WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

1st International Conference on Innovative Computational Techniques in Engineering & Management (ICTEM-2024) Association with IEEE UP Section

Friendly Firewall Security System

Neelaksh Sheel [1)], Abhay Rastogi [2)], Ayushmaan Agrawal [3)], Anurag Saini [4)], Ansh Kumar Gupta [5)]

[1]Associate Professor, CS&E Department, Moradabad Institute of Technology, Moradabad, India
[2,3,4,5]B. Tech Scholar's 4th Year, CS&E Department, Moradabad Institute of Technology, Moradabad, India
[1, 2, 3, 4, 5,] sheelneelaksh@gmail.com, rastogiabhay35@gmail.com, agarwalayushmaan88@gmail.com , anusaini80861@gmail.com, kumarguptansh0@gmail.com
DOI: https://doi.org/10.55248/gengpi.6.sp525.1916

Abstract

The increasing reliance on digital systems has made networks a critical asset in modern infrastructure, driving the need for robust and adaptive security mechanisms. This research focuses on the design and implementation of a modular network firewall to address the evolving landscape of cyber threats. Unlike traditional firewalls, which often operate with static rules and limited scope, this modular approach integrates multiple advanced features to ensure comprehensive protection. Key modules include mechanisms for detecting and mitigating web-based attacks, port and protocol filtering, real-time logging and monitoring, brute-force attack prevention, and dynamic access control. The firewall also incorporates a proactive Intrusion Prevention System (IPS) and Intrusion Detection System (IDS) to identify and neutralize potential threats before they can compromise network integrity. A default-deny policy is implemented to restrict access by default, further reducing vulnerabilities.

The study emphasizes modularity, allowing each component to function independently and adapt to specific network requirements. By leveraging technologies such as signature-based detection, anomaly detection, and real-time monitoring tools, the proposed system achieves a high detection rate with minimal performance trade-offs.

This research contributes to the field by demonstrating a scalable, efficient, and adaptable firewall framework suitable for enterprise and small-scale networks. Through rigorous testing and performance evaluation, the system's effectiveness in mitigating threats and enhancing overall network security is validated. This work provides a blueprint for the development of next-generation firewalls that balance security, performance, and adaptability.

Keywords: firewall, security management, security policy, policy conflict, cyber threats.

1. Introduction

In today's digital era, where critical infrastructure relies heavily on interconnected systems, ensuring the security of networks has become a top priority. Cyber threats are evolving rapidly, posing significant risks to sensitive data and network integrity. Firewalls serve as a fundamental defense mechanism, playing a vital role in safeguarding networks by filtering malicious traffic and enforcing strict security policies. However, traditional firewalls often fall short in addressing the dynamic nature of modern cyber threats, leaving networks vulnerable to sophisticated attacks.

Despite advancements in cybersecurity measures, networks remain susceptible to a wide range of vulnerabilities, including web-based exploits, brute-force attacks, and unauthorized access. A common limitation of existing solutions is their lack of modularity, which restricts their adaptability to meet the diverse and ever-changing security requirements of different organizations.

This research seeks to bridge these gaps by designing and implementing a modular firewall system that can cater to varied network security needs. The proposed firewall integrates features such as attack detection, traffic filtering, and access control to provide a comprehensive security framework. By combining robust protection

with efficient performance, this modular approach aims to strengthen network defenses while maintaining operational efficiency.
The study is particularly focused on developing a customizable firewall solution that can be effectively deployed in both enterprise and small-scale network environments. With capabilities like real-time monitoring, logging, and proactive threat mitigation, this research sets the foundation for

creating scalable and adaptive firewall systems to address the challenges posed by modern cybersecurity threats.

2. Firewall Technology

A host-based firewall is a type of firewall that is installed and operates on an individual computer or device, rather than on a network gateway or perimeter. It is designed to monitor and control incoming and outgoing network traffic for that specific host based on predefined security rules. Host-based firewalls are often integrated into the operating system or implemented as standalone software.

