An Analysis of Firewall Configuration Best Practices: Policy Design and Implementation to Improve Network Security

Jaydon Humphries, Latijah James, Ryan Johnson

Abstract—**Firewalls are essential for protecting digital systems from unauthorized access and cyber threats. This project investigates practical firewall configuration strategies using both Uncomplicated Firewall (UFW) and pfSense. A virtual lab environment was created in VirtualBox to simulate real-world network conditions. UFW was first implemented to evaluate rule-based traffic control in a Linux system. PfSense was then introduced as a more advanced solution to explore firewall rules, set up different network connections, review logs and help fix problems and log-based troubleshooting. Testing tools such as ping, curl, and nmap were used to simulate traffic and validate firewall behavior. The objective was to observe how different rule sets and firewall tools influence traffic control and security enforcement. This hands-on approach provides meaningful insight into the process of building secure, policy-driven firewall configurations in virtualized environments.**

I. INTRODUCTION

In today's digital world, cybersecurity has become a critical concern for organizations of all sizes. Among the key defenses in any security system, firewalls are one of the most important. They act as both the first line of protection and the last barrier that controls what comes in and what goes out of a network. When properly configured, they enforce an organization's access policies, filter out malicious or unauthorized traffic, and reduce exposure to data breaches and network-based attacks. However, misconfigurations or outdated rules often turn firewalls from powerful protectors into weak links. Therefore, understanding and applying best practices for firewall configuration is indispensable to safeguarding sensitive resources and keeping systems running smoothly and securely.

When firewalls are configured correctly, they block unauthorized users, help stop attacks, and enforce security rules. But if they're set up the wrong way or if they aren't updated regularly, they can leave the network open to serious risks. Some of the biggest data breaches have happened because firewalls were either misconfigured or not used to their full potential. For example, an organization might accidentally leave a port open that no one uses anymore, which could allow an attacker to slip through unnoticed. Learning and applying firewall configuration best practices is essential for anyone managing a network. It's not just about blocking bad traffic; it's about building a system that supports both safety and day-to-day business.

There are many reasons why firewalls remain a significant part of any security setup. They are often the first tools used to apply access controls to a network and play an important role in enforcing company policies. As attackers become more advanced, the way a firewall is configured can make a major difference in how prepared a system is to handle a threat. Misconfigured firewalls are one of the top issues found during security audits. Whether it is an overly permissive rule, a forgotten service still running, or poorly defined access controls, even a small error can expose entire systems to potential harm [1]. To understand why configuration matters, it helps to look at how networks are structured today. Most organizations do not just have one firewall. Instead, they use different layers of security, placing firewalls between zones such as internal networks, guest access, and public-facing services. Each zone has different needs and risks, and each one should be protected by its own set of rules. This approach makes it harder for attackers to move freely if they gain access to one part of the system. It also gives administrators better control and visibility into where traffic is coming from and going to [2].

A strong firewall setup begins with a clear set of rules that reflect the specific needs of the network. While the technical details may vary depending on the environment, the core goal is the same; allow only what is necessary and block everything else. This mindset is known as the principle of least privilege. If a service or port is not needed, it should be closed. If a user does not need access to a system, access should be restricted. Following this rule helps reduce the number of ways an attacker could try to break into the network [3]. Another important part of firewall management is regular review and maintenance. As systems grow and change, rules that were once useful may become outdated. Keeping unused rules, outdated IP addresses, or unnecessary services in the firewall can create confusion and increase the chances of an error. Regularly reviewing and cleaning up firewall settings helps maintain order and reduces the chances of leaving behind security holes. In some cases, automated tools are used to help identify old or unused rules, but manual review is still necessary to ensure accuracy and context [3].

Good firewall management also involves understanding how to respond to problems. When something unusual happens, whether it's a sudden spike in traffic or a potential breach, it is the firewall logs that provide the first clues. These logs help security teams understand what kind of traffic was involved and whether it should have been allowed. Logging is not only useful during incidents but also during regular audits and compliance checks. Organizations that keep detailed logs are often better equipped to detect issues early and respond effectively [4].

