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

Detecting inconsistent firewall configuration rules using range algorithm

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

[1] Faculty of Computer and Information Sciences, Mansoura University, Egypt, Email: Ahmed.Metwaly85@yahoo.com

[2] Faculty of Sciences, Mansoura University, Egypt, Email: Agizah@mans.edu.eg

[3] Faculty of Computer and Information Sciences, Mansoura University, Egypt, Email: Elsfradwan@yahoo.com

Abstract. Many companies and organizations use firewalls to control the access to their network infrastructure. When processing packages, conflicts due to rule overlaps can occur within the filtering policy. To solve these conflicts most firewall implementation use a first matching strategy through the ordering of rules. This way each packet processed by the firewall is mapped to the decision of the rule with highest priority. This strategy introduces however new configuration errors such as shadowing of rules and redundancy lead to inaccurate results. In this paper new algorithm called range algorithm introduced to get the best case for solving conflict and shadowing problems. Also get result rules that is free inconsistency and finding rules that cause inconsistency.

Keywords: Network Security, Firewalls, Redundancy and Shadowing of Rules, Conflict, 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]. 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 [7] in this paper a different approach introduced. The following is the common format of packet filtering rules in a firewall policy: <Order><protocol><src_ip><src_port><dst_ip><dst_port> <action> [7] In this work the rules defined will be written as follows Ri : {condition}→ decision [3] 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 .On the other hand, few related work [2] 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, this paper offers new techniques 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

+ Corresponding author.
E-mail address: Ahmed.Metwaly85@yahoo.com

2. PRELIMINARIES

2.1. Analysis of consistency problems

To understand the problem, it is important to firstly review the inconsistencies characterized in the bibliography. A complete characterization that includes inconsistency, shadowing, and redundancy has been given in [3,5and 7]. Inconsistency: Two rules Ri, Rj ∈RS are inconsistent if and only if the intersection of each of all of its selectors R [k] is not empty, and they have different actions, independently of their priorities. The inconsistency between two rules expresses the possibility of an undesirable effect in the semantics of the rule set. The semantics of the rule set changes if an inconsistent rule is removed.

Definition1

$$Inconsistent(R_i, RS), 1 \le i \le n \Leftrightarrow \exists R_j \in RS, 1 \le j \le n, j \ne i \bullet$$
$$R_i[k] \cap R_j[k] \ne \varnothing \wedge R_i[Action] \ne R_j[Action]$$
$$\forall k \in \{protocol, src\_ip, src\_prt, dst\_ip, dst\_prt\}$$

Inconsistency of one rule in a RS

$$Inconsistent(R_i, R_j, RS), 1 \le i, j \le n, i \ne j \Leftrightarrow$$
$$R_i[k] \cap R_j[k] \ne \varnothing \wedge R_i[Action] \ne R_j[Action]$$
$$\forall k \in \{protocol, src\_ip, src\_prt, dst\_ip, dst\_prt\}$$

Shadow: A rule Ry is shadowed by another rule Rx, with Rx>Ry, if all of its selectors to or supersets of the selectors of Ry, and Rx and Ry have different action.

Definition 2

$$\exists R_x, R_y \in RS \bullet R_x > R_y \bullet Shadow(R_y) \Leftrightarrow$$
$$\forall k \bullet R_y[k] \subset R_x[k] \wedge R_x[Action] \ne R_y[Action]$$
$$k \in \{protocol, src\_ip, src\_prt, dst\_ip, dst\_prt\}$$

Shadow

$$\exists R_x, R_y \in RS \bullet R_x > R_y \bullet ExactShadow(R_y) \Leftrightarrow$$
$$\forall k \bullet R_y[k] = R_x[k] \wedge R_x[Action] \ne R_y[Action]$$
$$k \in \{protocol, src\_ip, src\_prt, dst\_ip, dst\_prt\}$$

Exact shadow

Redundancy: A rule Rx is redundant to another rule Ry, with Rx>Ry, if all of its selectors are subsets or equal to the selectors of Rx, they have the same action, and if there is no rule between Rx and Ry which is correlated or subset of Rx. Redundancy of Ry respect to Rx is symmetrical. Redundancy is not really an inconsistency, since if all redundant rules are removed; the semantic of the rule set does not change.

Definition 3

$$\exists R_x, R_y \in RS \bullet R_x > R_y \bullet Redundant(R_x) \Leftrightarrow$$
$$\forall k \bullet R_x[k] \subseteq R_y[k] \wedge R_x[Action] = R_y[Action] \wedge$$
$$\neg\exists R_z \in RS, R_x > R_z > R_y \bullet$$
$$Correlation(R_x, R_z) \vee Generalization(R_z)$$
$$k \in \{protocol, src\_ip, src\_prt, dst\_ip, dst\_prt\}$$

$$\exists R_x, R_y \in RS \bullet R_x > R_y \bullet Redundant(R_y) \Leftrightarrow$$
$$\forall k \bullet R_y[k] \subset R_x[k] \vee \forall k \bullet R_y[k] = R_x[k] \wedge$$
$$R_x[Action] = R_y[Action]$$
$$k \in \{protocol, src\_ip, src\_prt, dst\_ip, dst\_prt\}$$

2.2. Related and previous work

A first approach to get a firewall configuration free of errors is by applying a formal security model to express the network security policy. Nonetheless, this approach is not enough to ensure that the firewall configuration is completely free of errors [4].A second approach in a configuration set, two rules are in conflict when the first rule in order matches some packets that match the second rule, and the second rule also matches some of the packets that match the first rule. This approach is very limited since it does not detect what we consider serious misconfiguration errors, redundancy and shadowing of rules [8].Latest approach goal is to find minimum set of rules that cause the security policy don't change by Detection of shadowing rules and removes it, Detection of redundancy rules and solves it and Test completeness of security policy and ensure that don't change Disadvantages of latest approach are Inaccurate results, Number of algorithms used to perform functions, Time is huge, Performance is low

