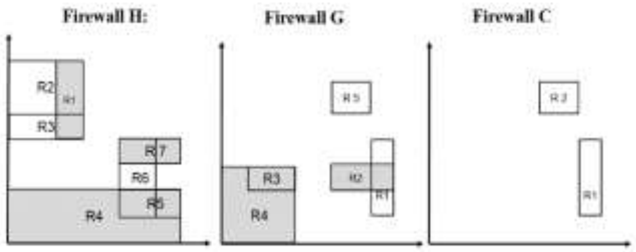


Figure 3. The IP address space for each of related filtering rules in firewalls H, G, and C

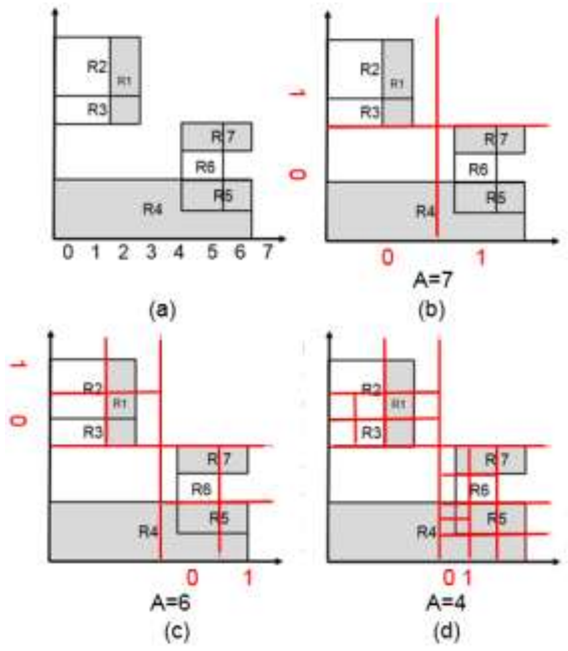


Figure 4. Exponential splitting on a traffic plane

Referring to the coding tree data structure widely used in data/video compression [6], the traffic plane will be split recursively as well as exponentially if a split block finds there are more than two rules within it (Fig. 4), rather than splitting the traffic plane into a matrix consisting of fixed-sized smaller blocks as done in our previous work [7]. After that, the address space of a filtering rule can be recorded in our ARAR tree in the form of □—○—△, where □ contains the values of the conditional fields of the rule, ○ is used to indicate the split block(s) spanned by the address space of the rule, △ shows the label (or the order) of the rule. By dealing with each rule in this fashion, the ARAR tree depicting the structural configuration of Fig. 4 can be shown as Fig. 5.

