Proceedings of International Conference on Computational Intelligence & IoT
(ICCIIoT) 2018

Firewall Rule Anomaly Detection: A Survey

[1]Zahid Ahmed, [2]S Md S Askari

Department of Computer Science and Engineering

Rajiv Gandhi University

Arunachal Pradesh, India

[1]zahidprince786@gmail.com, [2]askari.sikdar@gmail.com

Abstract

Firewalls are the fundamental component in the security mechanism of a network. Firewall is a series of ordered filtering rules set by the administrator of a network. To set a large number of rules is a complex task. Sometimes in setting the rules in firewall system, there may be some error. Which may lead to conflicts between two or more rules which will results in rule anomalies. A major contribution of this work is to know the mechanism of firewall system, details about the anomalies, and various types of anomalies and the different techniques to sort out the anomalies.
KEYWORDS: Firewall Rule relation, Rule anomaly, Co-relation.

1 Introduction

In the last few years security has become the most burning issue in all over the world. Day by day the threat of network attacks are increasing hence the challenges are also increasing to protect the networks (small home network to large enterprise networks) .Everyone is aware of protecting the networks from unwanted attacks as well as the unauthorized access. Firewall is one of the key element which protects the network from unwanted attacks and unauthorized traffic. Firewall performs the task by filtering the network traffic which are coming inside or going out from the network. Filtering process takes place on the basis of some predefined set of ordered rules. As the number of rules increases the difficulty of adding a new rule or modifying some existing rules also increases and the possibilities of conflicting between two or more rules also increases. Conflict between two or more rules is known as ANOMALY. [3, 5, 4, 1, 2, 6] When anomaly occurs the firewall does not work properly. Sometimes it may happen that firewall allows those packets which should have been blocked and blocks those packets which should have been allowed [2, 10, 15, 17, 20]. There are several types of anomalies in firewall system. Many researches have been done to find permanent solutions for these anomalies and many algorithms have been proposed for detection and correction of these firewall rule anomalies in a firewall system.

This paper comprehensively attempts to analyze various types of rule set anomalies in firewall system and to present different solutions by the scientists in the recent time.

2 Firewall Rules and Rule Relation

A firewall is a network preservation system designed to avoid illegal entry to or from a private network. It acts as a security gourd in the gateway. Whenever a packet comes inside of a network or goes outside from the network, the packet passes through the firewall. A firewall examines a packet and decides whether it should be allowed to pass or should be blocked. A firewall can be implemented both as a hardware or software and sometimes by combining both the hardware and software. Network firewalls are often used to avoid illegal Internet users from entering a private network which is connected to the Internet [3].

Firewall rule format:

Firewall rule format depends on the network administrator. But the standard format consist the following fields: protocol, source IP, source port, destination IP, destination port and action.[3, 5, 4, 1, 2, 6].

<order><protocol><s-IP><s-port><d-IP><d-port><action>

An example of filtering rule format.

The rule format consist two parts.

• **Predicate or Header**

In the previous study of Khummanee et al [19] have defined a part of rule format as predicate part on the other hand in another study by Benelbahri and Bouhoula [8] have defined the same part as header part which consists of 5 fields. These are sequentially 1 bit Protocol, 32bits source IP address, 16 bits source port number, 32 bits destination IP address and 16bits destination port number. Predicate holds the address where from the packet is coming and to where the packet is going to. Simply it is an address part.

• **Decesion**

The decision part consist only 1 bit action field which holds the decision information. According to this action field the firewall, allows or blocks a packet.

2.1 Firewall rule relations

Filtering rules control the whole firewall system and based on these rules firewall performs the task. Firewall policy is a collection of a huge number of filtering rules, set by the network administrator. Among these rules one rule may be related with other rules. And to build an efficient firewall system, an administrator needs to determine all the possible relations between them. As described by Al-Shaer et al [3] the rule relation can be categorized as:

• **Completely disjoint (CD)**

Rules $\Re_a$ and $\Re_b$ are said to be completely disjoint if the following equation is satisfied-

$$\forall p, \Re_a[p] \cap \Re_b[p] = \varphi \quad (1)$$

Where *a* and *b* are the orders of the rules and *p* is the individual fields of the rules.

i.e. *p ∈{protocol,s-IP,s-port,d-IP,d-port}*

• **Exactly matching (EM)**

Rules $\Re_a$ and $\Re_b$ are said to be exactly matching if the following equation is satisfied-

$$\forall p, [(\Re_a[p] \cap \Re_b[p] = \Re_a[p]) \wedge (\Re_a[p] \cap \Re_b[p] = \Re_b[p])] \quad (2)$$