In addition to the tools and rules, it is important to recognize the human side of firewall management. Even the best system can fail if it is not managed properly. Mistakes can happen when settings are changed without proper knowledge or when teams are not trained on how to handle configurations. That is why ongoing training, good documentation, and clear policies are important. Firewalls are not just about technology. They are also about how people use and manage that technology every day [4].

As networks continue to grow more complex and threats become harder to detect, firewalls will continue to be a key part of defense strategies. Their role may evolve, and new features may be added, but the foundation will always come back to how well they are configured. This paper will explore the most important best practices for setting up and managing firewalls, highlighting how proper configuration can help reduce risks and strengthen network security.

II. Literature Review

A. Importance of Firewall Configuration

Firewalls serve as the foundation for moder day network security; they are considered the first line of defense against unauthorized access or malicious attacks. They function similarly to a water filter, filtering out incoming and outgoing network traffic based on custom security rules. However, the actual power of a firewall is defined not by the technology used to design it, but rather by the complexity and attentiveness of its configuration rules. The successful configuration of a firewall ensures that only legitimate packets are permitted through the network while blocking and logging malicious traffic, thereby significantly reducing potential attack vectors.

Extensive research has been conducted on the critical role firewall configuration plays in reducing attack surfaces and the potential damage it can cause if not configured properly. According to the International Journal of Computer Science and Network Security [5], firewall security is regarded as one of the most essential and non-optional practices for defending networks against unauthorized access. A misconfigured firewall can pose an extreme threat to an organization's security and introduce a handful of vulnerabilities.

Voronkov et al. [6] further supports this by providing extensive data that explains the most common causes of security breaches being misconfigured firewalls. These findings reinforce the argument that proper firewall configuration is essential to safeguarding an organizations resources.

B. Common Configuration Mistakes

Each firewall is responsible for keeping the information of users and businesses safe. Mistakes in firewall configuration, or misconfigurations focus more so on the mistakes of how the firewall is set up. Misconfigurations can happen for a number of reasons, mainly due to bypassing authentication. These misconfigurations occur mainly because multiple rules overlap in scope. Generally, though, the rules will tend to overlap deliberately; by doing this it informs the user of a potential threat.

There are several different mistakes that can occur such as shadowing, this is a serious misconfiguration. Shadowing can lead to non-threatening traffic within the network getting denied, which can seriously impact the workflow of an organization. Redundancy can also severely impact network traffic. This clutters the firewall policy while also making it difficult to troubleshoot audit traffic.

Altogether, misconfigurations cause the network to lose efficiency and weaken the security structure. Internal errors and external threats are much more likely to occur within the network, but, with proper rules, frequent auditing, are essential steps for setting up a strong firewall and network security.

C. Firewall Types and Best Practices

1) Packet Filtering

The very first firewall to exist was the simple Packet Filtering firewall, which operates at the Network layer of the OSI model. A packet filtering firewall is considered a stateless firewall. A stateless firewall interprets each network frame or packet individually. The mentioned firewall above functions by continuously monitoring incoming and outgoing traffic, filtering out packets based on a set of predetermined rules. An example of this would be if a packet enters through the network and matches one of the rules set, if the rule is set to allow, the packet will pass through, however if set to deny, the packet will drop. Types of rules can vary, but the majority of them are constructed based on criteria like the type of protocol being used, source and/or destination IP, source and/or destination port, and if the traffic is inbound or outbound [7].

