CLOUD FIREWALL RULE INCONSISTENCIES RESOLUTION AND TIME ANALYSES

Mrs. Dhwani Rajkumar Hakani[1], Dr. Palvinder Singh Mann[2]*

Abstract

Firewalls are utilised in a variety of applications and are created using cutting-edge security strategies. In the early phases of internet technology, networks require specific security measures, particularly in a client-server paradigm, which is a fundamental architecture in contemporary computing. In this domain, firewalls are already providing security across networks of various complexity. Firewalls are well-known for monitoring traffic and reducing device vulnerabilities. The most critical network security component was the firewall. Each user that talks with their system need time to arrive. As a result, distinguishing which users are genuine among them and whose network interactions are secure, whether local or global, is wasteful. The firewall is one of the most important and often used functions. We are addressing these issues by building a firewall on a private network to offer adequate user analysis. This helps distinguish between normal and abnormal user behaviour, allowing harmful actions to be avoided.We are comparing execution time of firewall in different Operating systems. The efficiency of the firewall's security protection is determined by the quality of the policy provided in the firewall. The most important aspect is detecting and resolving network issues. This method might be used to avoid irreversible losses, strengthen company security, and provide secure access.

Keywords: Rule Engine, Rule Reordering, Shadowing, Redundancy, Policy Conflict, Correlation, Policy Resolution.

[1]Research Scholar, Gujarat Technological University, Ahmedabad, Gujarat, India,
Email: adf_dhwani@gtu.edu.in
[2]*Associate Professor, Gujarat Technological University, Ahmedabad, Gujarat, India,
Email: asso_psmaan@gtu.edu.in

***Corresponding Author: -** Dr. Palvinder Singh Mann
*Associate Professor, Gujarat Technological University, Ahmedabad, Gujarat, India,
Email: asso_psmaan@gtu.edu.in

DOI: 10.53555/ecb/2022.11.10.44

I. INTRODUCTION

Firewalls have been widely employed in fighting suspicious traffic and unauthorised access to Internet-based organisations as one of the fundamental aspects in network and information security. A firewall, which sits across an internal network and the open Internet, checks all packets that enter and leave the network based on security standards. A novel anomaly management framework for firewalls based on a rule-based segmentation technique is represented to facilitate not only more accurate anomaly detection but also effective anomaly resolution. The network packet space specified by a firewall's rules can be partitioned into a collection of distinct packet space segments using this approach. Each segment corresponds to a distinct set of rules for the firewall and appropriately displays an overlap relationship (either conflicting or duplicate) between those rules. We also present a flexible dispute resolution technique that allows for extremely fine dispute resolution with the assistance of numerous successful resolution strategies in terms of risk assessment of shielded networks and policy purpose.

When a confidential and a third-party network intersect, a firewall is typically established to guarantee that the firewall can evaluate all inbound and outbound traffic and choose whether to accept or refuse it based on regulations.

This work defines the anomaly policy and constructs the firewall tree to find the anomaly rule within the firewall. The following anomalous rules are detected in this paper: redundancy, shadowing, correlation, and generalisation. The following are the main contributions of this study:

• The firewall policy decision tree is established, which is a comparable policy refinement. To locate the anomalous rules, the firewall tree is employed.

• The created firewall tree is used to extract the sub-tree path, which is utilised to identify the redundant, shadowing, correlation, and generalisation anomalous rule.

• Many rules are gathered from several VM on a single host. Cloudsim is used to implement and test the firewall tree-based anomaly detection.

II. RELATED WORK

While different network functions have different behaviours and requirements, there is something in common when it comes to evaluating their performance. RFC 2544 [3] specifies benchmarking criteria for network devices such routers, switches, and firewalls. These recommendations cover important performance metrics for various DuTs in addition to the technique for assessing these devices' latency and throughput. Several more documents were issued to account for the expanded set of characteristics and abilities of network components [1,2]. . Most of the firewall policy visualization systems have used grid or Venn diagram method for expressing the anomalies they have[17] it requires the advanced level of proficiency and time-consuming analysis to the network administra- tors, even though recent studies have been trying to resolve errors among polices and optimize a performance of firewall [18] Classifying firewall data using ML and DL methods[19] Analyze firewall packet features using ML methods[20].Additional issues must be considered when trying to assess the effectiveness of virtualized network functions running on commodity hardware. Because of the additional abstract layer given by network function softwarization, system performance is dependent not only on the hardware that underlies it but also on the specific VNF implementation [5]. Unlike ASIC-based packet processing, issues such as scheduling and caching have an effect on the predictability and dependability of software solutions. Furthermore, dependencies among VNF instances operating on the same actual substrate might have an impact on system behaviour [6,7]. Security function chaining [4] presents a technique for characterising firewall performance and describing formats for delivering benchmarking results in the specific context of firewall benchmarking. The measurements in the rest of this article are based on the Request as a guideline.