Where *a* and *b* are the orders of the rules and *p* is the individual fields of the rules.

i.e. *p ∈{protocol,s-IP,s-port,d-IP,d-port}*

• **Inclusively matching (IM)**

Rules $\Re_a$ and $\Re_b$ are said to be inclusively matching if the following equation is satisfied-

$$\forall p, [(\Re_a[p] \cap \Re_b[p] = \Re_a[p]) \wedge (\Re_a[p] \cap \Re_b[p] \neq \Re_b[p])] \quad (3)$$

Where *a* and *b* are the orders of the rules and *p* is the individual fields of the rules.

i.e. *p ∈{protocol,s-IP,s-port,d-IP,d-port}*

• **Partially disjoint (PD)**

Rules $\Re_a$ and $\Re_b$ are said to be partially disjoint if the following equation is satisfied-

$$\exists p, [(\Re_a[p] \cap \Re_b[p] = \Re_a[p]) \vee (\Re_a[p] \cap \Re_b[p] = \Re_b[p])] \wedge \exists q [(\Re_a[q] \cap \Re_b[q] = \varphi)] \quad (4)$$

Where *a* and *b* are the orders of the rules and *p,q* are the individual fields of the rules.
i.e. *p,q* ∈*{protocol,s-IP,s-port,d-IP,d-port}* and *p≠q.*

• Correlated (C)
Rules $\Re_a$ and $\Re_b$ are said to be correlated if the following equation is satisfied:

$$\exists p, [(\Re_a[p] \cap \Re_b[p] = \Re_a[p])] \wedge \exists q[(\Re_a[q] \cap \Re_b[q] = \Re_b[q])] \quad (5)$$

Where *a* and *b* are the orders of the rules and *p,q* are the individual fields of the rules.
i.e. *p,q* ∈*{protocol,s-IP,s-port,d-IP,d-port}* and *p≠q.*

3 Problems of firewall rule management

It is highly problematic to manage the filtering rules inside a firewall system. The main problem of firewall rule management is anomaly detection and removal [19]. If the filtering rules are not set properly, it may create some conflict inside the firewall system. Conflict occurs when two or more firewall rules are overlapped and all are having different action [3, 19]. Conflict within firewall rules can make a system in a halting state or it may lead a system to perform improperly [11].

For example, suppose there are two rules $\Re_1$&$\Re_2$.
$\Re_1$: TCP,150.120.∗.∗,any,190.160.70.∗,any,accept
$\Re_2$: TCP,150.120.70.43,80,190.160.70.10,any,deny

Here both the rules are partially overlapped and they are inclusively matching with each other. i.e.$\Re_1$RIM$\Re_2$. But both of these two rules are having different actions. So these two rules are conflicting with each other, and this scenario is considered as an anomaly.
iff $\forall i : \Re_x[i] \cap \Re_y[i] \neq \varphi \wedge \Re_x[action] \neq \Re_y[action]$
Where *i* ∈*{protocol,s−IP,s−port,d−IP,d−port}*

4 Type of Anomalies

Because of the conflict between two rules, anomalies occur inside a firewall system. Several researches have been done on firewall rule anomalies. In different studies the types of anomalies are categorized and described differently [3, 16, 11, 8, 22, 9]. Based on those the following are the different types of rule anomalies in firewall system:

• Shadowing anomaly
As name implies shadowing anomaly takes place when one rule becomes the shadow of another previous rule. It means when the predicate part of one rule matches the predicate part of another previous rule but both are having different actions then the rule is known as shadowed rule of that previous rule.

Rule $\Re_a$ is said to be a shadowed by rule $\Re_b$ if the following equation is satisfied-

$$(\forall p, (\Re_a[p] == \Re_b[p]) \wedge (\Re_a[Decision] \neq \Re_b[Decision])) \quad (6)$$

$$(\forall p, (\Re_a[p] \supset \Re_b[p]) \wedge (\Re_a[Decision] \neq \Re_b[Decision])) \quad (7)$$

Where $\Re$ is a rule and *a* and *b* are orders of the rules such that $a < b$ and *p* is an individual predicate field of a rule.
i.e. p ∈*{protocol,s−IP,s−port,d−IP,d−port}*

• Correlation anomaly
When two rules have different action fields and some predicate fields of the first rule are equal or subset of the corresponding predicate fields of the second rule and rest predicate fields of the second rule are superset of the corresponding predicate fields of the first rule then the rules are known as correlated rules.