Key Features:

1. Traffic Filtering: Host-based firewalls inspect and filter network traffic at the host level, allowing or blocking traffic based on rules such as IP addresses, ports, or protocols.
2. Application Control: These firewalls can control network activity by specific applications, restricting unauthorized or potentially malicious software from communicating over the network.
3. Personalization: Rules and settings can be customized for the specific needs of the host.
4. User-Friendly Interfaces: Many host-based firewalls provide graphical interfaces for easier management by end-users.
5. Intrusion Detection: Some advanced host-based firewalls incorporate intrusion detection capabilities to monitor and respond to suspicious activity.
6. Log and Alerting: They maintain logs of network activity and generate alerts for suspicious or unauthorized traffic.

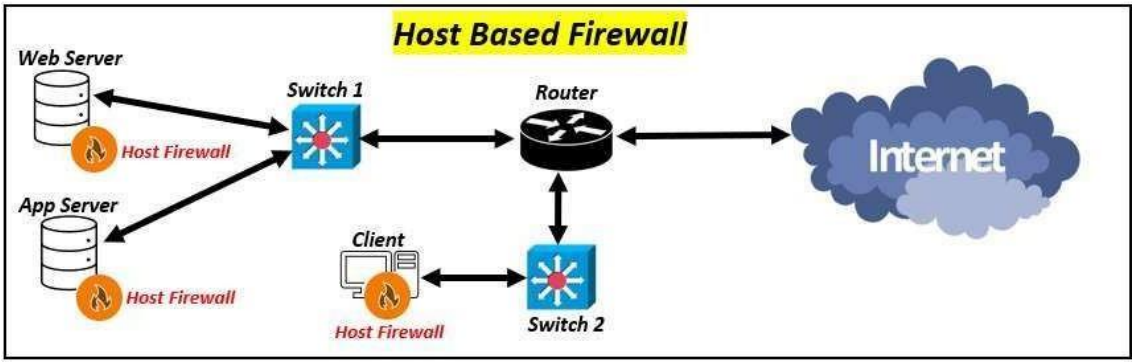

Figure 1. Diagram of the location of the firewall

Filtering Rule Format. It is possible to use any field in IP, UDP or TCP headers in the rule filtering part, however, practical experience shows that the most commonly used matching fields are: protocol type, source IP address, source port, destination IP address and destination port. Some other fields, like TTL and TCP flags, are occasionally used for specific filtering purposes. 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>

In this paper, we refer the order of the rule determines its position relative to other filtering rules. The protocol specifies the transport protocol of the packet, and can be one of these values: IP, TCP or UDP. The src_ip and dst_ip specify the IP addresses of the source and destination of the packet respectively. The IP address can be a host (e.g., 140.192.37.120), or a network address (e.g.,140.192.37.∗). The src_port and dst_port fields specify the port address of the source and destination of the packet respectively. The port can be either a single specific port number, or any port number indicated by “any”. As an example, the following security policy is to block all TCP traffic coming from the network 140.192.37.∗ except HTTP:

1: tcp, 140.192.37.∗, any, ∗.∗.∗.∗, 80, accept 2: tcp, 140.192.37.∗, any, ∗.∗.∗.∗, any, deny

3. The Main Functions of Firewall

In this section, we provide a detailed breakdown of the key features and functionalities of each module in the system. Each module is designed to perform specific tasks to ensure the overall system operates efficiently and meets the requirements.

3.1 Detecting Web Attacks

Web applications are increasingly targeted by cyber attackers due to their widespread deployment and the valuable data they handle. Common attacks such as SQL injection, Cross-Site Scripting (XSS), and remote code execution exploit vulnerabilities in web applications to steal sensitive information or compromise the system. To defend against these threats, a Web Application Firewall (WAF) can be implemented to monitor and filter HTTP requests. The WAF inspects incoming traffic for malicious patterns and prevents unauthorized access to web servers.