Table 1. Comparison of related works

Ref	Approach	Advantages / Limitations
[20]	Enhanced firewall rule management	These techniques did not completely consider the firewall's properties and examined existing static firewall rules.
[10]	Decision support system-based probability method	83% accuracy was attained when eliminating abnormalities.
[11]	Optimized resolution strategy	New abnormalities cannot predicted.
[12]	Intra and Inter firewall anomaly classification	It has a high degree of accuracy and efficiency in detecting shadowing and spuriousness anomalies.
[13]	Firewall anomaly detection framework	Network administrators need to have a high level of ability and conduct time-consuming analyses.

[14]	Double Decision Tree	It is unable to manage distributed firewall discrepancies.
[15]	Bit vector-based approach	Reduces time complexity
[16]	Knowledge graph-based firewall anomaly detection	There is no implementation and the results
[18]	Hierarchical visualization-based approach	The firewall administrators don't need to have advanced knowledge to manage it.
[19]	Visualization tool	Need to improve the functionality.

III. PROPOSED ARCHITECTURE

The disadvantage of the present system is overcome in our system. It contains new capabilities such as easy access to, management of, detection of, reordering, and resolution of rules for firewalls within the rule engine. It is advantageous to administrators and service providers. It is possible to establish personal incoming and outgoing rules utilising contemporary technology by applying network segmentation, detecting connected rules, and rearranging the previous rule. Detecting incoming traffic is a novel and strong tool for limiting fraudulent activity. This aids in informing administrative authorities of the harmful conduct. This project proposes a revolutionary firewall anomaly management system that utilises a rule-based segmentation method that enables not only more precise anomaly identification but also efficient anomaly resolution.

A. Features:

• Policy anomalies are easily understood thanks to the grid-like depiction.
• Capable of precisely identifying all rules involved in a policy anomaly.
• A firewall provides safe and trustworthy access.
• It is simple to identify and rearrange specified rules.
• Performs an anomaly analysis on both the previous and following rules.
• Giving us the ability to set incoming and outbound rules.

B. Applications:

• Detect unauthorised users or harmful data using a rule engine.
• Firewall policy analysis simplifies the examination of secure network communication.
• Use firewall policy analysis to reduce cybercrime.
• Provides security for both public and private networks
• Recognise user behaviours.
• Using the rule engine, we can simply reorganise the rules, and then use this reordering to create our own fresh rule list.
• We can simply block the unauthorised user using firewall policy analysis.

IV. FIREWALL APPLICATIONS

Firewalls can be utilised in both business and consumer contexts.

- Firewalls can include an information security and event handling strategy (SIEM) into current cybersecurity equipment and are put at the network's perimeter of organisations to protect against both external and inner threats.
- Firewalls may track and audit by discovering trends and refining policies by upgrading them to guard against urgent threats.
- Firewalls can be used to protect a home network, DSL, or cable modem using static IP addresses. Firewalls may simply filter traffic and alert users to intrusions.
- They are also utilised in antivirus software.
- When vendors identify new threats or fixes, firewall rule sets are updated to address the vendor concerns.