Rule $\Re_a$ is said to be a correlated with rule $\Re_b$ if the following equation is satisfied-

$$((\exists p, \Re_a[p] \cap \Re_b[p] == \Re_a[p]) \wedge (\exists q\, \Re_a[p] \cap \Re_b[q] == \Re_b[q]) \wedge (\Re_a[Decision] \neq \Re_b[Decision])) \quad (8)$$

Where $\Re$ is rule and *a* and *b* are orders of the rules such that $a < b$ and *p,q* are the individual predicate field of a rule .
i.e. *p,q* ∈*{protocol,s-IP,s-port,d-IP,d-port}* and *p≠q.*

• Generalization anomaly
When two rules have different action fields but all predicate fields of the second rule matches with all predicate fields of the first rule then the second rule is known as the generalization of the first rule.

Rule $\Re_a$ is said to be a generalization of rule $\Re_b$ if the following equation is satisfied-

$$((\forall p, (\Re_a[p] \cap \Re_b[p]) = \Re_a[p])) \wedge (\Re_a[Decision] \neq \Re_b[Decision]) \quad (9)$$

Where $\Re$ is a rule and *a* and *b* are orders of the rules such that $a < b$ and *p* is an individual predicate field of a rule.
i.e. p ∈*{protocol,s−IP,s−port,d−IP,d−port}*

• Redundancy anomaly
A filtering rule is said to be redundant if it is present or absent in firewall system does not create any changes in the system. It means if this type of rule is removed from the system security policy will remain unchanged. This type of anomaly occurs when a rule is set twice in a firewall at different position or some times one rule may be a subset of some other rule. Usually this type of anomaly does not create any accuracy problem, but the only thing is that, this type of anomaly make the rule set extra large which may reduce speed and reduces the performance of a system.
Rule $\Re_a$ is said to be a redundant of rule $\Re_b$ if the following equation is satisfied-

$$\forall p, (\Re_a[p] = \Re_b[p]) \wedge (\Re_a[Decision] = \Re_b[Decision]) \quad (10)$$

Where $\Re$ is a rule and *a* and *b* are orders of the rules such that $a < b$ and *p* is an individual predicate field of a rule.
i.e. p ∈*{protocol,s−IP,s−port,d−IP,d−port}*

• Irrelevance anomaly

A filtering rule is said to be irrelevant if it cannot examine any traffic that passes through the firewall system. As name implies this type of rule is irrelevant for a firewall system. Just like the redundant anomaly this type of anomaly also does not create any accuracy problem, but the only thing is that, this type of anomaly makes the rule set extra-large which may reduce speed and the performance of a system.

5 Anomalies Detection Approaches

The anomalies inside of a firewall system can create many problems. One of them is time complexity. Anomalies can decrease the performance of a system, creates improper functionalities and accuracy problems. Because of the anomalies the system cannot take accurate decisions. There are many different methods have been proposed till now to identify these rule anomalies of a firewall system. The different scientist has used different techniques to solve the problem. Followings are the approaches:

5.1 Policy Tree Representation

Al-Shaer et al[3] have proposed the Policy tree representation technique to discover the anomalies inside a firewall system. They have represented the firewall policy with the help of single rooted policy tree. They have claimed that this model maintain an elementary depiction of the filtering rules present in a ruleset and it has the ability to discover the rule relations and anomalies between two or more rules. Each node in a policy tree depict a network field and every path starting at the root node and ending at the leaf node represents an individual rule. Rules with the same field value at a distinct node will share the same branch of the tree. Leaf node consists the decision part of a rule.
They have also proposed an algorithm which is able to detect the anomalies in the firewall system. This algorithm works based on the Policy Tree representation. For two rules $\Re_x$ and $\Re_y$ the algorithm starts with an initial state assuming no relation between them. Then each field of $\Re_x$ is compared to the corresponding field of $\Re_y$. The comparison starts with the protocol

field then source IP address then source port then destination IP address and then stop at destination port. Based on this comparisons, the relation between the rules are determined and also the anomalies are determined.
According to their study in the worst case the algorithm takes 10-240 ms of processing time to examine a security policy of 10-90 rules in a single firewall system. In a large network it takes 20-180 second