In addition to WAF protection, rule-based filtering techniques can be used to identify suspicious queries that match known attack patterns. By examining the structure of incoming HTTP requests, the system can detect anomalies, such as attempts to inject malicious SQL code or cross-site scripting payloads. Integrating behavioural analytics helps further strengthen security by identifying unusual user activities or deviations from established norms. This combination of preventive measures creates a multi-layered defence that proactively identifies and mitigates web application

attacks.

3.2 Port and Protocol Filtering

Open ports and unnecessary protocols on a network present significant security risk. Attackers often target these exposed access points to gain unauthorized access or launch attacks. By identifying and limiting the services that require network access, organizations can minimize their attack surface. A critical step in network security is ensuring that only the necessary ports are open, and that traffic is allowed exclusively through those ports. Regular auditing of open ports helps assess potential risks, ensuring that vulnerabilities are discovered and mitigated in time.

To further reduce the attack surface, dynamic filtering of protocols is essential. Protocols that are either outdated or not required for the network's operation should be blocked to prevent exploitation. This process ensures that only relevant traffic is allowed to flow through the network, effectively reducing the chances of an attacker using unnecessary protocols as entry points. By proactively managing and monitoring ports and protocols, organizations can significantly improve their overall network security.

3.3 Logging and Monitoring

Effective logging and monitoring are fundamental to maintaining network security and ensuring compliance with regulatory requirements. By capturing detailed logs of all incoming and outgoing traffic, organizations can gain insights into the source and destination of data transfers, helping to identify potential threats or unusual activity. Real-time log monitoring tools provide administrators with the ability to track network activities and detect any suspicious patterns that could indicate an attack in progress.

Automating incident response based on predefined log anomalies enhances the efficiency of security operations. Once an anomaly is detected, predefined triggers can initiate immediate actions, such as blocking traffic or alerting the security team. These automated responses allow for quicker mitigation of threats before they can escalate into serious security breaches. By continuously logging and analysing network traffic, organizations can strengthen their security posture and ensure a timely response to emerging threats.

3.4 Brute force Attack Mitigation

Brute force attacks are one of the most common methods used by attackers to gain unauthorized access to systems by systematically attempting different passwords or encryption keys. To mitigate the risk of brute force attacks, continuous monitoring of login attempts is essential. This monitoring identifies repeated failed login attempts from specific IP addresses, signalling potential attack attempts. Once a threshold for failed login attempts is reached, the system can automatically block the originating IP address, preventing further access attempts.

This automated blocking mechanism helps reduce the likelihood of a successful brute force attack. Additionally, administrators can receive alerts whenever an IP address is blocked, allowing them to investigate and determine whether the attack is part of a larger campaign. By implementing brute force attack mitigation techniques, organizations can protect sensitive resources and ensure that unauthorized access is swiftly blocked before it causes harm.

3.5 Network Traffic Filtering

Network traffic filtering is a crucial aspect of proactive threat detection. By continuously monitoring and inspecting all incoming and outgoing traffic, security teams can identify known malicious signatures and block harmful data from reaching the network. This process helps protect against a wide range of network-based attacks, including viruses, malware, and denial-of-service (DoS) attempts. Deep Packet Inspection (DPI) is an essential technique in this regard, as it enables the examination of data payloads within packets, allowing for a more thorough detection of hidden threats.
IP reputation, geolocation, and known threat lists can be used to filter traffic based on the perceived risk of the source.

By analysing traffic against these criteria, organizations can identify and block traffic from known malicious sources, preventing attacks before they impact the network. With continuous monitoring and real-time filtering of network traffic, organizations can stay ahead of potential threats and safeguard their infrastructure.

3.6 Intrusion Prevention System (IPS)

An Intrusion Prevention System (IPS) is a critical component of any network security strategy, actively identifying and blocking threats in real time. The IPS uses a combination of signature-based detection, which relies on predefined patterns of known threats, and anomaly-based detection, which identifies deviations from normal network behaviour. This dual approach enables the IPS to detect both known attacks and novel threats that may not yet be classified.