V. DIFFERENT TYPES OF FIREWALL POLICIES

1) FIREWALL POLICY TYPE 1

Order	Rule	Protocol	Src IP	Src Port	Dest IP	Dest Port	Action
1	R1	TCP	140.192.37.2	Any	161.120.33.40-46	25	Deny
2	R2	TCP	140.192.34.*	Any	192.168.1.12	80	Allow
3	R3	UDP	10.1.2.*	Any	172.32.1.*	53	Deny
4	R4	UDP	10.1.*.*	Any	172.32.1.*	53	Deny
5	R5	TCP	192.168.1.1	Any	192.168.33.40	80	Allow

Table 2: Firewall policy 1

2) FIREWALL POLICY TYPE 2

Protocol	SIP	SPort	DIP	DPort	Action
TCP	[0-41]	[2-14]	[0-8]	[22-40]	Allow
TCP	[0-12]	[2-10]	[0-34]	[0-80]	Deny
TCP	[20-48]	[10-24]	[2-16]	[0-80]	Allow
TCP	[0-74]	[0-50]	[0-15]	[100-139]	Allow
TCP	[10-30]	[2-8]	[0-17]	[0-35]	Allow

Table 3: Firewall Policy 2

Firewall Policy Advisor

order	protocol	src_ip	src_port	dst_ip	dst_port	action
1:	tcp,	140.192.37.20,	any,	*.*.*.*,	80,	deny
2:	tcp,	140.192.37.*,	any,	*.*.*.*,	80,	accept
3:	tcp,	*.*.*.*,	any,	161.120.33.40,	80,	accept
4:	tcp,	140.192.37.*,	any,	161.120.33.40,	80,	deny
5:	tcp,	140.192.37.30,	any,	*.*.*.*,	21,	deny
6:	tcp,	140.192.37.*,	any,	*.*.*.*,	21,	accept
7:	tcp,	140.192.37.*,	any,	161.120.33.40,	21,	accept
8:	tcp,	*.*.*.*,	any,	*.*.*.*,	any,	deny
9:	udp,	140.192.37.*,	any,	161.120.33.40,	53,	accept
10:	udp,	*.*.*.*,	any,	161.120.33.40,	53,	accept
11:	udp,	*.*.*.*,	any,	*.*.*.*,	any,	deny

Figure 1: Firewall Policy Rule

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>

Firewall Policy Advisor

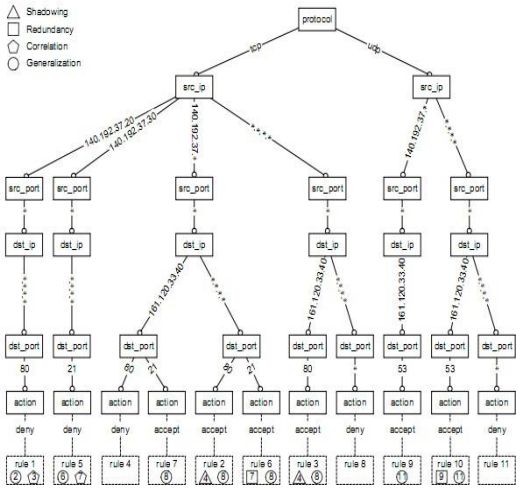

Figure 2: Firewall Tree

1) Shadowing Anomaly

The shadowing anomaly occurs when a particular rule becomes the shadow of the preceding rule, as the name indicates. When the predicate component of a single rule fits the premise part of another prior rule but the actions are different, that rule is known as the shadowing rule of that previous rule.

$$\left(\forall x, (R_a(x) \supseteq R_b(x))\right) \wedge (R_a(action) \neq R_b(action))$$

2) Correlation Anomaly

When two rules possess distinct action fields while certain predicates fields of the first rule are equal or smaller than the corresponding predication fields in the second rule and the rest of the second rule's predicate fields are superset of the first rule's predicate fields, the rules are said to be correlated.

$$\left(\exists x, (R_a(x) \supseteq R_b(x))\right) \wedge \left(\exists x', (R_a(x') \subseteq R_b(x'))\right) \wedge (R_a(action) \neq R_b(action))$$

3) Generalization Anomaly
When both rules have distinct action values but all of the second rule's predicate fields match all of the first rule's predicate fields, the second rule is referred to as the generalisation of the initial rule.
$\left(\forall x, (R_a(x) \subseteq R_b(x))\right) \wedge (R_a(action) \neq R_b(action))$

Although it resembles observing, the two are not the same. Shadowing happens when the preceding policy matches or contains a subordinate policy, as opposed to generalisation, which happens when a subsidiary policy incorporates the superordinate policy. When both rules have distinct actions and all of the packet found by one of them are a subset all the packet matched by the other, the generalisation anomaly occurs.