5.2 Tree-Rule Firewall

Xiangjian et al [14] have proposed the Tree rule. This is almost similar with the Policy Tree Representation technique proposed by E Al-Shaer et al [3].The only difference is, it is not a firewall anomalies detection technique. This is a technique to set the rules inside a firewall system which will provide the network administrator an anomaly free environment. They have claimed that this technique protects the system from conflict between rules. This technique reads the information, which is present in the header of a packet and then, it examines the first attribute of packet with the data, which is present in the root node of a rule tree. Then, the firewall examines the other attributes of the packet in a sequential manner with the help of search operation on relevant nodes at the corresponding levels. Attributes of the root node can be either Source IP, Destination IP or any attributes suitable for that particular work. For each column user can choose any attribute, before generating the Tree rules [14].In the basic design the root node has the number of lines equal to the number of users' computer, so each line is linked to some sub tree with duplicate data. To avoid this problem, they have also improved the design where they replaced the Single IP Address with a range of IP Addresses.
The time complexity of the basic design in worst case is *O(log2N)* and the time complexity of improved design in the worst case is *O(1 + log2N)*
In their study they have considered only three fields Destination IP, Destination port and Source IP.As rest of the fields are not considered, conflict may occur in those fields and which can create anomalies.

5.3 Tuple Based Approach

Benelbahri and Bohoula have proposed Tuple Based Approach for detection tof anomalies inside a firewall system [8]. In this technique for two rules $\Re_a$ and $\Re_b$ they have represented the fields relation as a 4-tuple:
(Filed#, $\Re_a$#, $\Re_b$#,code)
Where "Field" is the order in the filtering rule," $\Re_a$ and $\Re_b$" are the first and second rule and the " Code "is the relationship code.

For *m* number of fields in the filtering rule and n number of filtering rules in filter F. The anomalies are determined by the product of all field codes corresponding to the same indexes a and b of the rules $\Re_a$ and $\Re_b$, and is denoted by:

$$P_{ab} = \Pi^{n-1}{}_{n=1} C_{ab}(n) \quad (11)$$

This model calculates each 4-tuple and checks if it is equal to zero or not. If it is zero then rules are disjoint and then it stores them in the corresponding stack and then jumps to the next code. When an anomaly is detected, the system classifies based on these rules:

- if $P_{ab} > 1$ then redundancy anomaly.
- if $P_{ab} = -1$ then Irrelevant anomaly.
- if P_{ab} mod (g∗s) = 0 then correlation anomaly.
- if P_{ab} mod g = 0 then generalization anomaly.
- if P_{ab} mod s = 0 then shadowing anomaly.

5.4 Inter-Difference Matrices (IDM) Approach

Another work done by Bouhoula et al[9] in their study they have introduced the Inter-Difference matrix approach for detecting anomalies inside a firewall system. In this technique they have used matrix to store the differences between two rules. IDM approach can be defined as a matrix whose element represents the difference of elements belonging to the corresponding vector in the matrix F. For each field an IDM matrix is generated. The number of IDM matrices are equal to the number of fields m and each element e_{ab}, represents the difference between the values of the same field f_{an} and f_{bn} of the filtering rules $\Re_a$ and $\Re_b$. Some extra codes are used to denote the relationship between two or more rules; those are:

- 0 is the code for the difference relationship between two fields and -1 is the code for the difference between two action fields.
- 1 is the code for Equality relationship two fields including the action field.
- 2 is the code for Generalization relationship between two fields.
- 3 is the code for Shadowing relationship between two fields.

The Inter Difference Matrix E

$$E(v_k) = (e_{ab})_{1\le a,b\le n} = (f_{an} - f_{bn})_{1\le a,b\le n} \quad (12)$$

To prove the existence of anomalies they calculate the product of the element of the vector and use the equation:

$$P_{ab} = \Pi^{n-1}{}_{n=1} C_{ab}(n) \quad (11)$$

The above equation is similar with the equation which is used in tuple based approach done by Benelbahri and Bohoula [8].

5.5 Association Rule Mining (ARM)

Golnabi et all [13] have introduced another approach of detecting rule anomalies which is known as Association Rule Mining (ARM) technique. It is a comprehensive method to find out accurate, effective and desirable rules in the rule set of firewall system. Thousands of rules are present in a rule set among them some rules may not have any importance on firewall such as duplicate rules or rules which does not have any action on the decision field. These unwanted rules may create anomalies in the system. With the help of this method, these rules are filtered for further inquiry and to conclude a unique and distinct rule to more general rules.
This method executes in the following steps:[13]

• Analysis of firewall policy rules
Firewall consists of number of filtering rules, set by the network administrator. In this state the rules are, examine and generate a basic set of firewall rules to prepare firewall log raw data to select the rules for data mining.

• Association rule mining
Here it collects and extracts the rules from log raw data prepared in the previous step for performing the Apriori analysis. Here by calculating probability of the observed frequency based on the use of individual rule, the threshold for confidence and threshold for minimal support are evaluated. By applying the Apriori analysis, the rules having the confidence and support more than the respective thresholds are retrieved.