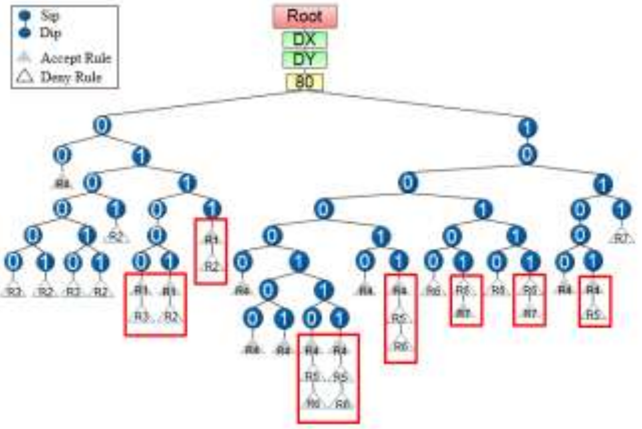


Figure 5. The corresponding ARAR tree of Fig. 4

B. *Diagnosis with ARAR Tree*

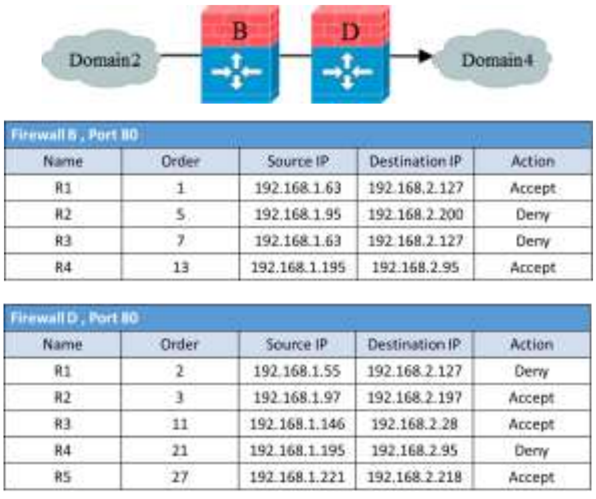
From Fig. 5, it can be found that there are nine branches containing more than one △ leaves, which indicates only the IP address spaces of those rules in these branches intersect with one another and hence incur intra-ACL (or, in this case, intra-firewall) rule anomalies. So, we simply have to do the pair-wise rule comparisons for anomaly checking on the rules hung at the same branch within these nine branches. Comparing to [2-4], without the ARAR tree, three times rule pair-wise comparisons (i.e., 24 times) are required for anomaly checking.

To isolate the inter-ACL (or, here, inter-firewall) rule anomalies, in our approach, it can easily be done by simply re-using the ARAR trees built for the diagnosis of intra-ACL (or intra-firewall) rule anomalies. We can first do the intra-ACL anomaly diagnosis for rules inside two designated firewalls individually, which can lead to the construction of two ARAR trees separately for the diagnosis of intra-ACL rule anomalies. Later, to obtain the diagnosis of inter-ACL (or inter-firewall) rule anomalies between these two firewalls, tree integration can be made by adjusting the tree level/height and collecting the leave △ nodes belonging to the same branch of the two individual ARAR trees and putting them together under the same branch of a new ARAR tree for inter-ACL rule anomaly diagnosis. Later, following the same logic 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 ARAR tree for inter-ACL rule anomaly diagnosis.

III. Diagnosis with Fixed-Strike Trie

While doing anomaly diagnosis among firewalls with ARAR tree, the height (or the size) of the tree could become extremely large in some cases; for example, the IP address spaces of rules in traffic filtering plane are of scattered distribution. It means the IP address spaces of those rules are small and scattered in the whole traffic plane. It will dramatically stretch the height of an ARAR tree and then increase the cost and time of anomaly diagnosis. To overcome the problem, another data structure is utilized and constructed – Fixed Stride Trie (FST) [8]. We use an example to illustrate how this data structure facilitates solving this problem.

Like the example network for ARAR Tree mentioned in Section 2.1, the network in Fig. 6 is built in our lab also. And, in Fig. 6, only those filtering rules for port 80 which are configured in firewall B and D are shown, for the routing path from network domain D2 to domain D4. Likewise, the IP address ranges of the source network domain and destination network domain of a designated routing path are employed as two axes to form a traffic plane. Later, with the fields of <source_IP> and <destination_IP>, the IP address space of each ACL filtering rule can be depicted as a rectangle and put their own location on the traffic plane (the leftmost sub-figure in Fig. 7). Once more, referring to the coding tree data structure widely used in data/video compression [6], the traffic plane will be split recursively as well as exponentially until the split block is exactly the same as the address space of some rule(s) (shown in two other sub-figures in Fig. 7).



Firewall B , Port 80				
Name	Order	Source IP	Destination IP	Action
R1	1	192.168.1.63	192.168.2.127	Accept
R2	5	192.168.1.95	192.168.2.200	Deny
R3	7	192.168.1.63	192.168.2.127	Deny
R4	13	192.168.1.195	192.168.2.95	Accept

Firewall D , Port 80				
Name	Order	Source IP	Destination IP	Action
R1	2	192.168.1.55	192.168.2.127	Deny
R2	3	192.168.1.97	192.168.2.197	Accept
R3	11	192.168.1.146	192.168.2.28	Accept
R4	21	192.168.1.195	192.168.2.95	Deny
R5	27	192.168.1.221	192.168.2.218	Accept

Figure 6. Filtering rules with port 80 for the routing path from domain D2 to domain D4