4) Redundancy Anomaly
A duplicate rule performs an identical action on the same packet as another rule, hence removing the redundant rule has no effect on the security policy. Rule If Rx comes before Ry in the sequence, and Ry is a portion or precisely the same of Rx, and the actions are comparable, Ry is redundant in determining Rx.
$\left(\forall x, (R_a(x) \supseteq R_b(x))\right) \wedge (R_a(action) \neq R_b(action))$

As indicated in Figure 1, rule 7 is redundant to rule 6, and rule 9 is redundant to rule 10, hence removing rule 7 and rule 9 has no influence on the resultant policy.
Redundancy is seen as a mistake. A duplicate rule might not contribute to the filtering decision, but it does increase the size of the filtering rule table and may increase search time and space needs. It is critical to identify duplicate rules before the administrator can amend or delete them entirely.

VI. CONCEPT OF PROGRAMMING

A. JAVA

Java is a tiny, simple, safe, object-oriented, interpreted or dynamically optimised, byte coded, constructional, trash gathered, multithreaded language for programming with strongly typed exception handling that allows for the creation of distributed and dynamically expandable programmes.Java is a computer language that is object oriented. Java, like C, FORTRAN, Smalltalk, Pearl, and many others, is a high-level, third-generation language. Java may be used to create computer programmes that crunch numbers, parse language, play games, store data, or perform any of the hundreds of other tasks that computer software is capable of.
It is easy to use and object-oriented it aids in the creation of user-friendly interfaces; it is extremely dynamic; and it enables multithreading.
It is platform agnostic, extremely secure, and resilient, and it enables internet programming.

B. NetBeans

The IDE NetBeans is an open-source, free, cross-platform IDE that includes Java programming language support.NetBeans is a type of IDE for creating in Java, as well as additional languages such as PHP, C or C++, and HTML5. It also serves as an application platform architecture for Java desktop apps and other applications.The NetBeans IDE is developed in Java and may be run on Windows, Mac OS X, Linux, Solaris, and other systems that support a JVM.The NetBeans Platform enables the development of applications using a collection of modular software components known as modules. Third-party developers can expand NetBeans Platform-based applications (including the NetBeans IDE).

C. Cloud Simulator

One of the most popular cloud simulators is CloudSim.
CloudSim is built on the SimJava framework and uses the robotics simulator Gazebo.
SimJava makes advantage of a continuous event simulator. It has the ability to display simulation items as blinking icons on the screen.

- CloudSim enables dynamically adding components and pausing/resuming simulations.
- CloudSim uses a collection of classes to model fundamental cloud capabilities.
- The DataCenter object is essential for simulator functioning.
- The DataCenter object creates a generic resource provisioning component that executes a set of allocation of resources policies.
- Data centre characteristics include: construction, operating system, machine list, assignment policy, period or space-shared, and resource pricing per time unit.
- The class DatacenterBroker. The broker's role is to act as a bridge among service providers and customers in the cloud.

VII. DETECTION OF ANOMALIES

A. Policy Tree Representation

To detect abnormalities within a firewall system, Al-Shaer et al. [8] presented the Policy tree depiction approach. They used a single-rooted policy tree to describe the firewall policy. They claim that their approach keeps a basic representation of the filters in a ruleset and can detect rule connections and anomalies between multiple rules. Every node in the policy tree represents a network field, and each path from the root node to the leaf nodes reflects an individual rule. The same branches of the tree will be shared by rules with an identical field value at different

nodes. The decision component of a rule is represented by a leaf node.

B. Tree-Rule Firewall

The Tree rule was proposed by Xiangjian et al[9]. This approach is quite close to the Policy Tree Representation technique presented by E Al-Shaer et al [8].The main distinction is that it isn't a strategy for detecting firewall irregularities. This is a method for configuring the rules within a firewall system to offer the network administrator with an anomaly-free environment. They believe that this strategy protects the system against rule conflicts. This approach scans the information included in a packet's header and then compares the initial attribute of the packet to the data contained in the base node of the rule tree.