• Mining firewall log using frequency
This is a similar method with Association Rule which reads each line of firewall log raw file, extracts the rules for individual log record and counts its occurrence and generate the count for the individual unique combination of these rules [13].

• Filtering rule generalization
It is an algorithm to produce a minimum number of firewall rules for detecting anomalies and efficient use. Here a decision tree is generated where each level shows the rules from the log raw file.

• Rule ordering
Here the rules are ordered in a significant way based on the outcomes and performance. Here the ordering is not based on its generalization, rather it is relied on the belief that a particular rule should be applied first and the assumption is taken from the previous step.

• Anomalies detection
This is the final step where the anomalies are detected and removed from the rule set.

5.6 XML based open tool for anomalies detection

There is another technique which was proposed by Benelbahri et al[7] based on XML and the technique is known as XML based open tool to detect and resolve filtering rule anomalies. They have proposed this scheme to remove

the conflicts between two rules, and it is based on the concept of putting some resolve filters. With this technique they have tried to characterize all the possible conditions which can create conflict among the rules and gave a solution how to resolve them.
According to them the main reason of conflict between the rules is the organization of rules. Since the rules reflect the security policy, the network administrators are always tried to apply a comprehensive method of organizing filtering rule set in a proper manner. That is if the rules are organized in the proper way then the conflict can be avoided and the anomalies will never be formed. This method executes in two parts. Those are:

• Translator
As name implies the translator translates all representations to a unified and standard representation using XML tool. One of the advantages of XML documents is the hierarchical representation which forms a tree, and they have used this tool to form a tree structure of the rule set. It represents and stores both the filtering rule formats and sequence number in variable form which provides the clarity. They have used the LEX and YACC to carry out the translator module. This module presents each firewall product as a DLL file.

• Analyzer
The analyzer finds in its input an XML file. It extracts the filtering rule fields presented as node in the XML file and it understands the file. Then it applies the technique chosen by the user to detect and resolve the anomalies by invoking the appropriate DLL which holds the algorithm. This module consists of a set of DLLs implementing analyzing algorithm. This tool is presented as successive interfaces. Which allows the user to choose the analysis context [7].

5.7 Graph Based Approach

Fulp [12] has proposed a technique which mainly optimizes the performance of a firewall system. This approach does not directly deal with the anomalies present inside a firewall system rather it improves the performance of firewall system. The listed firewall has some limitations [14]. One of them is the swapping positions between two rules, which changes the firewall policy and creates security problems as well as it reduces the performance. There are a few techniques which can be used to arrange the rules in a systematic manner so that the rules can be listed properly. The technique is based on the ordered sets and directed acyclic graph to arrange the rules in a linear way so that the performance can be improved.
This approach completes the task in three steps:

• Firewall Rules and Policy Models
A rule r is modeled as an ordered set of tuples $r = r[1],r[2],....,r[k]$. Where every tuple belongs to predicate part of individual rule. Here the decision part is not considered. Ordering is necessary among the tuples because it increases the speed of comparing each packet with the corresponding tuples present in the system. Here the tuples are represented in ordered manner and every new rule is compared with other rules to find out the relations among them.

• Modeling Precedence Relationship
The precedence relationship between existing rules are modeled as a directed acyclic graph. The directed acyclic graph represents the proper integrity of individual rule. The integrity refers to the probability of the matching two rules with one another. Let $G = (V,E)$ be a policy directed acyclic graph for some rule set $\Re$, where vertices V represents the rules and edges E represents the precedence of relationship. These graphs are used to represent the integrity of the individual task. This structure is the best way to model the precedence of firewall rules.

• Rule list optimization
In this step using the policy directed acyclic graph the optimal order of firewall rule set is determined based on the precedence of the integrity of individual rule. As a result it is possible to construct a properly ordered and anomaly free rule set. [12].

5.8 Firewall anomaly detection by using model checker and visibility logic

For detecting the anomalies inside a firewall different approaches have been introduced by researchers. This technique is also one among them, and it was introduced by Khorachani and Halle [18]. This technique is based on model checker and visibility logic. They have described the anomalies by using Boolean algebra method and they have solved the problem by their own algorithm which was developed in JAVA platform. They have defined a set of binary relations suitable for distinguishing the visibility concepts such as occlusion ◊, obstruction ○ and covering □ .