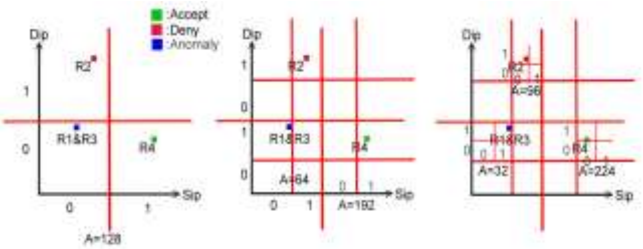


Figure 7. Exponential splitting on a traffic plane

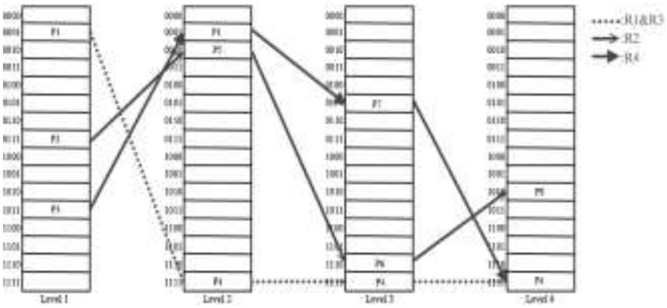


Figure 8. FST of Firewall B

After the splitting, the address space of each rule can be represented by a string of prefix; for example, it would come with 0000-1111-1111-1111 for Firewall B’s rule 1 (Fig. 8) where the prefix string stands for the split space on which rule 1 is located (on the two dimensional traffic plane). And the FST for filtering rules in Firewall B can be constructed accordingly where each strike level of the FST utilizes 4 binary digits (strike length). Then, as shown in Fig. 8, we can find rule 1 and rule 3 in Firewall B traverse through the same path to the end strike (dotted line in Fig. 8). It represents rule 1 and rule 3 can create an Inter-ACL anomaly for Firewall B. Using this logic, we would easily pinpoint all the anomalies among firewall rules. Comparing with the ARAR Tree created for the same example, the height of tree for rule 1 (or rule 3) would go to 8. It means we have to visit 8 internal nodes of the ARAR tree and find the anomaly, while only 4 strike visits would be made in this case.

With the same process mentioned in Section 3.1, we can construct the FST for the filtering rules of firewall D (Fig. 9). Then we can combine the two FSTs from firewall B and firewall D without any tree-height adaption, like ARAR Tree [9], and a new FST suitable for diagnosis of inter-ACL/inter-firewall anomalies between firewall B and firewall D is created. Likewise, we can use it and easily figure out anomalies among firewalls; for example, in Fig. 9, we can find rule 4 in firewall B and rule 4 in firewall D go through the same path from the start strike to the end. It means these two rules would filter some common traffic with the same or different action. This is an inter-firewall anomaly made by rule 1 in firewall B and rule 4 in firewall D. Thus, by doing so, diagnosis with FST can not only decrease the needed space of data structure in this circumstance, but also the system scalability can be retained; i.e., the intra-ACL diagnosis result (FST) of a firewall can be reused to combine the FST of another firewall for rule anomaly diagnosis among firewalls.

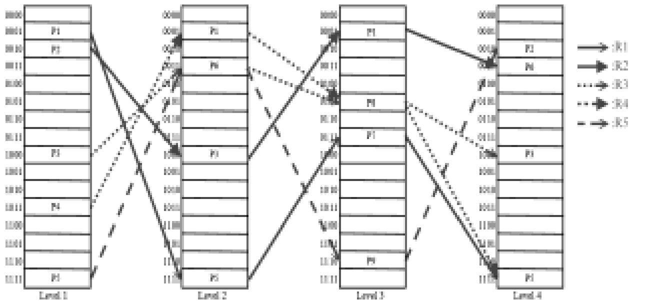


Figure 9. FST of Firewall D

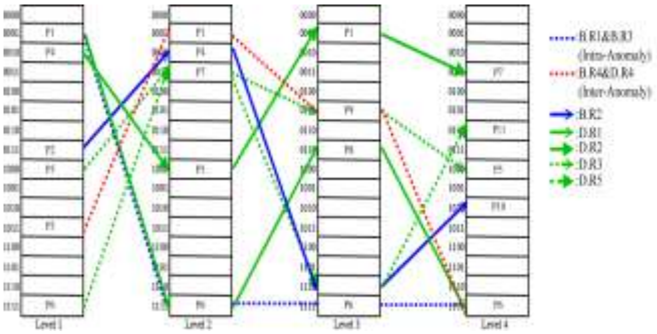


Figure 10. FST for Inter-ACL anomaly diagnosis between Firewall B and D

IV. Performance Evaluations

A comprehensive set of experiments had been conducted in our lab to obtain the performance evaluations [10]. The experimental results would not include those of [2-4] this time, which needs a substantial amount of pair-wise rule comparisons to do anomaly diagnosis and has been shown an exponential growth of computing time with the number of rules in our previous work [9]. In this paper, three developed systems based on different tree-based data structures/algorithms are compared. Please be kindly noticed that our first RAR-tree-based system is done and tested in 2013 [7], which splits the traffic plane into fixed-sized smaller blocks (where A is the size of blocks in Fig. 12). In performance testing tasks, we try to categorize four different cases to demonstrate the performance of each of our three systems:

- Case 1: rules with larger filtering address space and scattered distribution in the traffic plane;
- Case 2: rules with larger filtering address space and plenty of space intersections in the traffic plane;
- Case 3: rules with tiny (or much smaller) filtering address space and uniform distribution in the traffic plane;
- Case 4: rules with tiny (or much smaller) filtering address space and scattered distribution in the traffic plane.

From Fig. 12, it turns out that no matter in which case there is an apparent result in diagnosis performance: The FST-based system is better than the ARAR tree-based one which is much better than the system developed by RAR Tree.

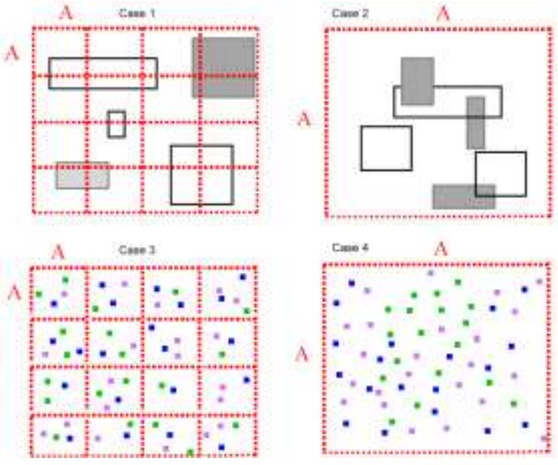
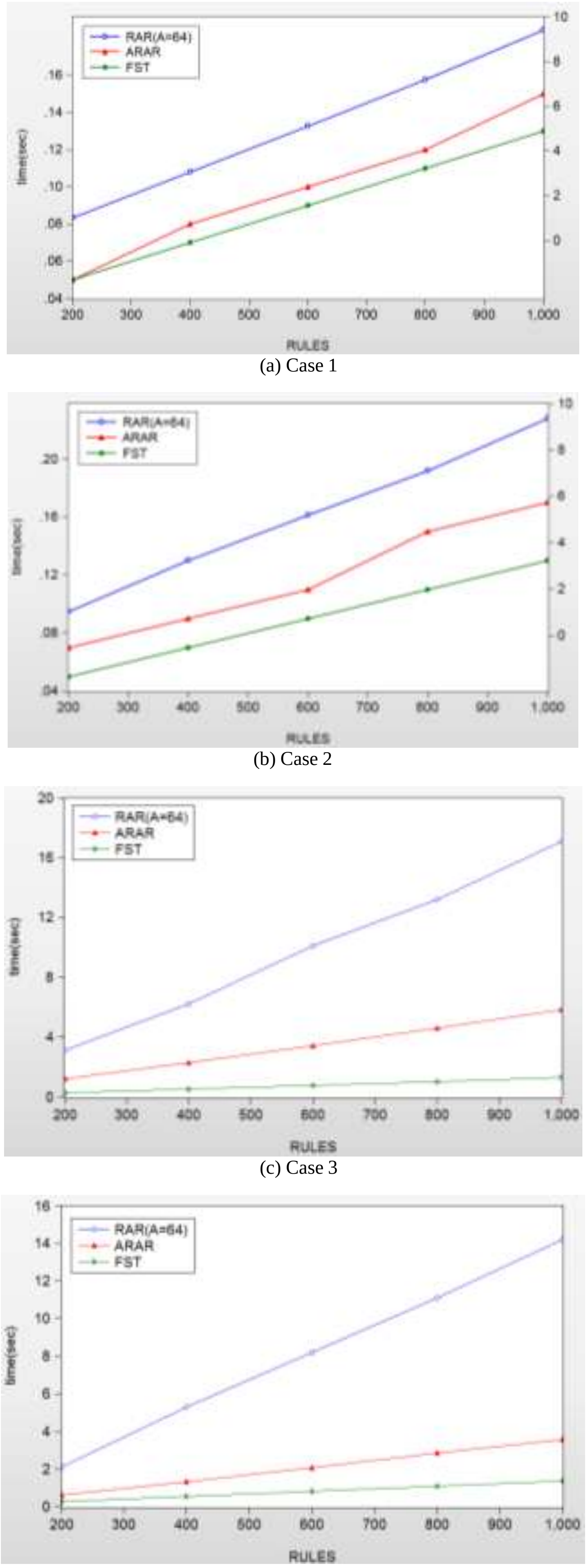


Figure 11. Four different Cases for System Performance Evaluations



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 12. System performance for our developed systems in different cases

Proc. of the Seventh International Conference On Advances in Computing, Electronics and Electrical Technology - CEET 2017.