The firewall then sequentially checks the various properties of the packet using a search operation against relevant routers at the appropriate levels. The root node's characteristics can be either the Source IP or Destination IP, or any other attributes appropriate for the operation. Before producing the Tree rules, the user can select any attribute for each column [9].

C. Tuple Based Approach

Benelbahri and Bohoula presented a Tuple Driven Approach for detecting abnormalities within the firewall system [10]. They have expressed the fields of the relation in the form of 4-tuple in this method for two rules Ra and Rb:

(Filed#, Ra#, Rb#, and code) Where "Field" is the filtering rule order, "Ra and Rb" are both of the initial rules, and "Code" is the connection between them code.

D. Inter-Difference Matrices (IDM) Approach

Another study conducted by Bouhoula et al[11] developed the Inter-Difference matrix technique for identifying abnormalities within a firewall system. They employed a matrix for storing the distinctions between two rules in this method. The IDM technique may be characterised as a matrix whose element indicates the difference between the components in the matrix F relating to the associated vector. An IDM vector is created for each field. The total number of IDM matrix equals the number of fields m, and each element eab reflects the disparity among the values of the filtering criteria Ra and Rb for the same sector fan and fbn.

E. Association Rule Mining (ARM)

Golnabi et al. [12] developed another method for discovering rule anomalies known as the Association Rule Mining (ARM) methodology.

Another study conducted by Bouhoula et al[11] developed the Inter-Difference matrix technique for identifying abnormalities within a firewall system. They employed a matrix for storing the distinctions between two rules in this method. The IDM technique may be characterised as a matrix whose element indicates the difference between the components in the matrix F relating to the associated vector. An IDM vector is created for each field. The total number of IDM matrix equals the amount of fields m, and every component eab reflects the disparity among the values of the filtering criteria Ra and Rb for the same sector fan and fbn.

F. XML based open tool for anomalies detection

Another solution based on XML developed by Benelbahri et al[12] is known as the XML-based open tool to discover and resolve filter rule anomalies. They presented this approach to resolve conflicts involving multiple rules, which depends on the idea of using resolution filters. They attempted to characterise all of the various scenarios that may lead to rule conflict and provided a way for resolving them using this technique.

G. Graph Based Approach

Fulp [14] presented a strategy for optimising the performance of the firewall system. This strategy does not directly address the abnormalities that exist inside a firewall system, but rather enhances the performance of the firewall system. The firewall described has certain restrictions [15]. One of them is shifting the locations of two rules, which modifies a firewall policy and causes security issues as well as a decrease in performance. There are a few strategies that may be utilised to organise the rules in an organised manner so that they can be correctly listed. The approach uses ordered sets as well as directed acyclic graphs to organise the rules in a manner that is linear in order to increase performance.

H. Relational Algebra and Raining 2D-Box Model

This is another approach proposed by Mukkapati and Bhargavi [16] 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.

VIII. EXPERIMENTAL RESULTS

This section examines the performance of the recommended intra-firewall rule anomaly detection, as well as implementation details. It is implemented using the Java platform, and cloudsim (a cloud simulator) constructs and manages one host and virtual machines. Windows 10 64-bit, an Intel Pentium 2.30 GHz CPU, and 4 GB of RAM are included in the system setup. Figure 3 depicts a Host and VM deployment scenario.

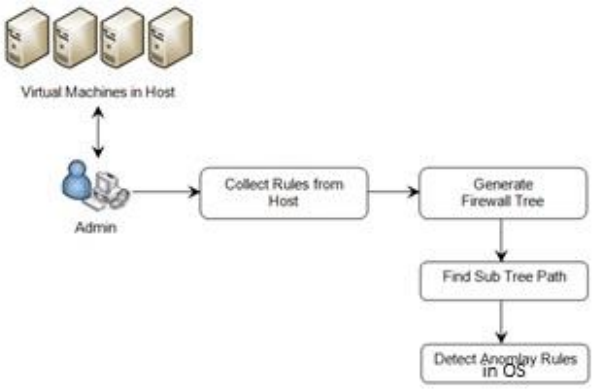

Figure 3: Proposed Architecture.