The Boolean connectives is a simple process. For a set of rules:
$\Re,r \models *\Re\varphi$, *where* $* \in \{\Diamond,\circ,\Box\}$ and $\Re \in \{\cap,\subseteq,\supseteq\}$ and executes in the following way. First this algorithm computes the current rule r, the rule set $\{r_1,r_2,....,r_n\}$ such that $R(r,r')$ holds. Now for different operators the process is as follows:

• If * is the operator ◊, the expression is true exactly when $\Re,r_{a,b} \models *\Re\varphi$ for some $r_{a,b}$.
• If * is the operator ○, the expression is true exactly when $\Re,r_{a,b} \models *\Re\varphi$.
• If * is the operator □, the expression is true exactly when $\Re,r_{a,b} \models *\Re\varphi$ for all $r_{a,b}$.

This visibility logic takes input according to the visibility logic formula φ, a rule base $\Re$ and a starting rule r and then compares the rule relation by considering their own logic $\Re,r \models \varphi$. Depending upon this logic, it returns an output as ”true” or ”false”. They have tested the approach using four different rule sets ranges from 100 to 4000 rules. The complexity of their algorithm is $O(\Re^k)$. Where $\Re$ is the size of the rule set and k is the number of nested operators in the formula to verify.

5.9 Algorithm Resolve-Anomalies

This is another algorithm proposed by Abedin et al[1] that can simultaneously detect and resolve the anomalies, which are present in the ruleset. By using reorder and split operation this approach generates a new ruleset which is purely anomaly free. The algorithm consists of two modules.

• **Module 1**:- It investigates all the rules present in a ruleset and brings out a set of dissimilar rules, that is anomaly free.

• **Module 2**:- It investigates the rules and tries to join all the rules so that the size of a ruleset can be reduced, thus it can able to bring out a ruleset without introducing any new anomaly.

In this algorithm, they had resolved the anomalies as follows:

In case of shadowing anomaly, if two rules are exactly matched, it preserves the rule having deny in action field. In case of inclusively matched, the rules are shifted in a manner such that the superset rule comes after the subset rule. In case of correlation anomaly, the rules are broken down into several dissimilar parts for inserting them into a list. And finally in the case of redundant anomaly, the rule is removed from the ruleset.
This algorithm performs the operation by maintaining two comprehensive lists of rules.

• Old list
• New list

An old list holds the original firewall configuration with having the original ruleset, and the new list holds ruleset without any anomaly, that is the outputs of this algorithm. This approach is an incremental process, where each rule is taken from the old list and insert it into new rules list in such a manner so that the new rules list remains anomaly free.
An algorithm is there to control the whole process which is known as *Resolve-Anomalies*. First the global rule list is initialized, then it accepts rules from the old list one by one and calls algorithm insert. After that, it scans for the new list to resolve redundant anomalies that might survive on

the list. After that *Insert* Algorithm is used to add a rule into the new list in such a manner so that the list remains anomaly free. In case of an empty list, the rule is added to the list without any condition. Otherwise, the resolve algorithm is used in a for loop. The insert examines each and every new rule with all the rules present in a new list and if any conflicts occur, resolve handles it with a return value true and breaks the loop. Therefore, when resolve handles the rule, it sets the flag as true. Otherwise, the rule is dissimilar or superset with all the rules in a new list and it is added into last of the list. After that, the algorithm *resolve* is used to identify and resolve the anomalies between two rules which are similar. If the first rule sent to algorithm *Resolve*, $\Re$ is the rule being added, and *S* is a rule already present in the new list. In comparing them, following are the possibilities:

• $\Re$ and *S* are equal. If both are equal and having same decision, then any one can be removed, else the one with the reject action is retained.
• $\Re$ is a subset of *S*. $\Re$ inserted before S without considering the decision field.
• $\Re$ is a superset of *S*. $\Re$ may match with proceeding rules present the list, so it is permitted to be examined further. In this situation, no operation is performed.
• $\Re$ and *S* are correlated. The correlated rules are decomposed into dissimilar rules. First, the set of attributes where two rules are different from each other is determined, and then split is called for each of the different attributes in the for loop. After execution of the split, < and S contain the common part of the rules ready to insert.

There is another algorithm which is used to decompose two similar rules and it is known as Algorithm *split*. First it obtains the portions that are dissimilar between two rules and, then it performs the *insert* algorithm on that. After that, it evaluates the common portions of the two rules. If, $\Re$ and *S* are the two rules and *A* be an attribute for which *split* is called. Although these two portions are dissimilar with $\Re$ and *S*, their relation with the other rules in the new list is unknown, the rule is added into the new list by applying *insert* algorithm. Now the common portions of the two rules is evaluated. The dissimilar portions are evaluated before the common part is evaluated and added to the list. After the execution of this whole process, the new rules list contains those rules which are anomaly free.