Once a threat is detected, the IPS immediately takes action to block the attack before it can exploit any vulnerabilities within the system. By preventing protocol violations and suspicious activities, the IPS helps protect the network from a wide range of cyber threats. Detailed logs and reports generated by the IPS provide valuable insights for security teams, helping them refine their security policies and stay prepared for future threats.

3.7 Access Control Policies

Access control is an essential aspect of network security, ensuring that only authorized users can access sensitive resources. Implementing Role-Based Access Control (RBAC) allows administrators to assign permissions based on the specific roles of users within the organization. This ensures that users have access only to the resources necessary for their work, minimizing the potential for misuse or accidental exposure of sensitive data.

Secure authentication protocols, such as OAuth and LDAP, further strengthen access control by verifying the identity of users before granting access. The principle of least privilege is also enforced, meaning that users are only granted the minimum permissions required to perform their tasks. By implementing strict access control policies, organizations can ensure that sensitive resources are protected from unauthorized access, thereby reducing the risk of data breaches and security incidents.

3.8 Intrusion Detection System (IDS)

An Intrusion Detection System (IDS) is designed to monitor network traffic for signs of suspicious or malicious activity. By establishing a baseline of normal network behaviour, the IDS can identify deviations that may indicate an intrusion or policy violation. Once suspicious activity is detected, the IDS sends an alert to network administrators, enabling them to take appropriate action to investigate and mitigate the threat.

In addition to providing real-time alerts, the IDS logs detected incidents for later analysis and forensic investigation. This allows security teams to review the events leading up to the intrusion and better understand how the attack was carried out. By combining real-time monitoring with historical data, the IDS helps organizations stay vigilant against threats and continuously improve their security measures.

3.9 Default Deny Policy

A Default Deny Policy is a powerful security measure that strengthens a network's defense by blocking all incoming and outgoing traffic unless it is explicitly permitted. This "deny by default" rule ensures that only trusted IPs, protocols, and ports are allowed to communicate with the network, significantly reducing the attack surface. By requiring administrators to whitelist only trusted sources, the policy helps prevent unauthorized access and accidental exposure of network resources.

The default deny policy also makes it easier to manage security by focusing on exceptions rather than inclusions. This approach simplifies rule management, as only specific traffic needs to be explicitly allowed. By adopting this security model, organizations can minimize the risk of attacks and ensure that all network traffic is thoroughly vetted before being permitted to pass through the firewall.

4. Firewall Policy Modelling

As a basic requirement for any firewall policy management solution, we first modeled the relations and the representation of firewall rules in the policy. This model is complete (i.e., means includes all rules possible in any firewall policy) and efficient (i.e., means easy to implement and easy to use). Rule relation modeling is necessary for analyzing firewall policy and designing management techniques such as conflict detection and rules editing. The rules or policy representation modeling is important for implementing these management techniques and visualizing the firewall policy structure. In this section, we describe formally our model of firewall rule relations and policies.

4.1 Formalization of Firewall Rule Relations

To create an effective model for filtering rules, we first need to identify all the possible relationships that could exist between two or more packet filters. In this section, we outline the various types of relationships between filtering rules and demonstrate that no other relationships exist. These relationships are determined by comparing the network fields of the filtering rules. The values of the same field in two different rules can either be equal, inclusive, or distinct. Two values are considered equal if they exactly match, inclusive if one value is a subset of the other (but not equal), and distinct if neither condition applies. A field match occurs when the values are either equal or inclusive. For example, a source address like 140.192.37.10 matches 140.192.37.* but does not match 140.192.37.20.