This study proposes a firewall tree-based technique for identifying abnormalities. The important steps are as follows:

When building a tree, the policy is translated into an analogous firewall tree using a comparable firewall tree generation mechanism, and the overlapping sections of the rules are recorded. The comparable tree is used to establish a route within a subtree. The corresponding anomalous rules are identified based upon the subtree path.
The administrator can detect traffic packets that meet a number of criteria. The report will specify which policy determines whether data packets are allowed or refused. When a new rule is added, the administrator may compare it to the associated tree to determine if it varies from the previously set policy.

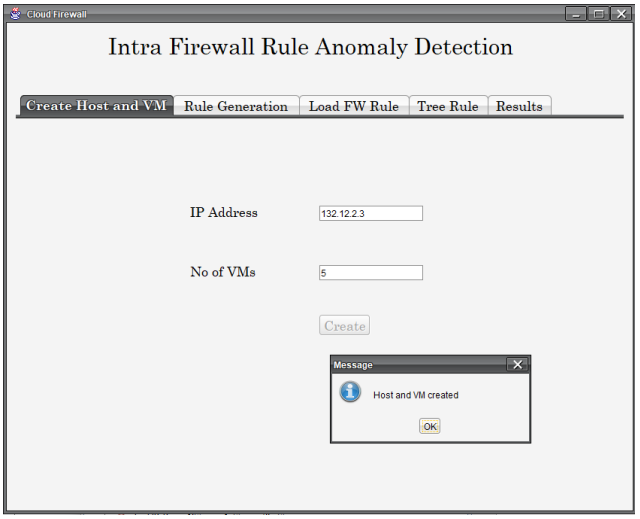

Figure 4: Windows Host and VM created

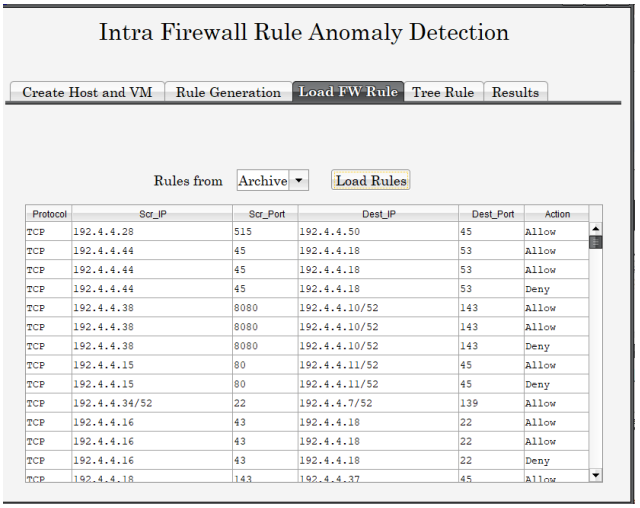

Figure 5: Firewall Rules are loaded

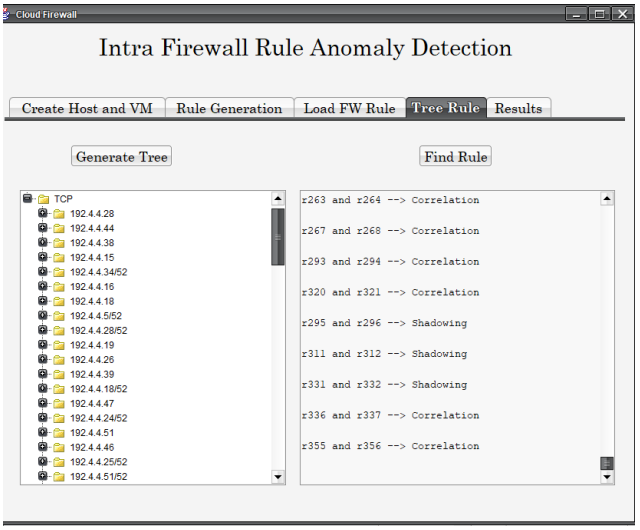

Figure 6: Tree Rule Firewall

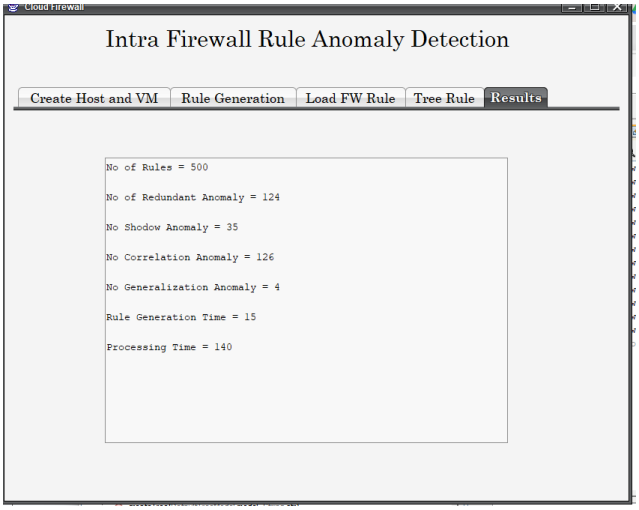

Figure 7: Firewall Execution time in Windows platform

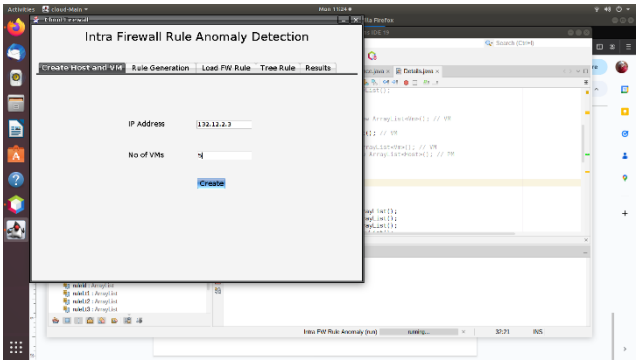

Figure 8: Ubuntu/Linux Implementation

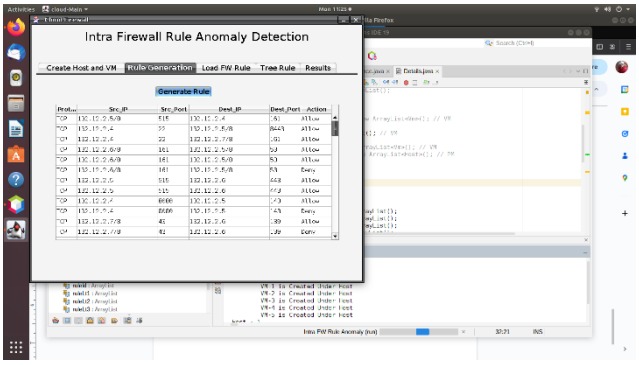

Figure 9: Rule Generation

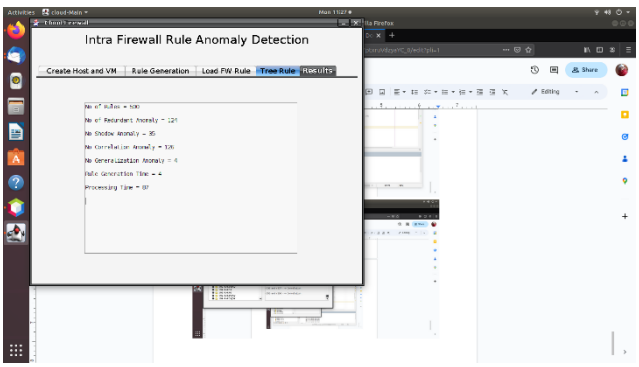

Figure 10: Processing Time

IX. CONCLUSION

To provide efficient security services, firewalls must be properly managed. Network vulnerabilities caused by firewall rule complexity and rule abnormalities, on the other hand, may render the network hazardous. The anomaly finding technique is used in this study to uncover discrepancies in a firewall policy. This study creates a firewall tree model in order to detect firewall anomalies. To uncover and simplify the policy, the administrator might utilise the firewall anomaly identification technique. This investigation will look for rule irregularities in a certain host firewall. We discovered that the Linux System firewall operates more quickly than the Windows firewall.

Declarations

The authors state that they do not have any known conflicting business interests or personal ties that may seem to have influenced the work disclosed in this study. The authors declare their subsequent financial interests and personal ties as potential conflicting interests.

Conflicts of Interests

There are no conflicts of interest.

Funding

There was no financing from anybody. This research project is in the No Funding Zone. The research expenditures have been entirely borne by the authors and co-authors.