5.10 Relational Algebra and Raining 2D-Box Model

This is another approach proposed by Mukkapati and Bhargavi [21] to detect the anomalies inside a firewall. In the meantime while all the other approaches are focused to find out the anomalies exist between any two rules in a ruleset, this approach also focused to find out those anomalies, which exist between more than two rules together at the same time. Therefore this approach can able to find out all the hidden anomalies in the ruleset. This method can help the administrator to examine and modify a complex firewall policy too. This approach defined and classified the various kind of anomalies by using a technique which is known as Raining 2D-Box Model. If *Rule-u* is defined as a rule number *u* from the ruleset and $u < v$, then *Rule-v* is the proceeding rule of *Rule-u* then, $\Re_u$ is a rule relation that has been mapped from a *Rule-u* by using some PROJECT operation to exclude the action attribute from the rule format. This approach has also present many definitions to discover the anomalies. Here the anomalies are analyzed and removed by using 6 theorems. And those are as follows:

• **Theorem 1**: A firewall does not change a policy even if *Rule-u* is removed from the ruleset, when *Rule-u* is shadowed.
• **Theorem 2**: A firewall does not change a policy even if *Rule-u* and *Rule-v* are swaped with each other, when Rule-u and Rule-v are sequentiall non-correlated.
• **Theorem 3**: A firewall does not change a policy even if *Rule-u* and *Rule-v* are swaped with each other, where $u < v$, and *Rule-m* is sequentially non-correlated and downward to *Rule-v*, and *Rule-v* is sequentially noncorrelated and upward to *Rule-u*.
• **Theorem 4**: A firewall does not change a policy even if *Rule-u* and *Rule-v* are swaped with each other, where *Rule-u* and *Rule-v* are correlated and *Rule-v* is shadowed, and *Rule-u* is sequentially non-correlated to *Rule−(u−1)* and also downward, and *Rule-v* is sequentially non-correlated upward to *Rule−(u + 1)*

• **Theorem 5**: A firewall does not change a policy even if *Rule-u* is removed from the Ruleset when *Rule-u* is sequentially redundant by Rule-v.
• **Theorem 6**: A firewall does not change a policy even if *Rule-u* is removed from the Ruleset when *Rule-u* is consecutively non-correlated to *Rule−(v −1)* and also downward, and *Rule-u* is redundant by *Rule-v*.

Here those rules are removed which are shadowed, and those rules too which are redundant with each other. When generalization anomaly or correlation anomaly are discovered an alert message is given to the administrator. Removing the shadowing and redundancy anomalies, and rules combination method can able to reduce the size of rule-set and able to make firewall policy easier to understand. Reordering the rules in the ruleset may also help an administrator to understand the ruleset easily. It can also increase the performance of firewall because the rules that are similar by many packets are set on top of the ruleset. Here many rules are combined together by using the UNION operation to the Relations. This method also present another theorem to describe the combination of rules. And the theorem is:

• Theorem 7: *Rule-u* and *Rule-v* can be combined to *Rule-w*, without changing the policy, if $\Re_w = \Re_u \cup \Re_v$, and decision fields are same, and $v = u + 1$.

For proving purpose they have used the 2D-Box Model technique to check the accuracy of anomaly formation so that after removing certain rule the security policy does not change.

6 Conclusion

With the exponential increase of threats on networks, the challenges of protecting networks is also increasing gradually. As the main gateway of a network, the firewall has a complete responsibility to protect the network from unauthorized access. The firewall can only be able to work properly when it has the list of rules without any anomalies. It is a big issue, and everyone is trying to resolve it, many researches have been done worldwide to sort out the anomaly problems. There are various approaches has been introduced by researchers to detect the anomalies, to resolve them and to improve the performance. Everyone is trying their best to find out the finest possible way to fix these problems. The efficient method is highly acceptable. We hope that our effort will help the other researchers around the globe to deploy the various anomaly concepts of firewall system, various issues related with anomalies and it can also give a brief summary of the existing approaches to proceed further in finding new optimal solutions.

