2009 International Conference on Machine Learning and Computing
IPCSIT vol.3 (2011) © (2011) IACSIT Press, Singapore

Performance analysis of range algorithm

Ahmed Farouk [1], Hamdy N.Agiza [2], Elsayed Radwan [3]

[1] Faculty of Computer and Information Sciences, Mansoura University, Egypt
[2] Faculty of Sciences, Mansoura University, Egypt
[3] Faculty of Computer and Information Sciences, Mansoura University, Egypt

Abstract. The need for network security will become more urgent over time. Not only are cybercriminals constantly inventing new ways to steal electronic information, they are getting more nefarious in their use of it. An effective solution for many businesses is the firewall, which monitors and regulates traffic flow between public and private network environments to thwart attacks and prevent the entry of hostile code. Firewall ACLs can contain inconsistencies. There is an inconsistency if different actions can be taken on the same flow of traffic, depending on the ordering of the rules. Many approaches to solve inconsistency and redundancy problems last approach called range algorithm. In this paper analysis performance of the range algorithm will introduce due to range, time processing, CPU utilization, number of rules and accuracy.

Keywords: Network Security, Firewalls, Redundancy and Shadowing of Rules, performance analysis, and Range Algorithm.

1. Introduction

Firewalls are network security components which provide means to filter traffic within corporate networks, as well as to police incoming and out coming interaction with the Internet [5]. For this purpose, it is necessary to configure firewalls with a set of filtering rules. Nevertheless, the existence of errors in a set of filtering rules is very likely to degrade the network security policy. The discovering and removal of these configuration errors is a serious and complex problem to solve Firewall ACLs can contain inconsistencies. There is an inconsistency if different actions can be taken on the same flow of traffic, depending on the ordering of the rules. Although many algorithms have been proposed to solve this problem, all reviewed ones work with the full ACL with no approximate heuristics, giving minimal and complete results, but making the problem intractable for large, real-life ACLs [6].

Although firewall security has been given strong attention in the research community, the emphasis was mostly on the filtering performance and hardware support issues [3, 7]. On the other hand, few related work [8] present a resolution for the correlation conflict problem only. Other approaches [1, 8] propose using a high-level policy language to define and analyze firewall policies and then map this language to filtering rules. Firewall query-based languages based on filtering rules are also proposed in [7]. In recent work present range algorithm to detect consistent and inconsistent rule. Range Algorithm leads to solve conflict and overlap problems depend on hybrid comparison and intersection modules compared to previous approaches that use only intersection method that leads to inaccurate results [4]format of packet filtering rules in a firewall policy: <Order><protocol><src_ip><src_port><dst_ip><dst_port> <action> [6] In this work the rules defined will be written as follow Ri : {condition }→decision [2] equivalent to where i is the relative position of the rule within the set of rules, decision i is a Boolean expression in {accept; deny}, and {condition} I is a conjunctive set of condition attributes such that {condition} I equals A1 ^ A2 ^ ::: ^ Ap, and p is the number of condition attributes of the given filtering rules. Attributes of conditions are Source and Destination IP address, in our work we suppose Source port, Destination port and IP protocol true value, there're two types of decision are Accept and Deny,. On the other hand, few related work [3] present a

resolution for the correlation conflict problem only. Other approaches [1] propose using a high-level policy language to define and analyze firewall policies and then map this language to filtering rules. Firewall query-based languages based on filtering rules are also proposed in [6]. So in general, In this paper a new progress in this area because it offers new technique for complete anomaly discovery and rule editing that can be applied on legacy firewall policies of low-level filtering rule representation, solving Disadvantages of latest approach are Inaccurate results, Number of algorithms used to perform functions, Time is huge, Performance is low.