Participant consent and ethics approval

This work may be published in this journal with the complete permission of the authors and co-authors.

Consent for publication

This paper can be published in this journal with full consent of Authors and Co-Auhtors

Competing interests

There are no potential conflicts of interest.

Acknowledgements

First and foremost, I'd want to thank Almighty for all of His kind favours, without which my labour would have been a shambles. I'd also want to thank those who remember me in their prayers for my accomplishments.

Data Availability Statement

Datasets are created by own

REFERENCES

[1] Asati, R., Pignataro, C., Calabria, F., Olvera, C.: RFC26201: Device Reset Char- acterization. IETF (2011)

[2] Bradner, S., Dubray, K., McQuaid, J., Morton, A.: RFC6815: Applicability State- ment for RFC 2544: Use on Production Networks Considered Harmful. IETF (2012).
[3] Bradner, S., McQuaid, J.: RFC2544: Benchmarking Methodology for Network Inter- connect Devices. IETF
[4] G. Li, H. Zhou, B. Feng, G. Li, H. Zhang, and T. Hu. Rule anomaly-free mechanism of security function chaining in 5g. IEEE Access, Vol. 6, pp. 13653-13662 (2018)
[5] Lange, S., Nguyen-Ngoc, A., Gebert, S., et al.: Performance benchmarking of a software-based LTE SGW. In: 2nd International Workshop on Management of SDN and NFV Systems (2015)
[6] Morton, A.: Considerations for Benchmarking Virtual Network Functions and Their Infrastructure. Internet-Draft draft-morton-bmwg-virtual-net-03 (2015)
[7] Overture, Brocade: Intel, Spirent, and Integra. NFV Performance Benchmarking for vCPE, Executive Summary (2015)
[8] 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).
[9] 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).
[10] S. Khummanee, P. Chomphuwiset, and P. Pruksasr. DSSF: Decision Support System to Detect and Solve Firewall Rule Anomalies based on a Probability Approach. ECTI Transactions on Computer and Information Technology (ECTI-CIT), Vol. 16, No. 1, pp. 56-73 (2022).
[11] F. Valenza, and M. Cheminod. An Optimized Firewall Anomaly Resolution. Journal of Internet Services and Information Security, Vol. 10, No. 1, pp. 22-37(2020)
[12] YZ. Cheng, and Q.Y. Shi. Analysis of policy anomalies in distributed firewalls. International Journal of Network Security, Vol. 24, No. 4, pp. 617-627(2022).
[13] C. Togay, A. Kasif, C. Catal and B. Tekinerdogan.A Firewall Policy Anomaly Detection Framework for Reliable Network Security. IEEE Transactions on Reliability, vol. 71, no. 1, pp. 339-347(2022).
[14] Z. Lin, and Z. Yao. Firewall Anomaly Detection Based on Double Decision Tree. Symmetry, Vol. 14, No. 12 (2022)
[15] C.S. Chao, and S.J. Yang. A Bit Vector-Based Diagnosis Mechanism for Firewall Rule Anomalies in IPv6 Networking Environment. Journal of Internet Technology, Vol. 22, No. 4, pp. 867-876(2021).
[16] T. Kim, T. Kwon, J. Lee, and J. Song. F/wvis: hierarchical visual approach for effective optimization of firewall policy. IEEE Access, Vol. 9, pp. 105989-106004 (2021)
[17] H. Lee, S. Lee, K. Kim, and H.K. Kim. HSViz: Hierarchy simplified visualizations for firewall policy analysis. IEEE Access, 9, pp. 71737-71753 (2021)
[18] T. Kim, T. Kwon, J. Lee, and J. Song. F/wvis: hierarchical visual approach for effective optimization of firewall policy. IEEE Access, Vol. 9, pp. 105989-106004. (2021).
[19] Aljabri, M., Alahmadi, A.A., Mohammad, R.M.A., Aboulnour, M., Alomari, D.M., & Almotiri. S.H..Classification of Firewall Log Data Using Multiclass Machine Learning Models. Electronics. (2022).
[20] Al-Haijaa, Q.A., & Ishtaiwia, A.Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense. International Journal on Advanced Science, Engineering and Information Technology, 11(4), 1688-1695. (2021)