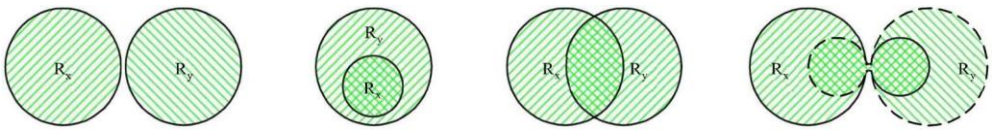

Figure 2. Relations between two filtering rules R_X and R_Y.

The diagrams shown in Figure 2 illustrate these relations between two filtering rules, R_X and R_Y. Below we give an intuition of our proof that there is no other relation between R_X and R_Y could exist.

4.2 Firewall Rule Policy Representation

Firewall rules are represented using a policy tree, a rooted structure that simplifies rule management and highlights relationships and anomalies. Each tree node represents a rule field, branches denote field values, and the root corresponds to the protocol field, with leaf nodes defining actions. Rules sharing field values follow the same branch, while subset and superset rules propagate across branches.

To insert a rule, a recursive algorithm checks for branch matches at each node. If matched, the rule is added; otherwise, a new branch is created. The process ensures relationships among rules are maintained. The focus is on clarity and simplicity, making the model effective for rule visualization and anomaly analysis, rather than optimizing search time.

5. Firewall Policy Anomaly Detection

The order of filtering rules in a firewall policy plays a critical role in determining the overall security behaviour, as the packet filtering process sequentially checks each rule until a match is found. When filtering rules are independent or completely disjoint, their ordering is not important. However, in most cases, filtering rules are interrelated, and if the rule order is not carefully managed, some rules may be inadvertently overridden by others, leading to incorrect security actions. Furthermore, with a large number of filtering rules, the chances of introducing conflicting or redundant rules increase significantly. A firewall policy anomaly occurs when two or more distinct filtering rules match the same packet. In this section, we define various types of anomalies that can arise within filtering rules of a firewall policy and outline a technique for detecting these anomalies

5.1 Firewall Policy Anomaly Types

Here, we describe and then define a number of possible firewall policy anomalies. This includes clear conflicts that cause some rules to be always suppressed by other rules, or warnings for potential conflicts between related rules.

(1) Shadowing anomaly: A rule is shadowed when a preceding rule matches all the packets that the rule would match, causing the shadowed rule to never be evaluated. If the shadowed rule is removed, the policy remains unchanged. Rule Rx is shadowed by rule Ry if Rx follows Ry in the rule order, Rx is a subset match of Ry, and the actions of Rx and Ry differ. For instance, in Figure 4, rule 4 is a special case of rule 3 with a different action, so removing rule 4 will not affect the policy. Shadowing is a significant error because the shadowed rule will never take effect, potentially causing traffic to be incorrectly blocked or permitted. It's crucial to identify shadowed rules and notify administrators to fix the issue, either by reordering or removing the shadowed rule.

(2) Correlation anomaly: Two rules are considered correlated if the first rule matches some packets that also match the second rule, and vice versa. Rule Rx and rule Ry have a correlation anomaly if they are correlated and have differing actions. In Figure 4, rule 2 is correlated with rule 3, and reversing the order of the rules would change the effect of the policy, although rule 2 wouldn't be shadowed by rule 3. Correlation anomalies are warning signs, as they indicate rules that imply an action that isn't explicitly managed. For example, if the rules accept all HTTP traffic from "140.192.37.10" to "140.192.37.*", reversing the order would deny the same traffic. Detecting such correlations helps administrators resolve conflicts by selecting the appropriate rule order that aligns with the security policy.

(3) Redundancy anomaly: A redundant rule performs the same action as another rule on the same packets, so removing the redundant rule won't change the policy's effect. Rule Rx is redundant to rule Ry if Rx is a subset match of Ry, and both rules have identical actions. In Figure 4, rule 7 is redundant to rule 8, meaning removing rule 7 has no impact on the policy. Redundancy is an error because the redundant rule doesn't contribute to the filtering decision, but it unnecessarily increases the rule table size, which may affect search times and space efficiency. Identifying redundant rules allows administrators to remove or modify them, improving the policy's efficiency.