ISBN: 978-1-63248-126-9 doi: 10.15224/ 978-1-63248-126-9-31

V. Conclusions and Future Work

With implementations of the ARAR tree and FST, both of our diagnosis mechanisms for firewall rule anomalies can meet the planned requirements: Efficiency, scalability, and feasibility. Shortening the time needed for the diagnosis of rule anomalies among/inside firewalls means reducing the possibilities of the loss of company estates, caused by network attacks. This is also very important for those systems which run on-line and need speedy responses regularly with their users, e.g., on-line banking or online shopping. They tolerate no room for a second service break, leading to prompt and correct firewalls configuration in response to various threats coming from networks. In reality, a prototype system based on our developed diagnosis mechanisms went live since early Apr. this year, to facilitate the configuration and management of firewalls in our campus network.

Although we get a noticeable achievement on our system development, as the next steps, more interesting ingredients and plenty of technical challenges are expected to be considered and dealt with to complete our diagnosis system and meet coming demands, e.g., migrating the current mechanism(s) to IPv6 networking environment, adding inspection functions for behavior mismatching among firewalls, and developing multi-dimensional and usable visualized tools.

Acknowledgment

This work is mainly supported by MOST, R.O.C., under contract MOST-104-2221-E-035-023.

References

[1] T. Wong, “On the usability of firewall configuration,” Symposium on Usable Privacy and Security, 2008.

[2] E. Al-Shaer, H. Hamed, R. Boutaba, and M. Hasan, “Conflict classification and analysis of distributed firewall policies,” IEEE J. on Selected Areas in Communications, Vol. 23, No. 10, pp. 2069–2084, 2005.

[3] E. Al-Shaer and H. Hamed, “Discovery of policy anomalies in distributed firewalls,” 23rd Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 4, pp. 2605─2616, 2004.

[4] E. Al-Shaer E. and H. Hamed, “Firewall policy advisor for anomaly discovery and rule editing,” 8th International Symposium on Integrated Network Management, pp. 17─30, 2003.

[5] S. Hanasegaran, Y. Yin, Y. Tateiwa, Y. Katayama, and N. Takahashi, “Topological approach to detect conflicts in firewall policies,” 23rd IEEE International Parallel and Distributed Processing Symposium, SSN-1569173665-paper-3.pdf, IEEE Press, 2009.

[6] X. Q. Dai, Data Compression, ISBN 978-957-442-517-4, Flag Press, Taiwan, 2009.

[7] C. S. Chao, “A novel and feasible system for rule Anomaly and behavior mismatching diagnosis among firewalls,” Springer LNEE234, 2013.

[8] E. Horowitz, S. Sahni, and S. A. Freed, Fundamentals of Data Structures in C, 2nd ed., Computer Silicon Press, ISBN 0-929306-40-6, 2008, pp. 617─625.

[9] C. S. Chao and C. T. Chiu, “An adaptive RAR tree-based diagnosis system for rule anomalies and behavior mismatching among firewalls,” 2013 National Computer Symposium, Session 3, No. 20, 2013.

[10] C. S. Chao and Y. C. Zi, "A Fixed-Stride Trie-based diagnosis system for rule anomalies and behavior mismatching among modern network firewalls," TANet2015, Session 5, No. 1, 2015.

About Author :

Dr. Chi-Shih Chao currently is an associated professor at the Communications Engineering Dept. of Feng Chia University, Taiwan. His research interests include network and system security, network fault management, high-speed networks, and wireless LANs. Dr. Chao received the Annual Best Paper Awards from Taiwan *TANet* in 2015 and *IMP* in 2016, respectively. He also serves for plenty of international relevant conferences and journals. In addition, he is a member of IEEE and Phi-Tau-Phi.