3. Range algorithm

3.1. structure of range algorithm

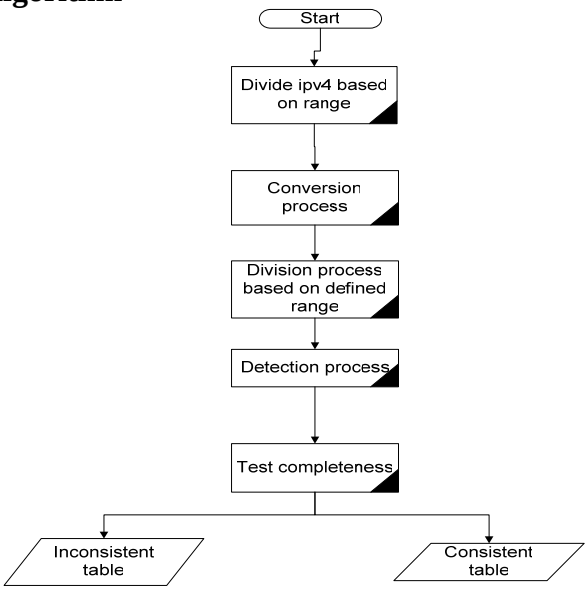

Figure 1: Flow chart of range algorithm

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

3.2. Applying the Algorithms

Example:- As in Table 1 example of a set of filtering rules with five condition attributes that contains overlap, conflict, inconsistency as in Figure 2

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

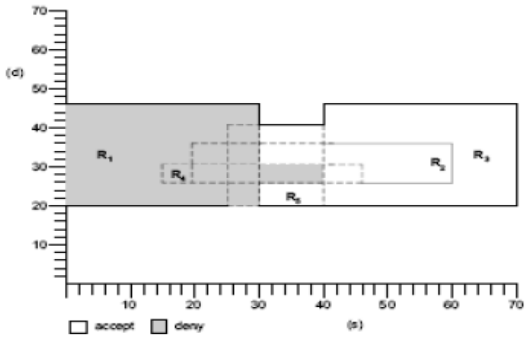

Figure 2 Main overlaps with rules of Table 1

Table2: final inconsistent table

/*construct final inconsistent table that contains only inconsistent rules that cause conflict and overlap */				
Si	Sj	Di	Dj	Aij
20	45	25	30	0
20	45	25	30	1
25	30	20	40	0
25	30	20	40	1
31	45	25	30	0
31	45	25	30	1

Table3: final consistent table

/*construct final consistent table that contains only consistent rules */				
Si	Sj	Di	Dj	Aij
1	19	20	45	0
20	24	20	24	0
20	24	36	45	0
20	30	40	45	0
31	45	20	24	1
31	45	31	40	1
40	45	41	45	1
46	70	25	45	1

By applying Range algorithm that uses 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 as shown in Table2, Table3 respectively and as shown in Figure 3 that 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.

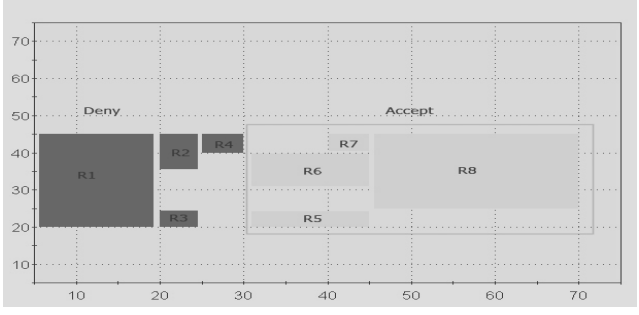

Figure 3: result of applying Range algorithm

4. Conclusions and future work

A firewall is a system or group of systems that enforces an access control policy between two networks. 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 new algorithm introduced that can guide to construct firewall systems free of inconsistency that remove inconsistent rules. In this paper range algorithm solve many problems 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. The complete independence between rules, moreover, enables the possibility to perform a second rewriting of rules in a positive manner " final consistent table " or in a negative manner " final inconsistent table " After performing this second transformation, the security officer will have a clear view of the accepted traffic or the rejected traffic in our future work that we will find the probability for each inconsistent rule that we can expect which will be accepted. Our future research plan includes detecting inconsistent firewall rules probability using hybrid rough sets and range algorithm using importance rule to detect probability of each inconsistent rule that higher importance rule probability cause first to execute, evaluation of range algorithm related to relation between range and processing time, and extending the proposed techniques to handle class B and class A.

5. References

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

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

[3] Cuppens, F., Cuppens-Boulahia, N., and J. Garcia-Alfaro. November 2005, Misconfiguration Management of Network Security Components. *In Proceedings of the 7th International Symposium on System and Information Security, Sao Paulo, Brazil*

[4] Frédéric Cuppens, Nora Cuppens-Boulahia and Alexandre Miège, 2004 and formal approach to specify and deploy a network security policy. *In Second Workshop on Formal Aspects in Security and Trust,* pages 203-218

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

[6] P. Eronen and J. Zitting, November 2001 "An Expert System for Analyzing Firewall Rules." *Proceedings of 6th Nordic Workshop on Secure IT-Systems (NordSec 2001),*

[7] Pozo4, S., Ceballos, R., Gasca, R.M, 2008 "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.*