References

[1] Muhammad Abedin, Syeda Nessa, Latifur Khan, and Bhavani Thuraisingham. Detection and resolution of anomalies in firewall policy rules. In IFIP Annual Conference on Data and Applications Security and Privacy, pages 15–29. Springer, 2006.
[2] Ehab Al-Shaer. Managing firewall and network-edge security policies. In Network Operations and Management Symposium, 2004. NOMS 2004. IEEE/IFIP, volume 1, pages 926–Vol. IEEE, 2004.
[3] Ehab Al-Shaer, Hazem Hamed, Raouf Boutaba, and Masum Hasan. Conflict classification and analysis of distributed firewall policies. IEEE journal on selected areas in communications, 23(10):2069–2084, 2005.
[4] Ehab S Al-Shaer and Hazem H Hamed. Firewall policy advisor for anomaly discovery and rule editing. In Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on, pages 17–30. IEEE, 2003.
[5] Ehab S Al-Shaer and Hazem H Hamed. Discovery of policy anomalies in distributed firewalls. In INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, volume 4, pages 2605–2616. IEEE, 2004.
[6] Muhammad Qasim Ali, Ehab Al-Shaer, and Taghrid Samak. Firewall policy reconnaissance: Techniques and analysis. IEEE Transactions on Information Forensics and Security, 9(2):296–308, 2014.
[7] Mohamed Anis Benelbahri, Adel Bouhoula, and Zouheir Trabelsi. Xml based open tool for anomalies detection in firewall filtering rules. In

Innovations in Information Technology, 2007. IIT'07. 4th International Conference on, pages 163–167. IEEE, 2007.
[8] Mohammed Anis Benelbahri and Adel Bouhoula. Tuple based approach for anomalies detection within firewall filtering rules. In Computers and Communications, 2007. ISCC 2007. 12th IEEE Symposium on, pages 63–70. IEEE, 2007.
[9] Adel Bouhoula, Zouheir Trabelsi, Ezedin Barka, and Mohammed-Anis Benelbahri. Firewall filtering rules analysis for anomalies detection. International Journal of Security and Networks, 3(3):161–172, 2008.
[10] Chi-Shih Chao. A flexible and feasible anomaly diagnosis system for internet firewall rules. In Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific, pages 1–8. IEEE, 2011.
[11] Thawatchai Chomsiri and Chotipat Pornavalai. Firewall rules analysis. In Security and management, pages 213–219, 2006.
[12] Errin W Fulp. Optimization of network firewall policies using ordered sets and directed acyclical graphs. In Proc. of IEEE Internet Management Conference, 2005.
[13] Korosh Golnabi, Richard K Min, Latifur Khan, and Ehab Al-Shaer. Analysis of firewall policy rules using data mining techniques. In Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP, pages 305–315. IEEE, 2006.
[14] Xiangjian He, Thawatchai Chomsiri, Priyadarsi Nanda, and Zhiyuan Tan. Improving cloud network security using the tree-rule firewall. Future generation computer systems, 30:116–126, 2014.
[15] Hongxin Hu, Gail-Joon Ahn, and Ketan Kulkarni. Fame: a firewall anomaly management environment. In Proceedings of the 3rd ACM workshop on Assurable and usable security configuration, pages 17–26. ACM, 2010.
[16] Hongxin Hu, Gail-Joon Ahn, and Ketan Kulkarni. Detecting and resolving firewall policy anomalies. IEEE Transactions on dependable and secure computing, 9(3):318–331, 2012.
[17] Tihomir Katic and Predrag Pale. Optimization of firewall rules. In Information Technology Interfaces, 2007. ITI 2007. 29th International Conference on, pages 685– 690. IEEE, 2007.
[18] Bassam Khorchani, Sylvain Hall´e, and Roger Villemaire. Firewall anomaly detection with a model checker for visibility logic. In Network Operations and Management Symposium (NOMS), 2012 IEEE, pages 466–469. IEEE, 2012.
[19] Suchart Khummanee, Atipong Khumseela, and Somnuk Puangpronpitag. Towards a new design of firewall: Anomaly elimination and fast verifying of firewall rules. In Computer Science and Software Engineering (JCSSE), 2013 10th International Joint Conference on, pages 93–98. IEEE, 2013.
[20] Alex X Liu. Formal verification of firewall policies. In Communications, 2008. ICC'08. IEEE International Conference on, pages 1494–1498. IEEE, 2008.
[21] Naveen Mukkapati and Ch V Bhargavi. Detecting policy anomalies in firewalls by relational algebra and raining 2d-box model. International Journal of Computer Science and Network Security (IJCSNS), 13(5):94, 2013.
[22] Lihua Yuan, Hao Chen, Jianning Mai, Chen-Nee Chuah, Zhendong Su, and Prasant Mohapatra. Fireman: A toolkit for firewall modeling and analysis. In Security and Privacy, 2006 IEEE Symposium on, pages 15–pp. IEEE, 2006.