2. Range algorithm

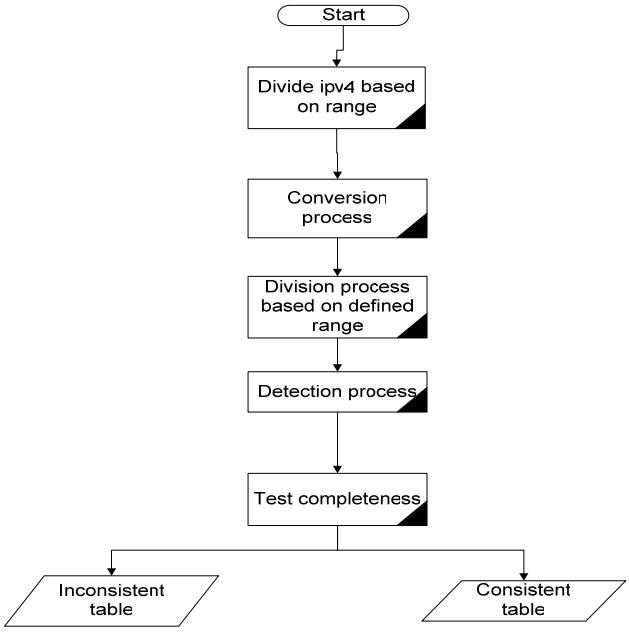

Figure 1: Flow chart of range algorithm

Range Algorithm leads to solve conflict and overlap problems as in figure 1 through five stages Thus, first stage divided ip function divide ipv4 address whose input is Range specified in algorithm, Using ceiling function that approximate to the largest integer and comparing result to our limit 254.Second stage conversion function whose input is the initial set of filtering rules, and output is conversion R by extract last byte from source and destination convert decision to be 0 and 1.Third stage division function whose input is extracted source and destination of R, in this stage source and destination of each rule will be compared to divided IPv4 result from first stage and put result in division table .Fourth stage detection conversion function whose input is the initial set of filtering rules, and output is conversion R by extract last byte from source and destination convert decision to be 0 and 1.Third stage division function whose input is extracted source and destination of R, in this stage source and destination of each rule will be compared to divided IPv4 result from first stage and put result in division table .Fourth stage detection function is recursive whose input is R from division table and take each rule and compare to other rules , that if there is intersection between source and also destination between rules with same decision then extract consistent rules by comparing if there is intersection between source and also destination between rules with different decisions then extract consistent and inconsistent between rules by comparing , else add two rules directly to consistent , because no intersection between source or destination that means two rules applied to different destination or two different sources , the output of the main detection function is the set which results as a transformation of the initial set R. This new set is equivalent to the initial one, R, and all its rules are completely disjoint. Therefore, the resulting set is free of both redundancy and shadowing of rules, as well as any other possible configuration error. Fifth stage test completeness by drawing result of consistent rules and if drawing is independent so, achieving to best case of independency [4].

3. Performance analysis

3.1 Accuracy between past firewall configuration approaches' and range algorithm

Example:-

As in Table 1 example of a set of filtering rules with five condition attributes that contains overlap, conflict, inconsistency

Table1: Example of a set of filtering rules with five conditions attributes.

order	condition					decision
	(p)rotocol	(s)ource	(sP)ort	(d)estination	(dP)ort	
1	any	xxx.xxx.xxx.[001,030]	any	xxx.xxx.xxx.[020,045]	any	deny
2	any	xxx.xxx.xxx.[020,060]	any	xxx.xxx.xxx.[025,035]	any	accept
3	any	xxx.xxx.xxx.[040,070]	any	xxx.xxx.xxx.[020,045]	any	accept
4	any	xxx.xxx.xxx.[015,045]	any	xxx.xxx.xxx.[025,030]	any	deny
5	any	xxx.xxx.xxx.[025,045]	any	xxx.xxx.xxx.[020,040]	any	accept

By applying Range algorithm as shown in Figure 2 apply hybrid intersection and comparisons sequence modules recursively that result is inconsistent table that contains only inconsistent rules that cause inconsistency problems and consistent table that contains only consistent rules