(4) Generalization anomaly: A rule is a generalization of another rule if it matches all the packets that the second rule would match, but not vice versa. Rule Rx is a generalization of rule Ry if Rx follows Ry in the rule

order, Rx is a superset match of Ry, and the actions of Rx and Ry differ. For example, rule 2 is a generalization of rule 1 in Figure 4; if the rules are reversed, rule 1 would be shadowed by rule 2 and would no longer be effective. Generally, the superset rule should come after the subset rule. Although generalization is a warning rather than an error, it requires careful handling by administrators to ensure the correct rule order is maintained.

6. Firewall Policy Editor

Firewall policies are often created by different network administrators and may be updated periodically to address new security needs or changes in the network topology. Editing an existing policy—whether by adding, removing, or modifying rules—can be more challenging than creating a new one. Since firewall rules are typically ordered, it is crucial to insert a new rule in the correct position to avoid introducing anomalies. The same care must be taken when modifying any network field in an existing rule.

In this section, we introduce a policy editor tool designed to make rule editing much easier while preventing the introduction of anomalies during policy updates. The policy editor (1) guides the user to the correct position(s) for adding or modifying a rule, (2) displays the changes in the security policy's behaviour before and after removing a rule, and (3) provides visual aids to help users track and verify policy modifications. With this tool, administrators can insert, modify, or remove rules without needing prior knowledge of the firewall policy structure. The following sections describe

how the policy editor handles rule editing, modification, and removal.

6.1 Rule Insertion

Since the ordering of rules in the filtering rule list directly impacts the semantics of the firewall security policy, a new rule must be inserted in the proper order in the policy such that no shadowing, correlation or redundancy is created. The policy editor helps the user to determine the correct position(s) of the new rule to be inserted. It also identifies anomalies that may occur due to improper insertion of the new rule, and suggests the proper resolution.

The algorithm describes the mechanism to insert a new rule. The general idea is that the order of a new rule is determined based on its relation with other existing rules in the firewall policy. In general, a new rule should be inserted before any rule that is its superset match, and after any rule that is its subset match. The policy tree is used to keep track of the correct order of the new rule 7, and detect any potential anomalies. The algorithm is organized into two phases: the browsing phase and the insertion phase. In the browsing phase, the fields of the new rule are compared with the corresponding tree node values one at a time. If the field value of the new rule is a subset of an existing branch, then the new rule must be inserted before the minimum order of all the rules/leaves in this branch. If the field value is a superset of an existing branch, the rule must be inserted after the maximum order of all the rules in this branch. In addition, if the field value is an exact match or a subset match of a branch, evaluating the next field continues recursively by browsing through the branch sub-tree until correct position of the rule within the sub-tree is determined. Otherwise, if disjoint or superset match occurs, a branch is created for the new rule.

The algorithm enters into the insertion phase when the action field of a new rule is to be inserted. If an action branch is created for the new rule, then the rule will be inserted and assigned the order determined in the browsing phase. If there is more than one possible order for this rule, the user is asked to select an order from within a valid range of orders as determined in the browsing phase. However, if the order state of the new rule remains UNDETERMINED or it coincides with the branches for all 5-tuple fields of an existing rule, which has the same action, then policy editor ignores this new rule and prompts the user with the appropriate message. In the former case, the rule exactly matches an existing rule and considered redundant or directly conflicting depending on the action. In the latter case, the new rule is an inclusive subset match of an existing rule but they have the same action, and thus it is considered redundant rule.

6.2 Rule Removal and Modification

In general, removing a rule has much less impact on the firewall policy than insertion. A removed rule does not introduce an anomaly but it might change the policy semantics and this change should be highlighted and confirmed. To remove a rule, the user enters the rule number to retrieve the rule from the rule list and selects to

remove it. To preview the effect of rule removal, the policy editor gives a textual translation of the affected portion of the policy before and after the rule is removed. The user is able to compare and inspect the policy semantics before and after removal, and re-assure correctness of the policy changes. Modifying a rule in a firewall policy is also a critical operation. However, this editing action can be easily managed as rule removal and insertion as described before.