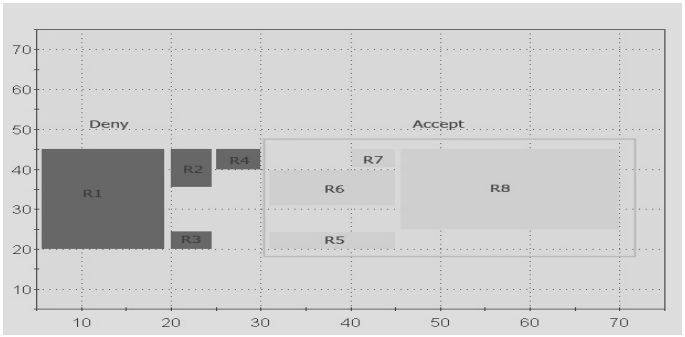

Figure 2: result of applying Range algorithm

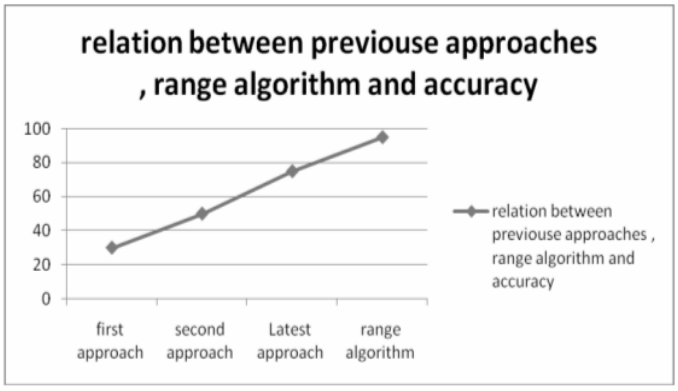

Figure 3: Previous approaches and range algorithm/accuracy

By applying Range algorithm as shown in Figure 2 of resulting rules that indicate achieving best case for inconsistency detection, that no overlap each rule is separated from other rules that means shadowing and redundancy our goal problems solved using this new technique, as shown in Figure 3 detect relation between Previous approaches and range algorithm/accuracy

3.2 Range, processing time

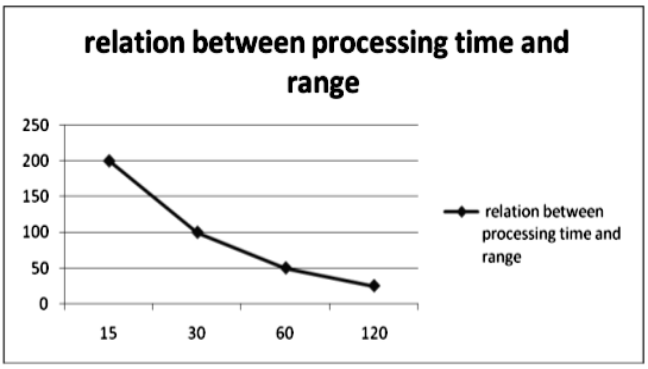

Figure 4: relation between processing time and range

Detect relation between Range and Processing time is very simple because while range increase, time processing decrease, so relation between processing time and range is reverse relation. for example if range selected equal 15 then will take more time than by comparing to range selecting equal 60 , so in Figure 4 shown relation between processing time and range

3.3 Range, CPU utilization

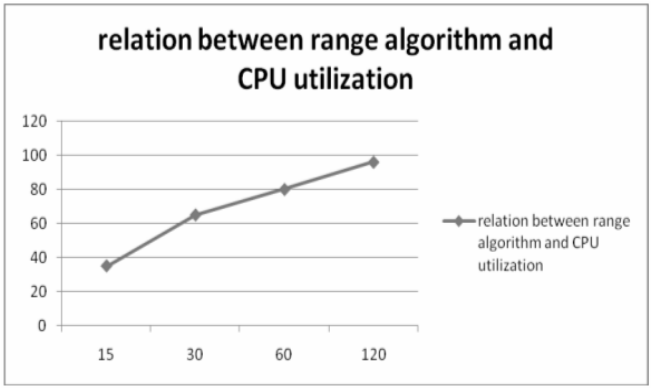

Figure 6: relation between range algorithm and CPU utilization

Most important performance measure is CPU utilization, so while processing time decrease CPU utilization increase, so while range increase, processing time decrease and CPU utilization increase, so relation between range algorithm and CPU utilization is reverse relation as in Figure 6

3.4 Number of rules & processing time

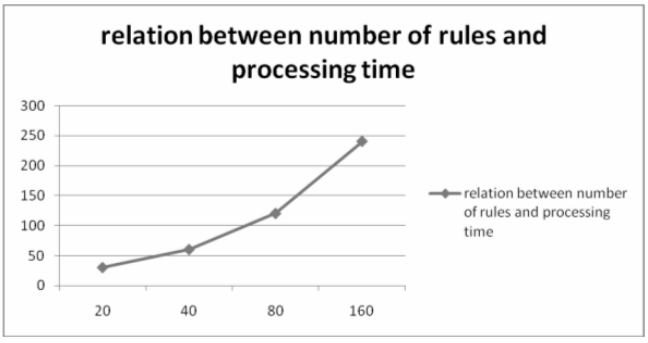

Figure 7: relation between number of rules and processing time

Relation between number of rules and processing time is extrusive relation. As number of rules increase, processing time will also increase as in Figure 7 that show relation between number of rules and processing time

4. Conclusions and future work

A firewall is a system or group of systems that enforces an access control policy between two networks. The actual means by which this is accomplished varies widely, but in principle, the firewall can be thought of as a pair of mechanisms: one which exists to block traffic, and the other which exists to permit traffic. Probably the most important thing to recognize about a firewall is that it implements an access control policy. If you don't have a good idea of what kind of access you want to allow or to deny, a firewall really won't help you. It's also important to recognize that the firewall's configuration, because it is a mechanism for enforcing policy, imposes its policy on everything behind it. Administrators for firewalls managing the connectivity for a large number of hosts therefore have a heavy responsibility and many problems occurred when configuration of firewall systems so in this paper we suggest algorithm that can guide to construct firewall systems free of inconsistency that remove inconsistent rules. In this paper in this paper analysis performance of the range algorithm due to range, time processing, CPU utilization, number of rules and accuracy because in latest approach are the most important Problem Inaccurate results that solved and reach the best case " independency case ", so more accurate results can achieve using range algorithm Some other advantages of our approach are the following. First of all, our transformation process verifies that the resulting rules are completely independent between them. Otherwise, each redundant or shadowed rule considered as useless during the process is removed from the configuration. On the other hand, the discovering process provides an evidence of error to the administration console. This way, the security officer can check whether the security policy is consistent, in order to verify the correctness of the process So results from analysis of performance are indicate achieving best case for inconsistency detection so best accuracy , for Detect relation between Range and Processing time is very simple because while range increase, time processing decrease ,Most important performance measure is CPU utilization, so while processing time decrease CPU utilization increase, so while range increase, processing time decrease and CPU utilization increase, As number of rules increase, processing time will also increase Our future research plan includes, extending the proposed techniques to handle class B and class A, and use rough sets to detect inconsistency also use c4.5 algorithm to detect inconsistency

5. References

[1] A Mayer, A Wool, E Ziskind, "Fang: A Firewall Analysis Engine." *IEEE SYMPOSIUM ON SECURITY AND PRIVACY*, 2000

[2] Al-Shaer, E. S. and Hamed, H. H. Discovery of Policy Anomalies in Distributed Firewalls. In *23rd Annual Joint Conference of the IEEE Computer and Communications Societies,* 2004

[3] B. Hari, S. Suri and G. Parulkar. "Detecting and Resolving Packet Filter Conflicts" *Proceedings of IEEE INFOCOM'00*, March 2000

[4] Detecting inconsistent firewall configuration rules using range algorithm [under publication]

[5] Fr´ed´eric Cuppens, Nora Cuppens-Boulahia, and Joaqu´ınGarc´ıa-Alfaro,Detection of Network Security Component Misconfiguration by Rewriting and Correlation ,Universitat Autonoma de Barcelona, December 2007,page 1-3

[6] Pozo4, S., Ceballos, R., Gasca, R.M. "Polynomial Heuristic Algorithms for Inconsistency Characterization in Firewall Rule Sets"*2nd International Conference on Emerging Security Information, Systems and Technologies* (SECURWARE). Cap Esterel, France. IEEE Computer Society Press, 2008

[7] S. Hazelhusrt. "Algorithms for Analyzing Firewall and Router Access Lists" Technical Report TRWitsCS-, Department of Computer Science, University of the Witwatersrand, South Africa, July 1999

[8] S. Hinrichs. "Policy-Based Management: Bridging the Gap." *Proceedings of 15th Annual Computer Security Applications Conference* (ACSAC'99), December 1999