7. Clustering Algorithms

Clustering algorithms use the dataset to form clusters and detect intrusions. Most all of them work on the following two assumptions:

- The normal instances have similar properties and occur close together to form one single cluster while anomalies are far apart from them.
- The number of normal instances is exceedingly large than the number of anomalies i.e. normal instances constitute around 95-98 % of the total data while anomalies constitute the rest.

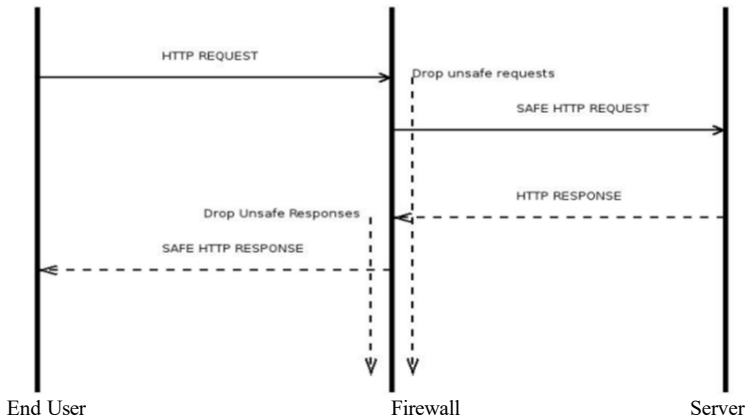

Figure 3. Process Flow

7.1 K-Means

K-Means is a typical clustering algorithm. It partitions the dataset into k clusters according to the following steps:

1. Choose k instances randomly from the dataset and make them initial centres for the clustering space.
2. Take each instance from the dataset and assign it to the closest cluster.
3. Replace each centre with the mean of its members.
4. Repeat steps 2 and 3 until there is no more updating of the centre.

The difference it has with the Leonid Portnoy algorithm is choosing of k, the initial number of clusters, in the first case and W, the cluster width, in the latter one. When applied on the KDD-99 dataset, K-means gives a result of around 80%. As already stated, our dataset consists of a set of 18 vectors which we get from the contents of a HTTP packet. Some of the vectors are protocol, source IP address, destination IP address, source port number, destination port number, method, host, payload size, response status and time. In our assumption, we have mentioned that a large number of instances are normal while the rest are anomalies. So, we label some percentage N of the clusters containing the largest number of instances as normal, rest being anomalies. After having trained the dataset by our algorithms, a new instance is put into some cluster using the formulae and the cluster is checked against a normal or an anomalous one. A voting mechanism is used and if both the algorithms classify the new instance as an anomaly, then only we consider it as an anomaly otherwise not.

8. Conclusions

Computer networks are highly vulnerable to a wide range of cyberattacks, which can come from various sources, such as hackers, inexperienced users, dishonest vendors, or even disgruntled employees. These attacks do not always originate from external parties; internal issues like weak information security or poor policies and procedures can also lead to vulnerabilities. Additionally, new security risks can emerge due to evolving attack techniques or newly discovered flaws in existing software and hardware. Some common types of attacks include social engineering, war dialling, denial-of-service (DoS) attacks, protocol-based attacks, host-based attacks, password guessing, eavesdropping, and others. These threats, including backdoors, brute-force attacks, exploiting known vulnerabilities, sniffing, spoofing, Trojan horses, viruses, impersonation, and transitive trust, can bypass conventional firewalls and cause significant harm to both individual systems and entire networks. To mitigate these risks and prevent serious consequences, Distributed Firewalls are employed. A Distributed Firewall is designed to enforce a network's security policies through a centralized management system, using a policy language and distribution scheme. It enables efficient control of the security policies while using certificates to identify every member of the network domain. This firewall solution helps protect critical network endpoints, which are the primary targets of hackers. By filtering traffic from both internal and external sources, Distributed Firewalls provide greater scalability and eliminate the single point of failure associated with traditional perimeter firewalls, offering a more robust defence against cyber threats.

Acknowledgements

We would like to thank Moradabad Institute of Technology for their support in developing this project. Special thanks to Dr. Neelaksh Sheel for their guidance throughout the research and development process.

References:

[1] M. Cheminod, L. Durante, L. Seno and A. Valenzano, "Performance evaluation and modeling of an industrial application-layer firewall", IEEE Trans. Ind. Informat., vol. 14, no. 5, pp. 2159-2170, May 2018.

[2] K. Scarfone and P. Hofman, "Guidelines on firewalls and firewall policy", Sep. 2009.

[3] W. Cheswick and S. Belovin. Firewalls and Internet Security. Addison-Wesley, 2015.

[4]] Gaud, J.V., & Bartere, M.M. (2014). Data Security Based on LAN Using Distributed Firewall. International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 3, Issue. 3, March 2014, ISSN pp: 386391.Retrieved from IJCSMC official website:www.ijcsmc.com

[5] A. El-Atawy, E. Al-Shaer, T. Tran, and R. Boutaba, "Adaptive early packet filtering for defending firewalls against DoS attacks," in Proc. Int. Conf. on Computer Communications, 2009, pp. 2437–2445..

[6] E. Dadzie and A. Veselý, "User perception of security on social networking sites using fuzzy logic," British Journal of Applied Science and Technology, vol. 3, no. 4, pp. 714-734, 2013.

[7] Z. Fu, F. Wu, h. Huang, K. Loh, F. Gong, I. Baldine and C. Xu. "IPSec/VPN Security Policy: Correctness, Conflict Detection and Resolution." In Proceedings of Policy'2001 Workshop, January 2011.

[8] J. Guttman. "Filtering Posture: Local Enforcement for Global Policies." In Proceedings of 1997 IEEE Symposium on security and Privacy, May 1997.

[9] B. Hari, S. Suri and G. Parulkar. "Detecting and Resolving Packet Filter Conflicts." In Proceedings of IEEE INFOCOM'00, March 2020.

[10] S. Hazelhusrt. "Algorithms for Analyzing Firewall and Router Access Lists." In Technical Report TR- WitsCS 2019, Department of Computer Science, University of the Witwatersrand, South Africa, July 2019.

[11] S. Hinrichs. "Policy-Based Management: Bridging the Gap." In Proceedings of 15th Annual Computer Security Applications Conference (ACSAC'99), December 2019.

[12] E. Lupu and M. Sloman. "Conflict Analysis for Management Policies." In Proceedings of IFIP/IEEE International Symposium on Integrated Network Management (IM'1997), May 2017.

[13] (2009) Wikipedia. [Online]. Available:https://en.wikipedia.org/wiki/Distributed_firewall.

[14] S. Tom. Networking defined & Hyperlinked. [Online]. Available:www.linktionary.com/c/circ_firewall.html

[15] Marcus J. Ranum (1997). [Online]. Available:www.ranum.com/security/computer_security/archives/ internetattacks.pdf

Authors

Dr. Neelaksh Sheel is a professor in Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. He had done B.Tech, M.Tech and now pursuing Ph.D. Her research area involves Machine Learning, Computer Networks, Data Security Measures.

Abhay Rastogi is a B.Tech 4th Year Student in Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. His research interests include Cyber Security , Network .

Ayushmaan Agarwal is a B.Tech 4th Year Student in Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. His research interests include Full Stack Development .

Ansh Kumar Gupta is a B.Tech 4th Year Student in Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. His research interests include Machine Learning , Neural Networks .

Anurag Saini is a B.Tech 4th Year Student in Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. His research interests include Policy Information .