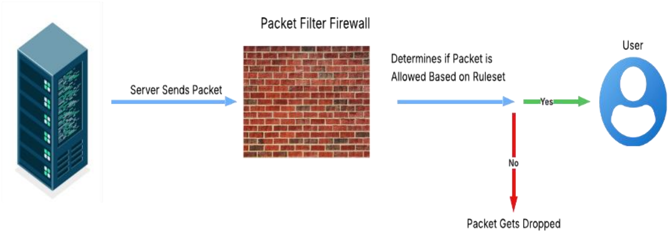

Fig. 1. Packet Filtering

Packet Filtering Firewalls can be easy to implement due to their simplified nature. However, it is not considered secure due to its many vulnerabilities such as:

- DOS Attacks
- IP Spoofing
- Tiny Fragment attack

2) Stateful Firewall

Stateful firewalls are the next step up from packet filtering. They build upon the basic concepts from packet filtering by tracking the state of active connections like the TCP three-way handshake. The key feature of the stateful firewall is that it can comprehend if a packet is a part of an existing connection, the start of a new connection, or if it is an invalid packet, thus tracking established flows and all packets that are inside the traffic flow in both directions. What makes this possible is the integration of a cache inside the firewall, called a state table, that allows it to store and keep track of each network connection. A good example is whenever a SYN-packet passes through a network, a record is stored containing header contents like port number and IP address. After doing so, any connections made next are then checked for in the state table. If the connection already exists, the packet is allowed through, subsequently, if it does not exist, the firewall evaluates the packet based on basic packet filtering rules to determine if it should be allowed to pass through. In order to reduce latency and resource consumption, inactive connections are eventually removed from the state table, this is also true for connections torn down using a FIN packet. Unfortunately, stateful firewalls remain vulnerable to Denial of Service (DoS) attacks when their state tables are overwhelmed by excessive connection attempts. This can occur through SYN flood attacks or by generating an unusually high number of traffic flows, which exhaust available resources and prevent legitimate connections from being established [4].

3) Application-Gateway Firewall

Unlike the traditional firewalls that were just covered, application gateway firewalls differ due to the fact that packets are inspected at the application layer of the OSI model. It provides controlled use of certain applications across a network. Let's use FTP as an example, when a user tries to access a remote service like FTP, the gateway prompts the user to provide the address of the intended remote host. After the user provides the required credentials, like username and password, the gateway initiates the connection to the application on the remote host on behalf of the user. After the connection is established, the gateway manages the flow of data between the two parties. However, if the data being transmitted falls out of bounds of the gateway's ruleset, the firewall will block the session and will not forward the data to the destination server [8].

D. Gaps and Weaknesses in Current Research

With extensive knowledge in firewall configurations, gaps and weaknesses still persist in hypothetical and applied research. Automatic translation of organizational security policies into concrete firewall rule sets remains under-explored. Kovacevic et al. (2022) surveyed over 20 approaches aimed at formal or graphical policy languages compiled into enforceable firewall rules. While these methods improve readability and correctness, most still require significant manual intervention, and none are widely adopted in practice. The authors note that "[automatic translation] has seen much less success … there are still many drawbacks" limiting practical deployment [9].

Voronkov et al. [10] conducted semi-structured interviews and a systematic literature review on firewall configuration usability. They found most proposed tools lacked empirical evaluation, and that usability itself was often not clearly defined. Only a minority of models employed human-computer interaction or usability design guidelines, leaving administrators without validated aids for complex rule creation [10]. This denotes a persistent disconnect between published techniques and their validation in real-world administrative workflows.

Misconfiguration issues—such as rule shadowing, correlation, and redundancy—have been characterized, but tools typically target single firewalls. MDPI's review highlights that inter-firewall rule conflicts (e.g., upstream/downstream inconsistencies) are under-diagnosed. Human operators often remain unaware until anomalous behavior occurs, and existing visualization tools do not scale to enterprise networks with heterogeneous devices [11].

Current firewall research is strong in most conceptual models and classification of configuration errors, but it under delivers in operational applicability.

III. Firewall Configuration and Testing

A. Initializing Environment

To evaluate firewall configuration best practices, we are building a virtual lab environment using Oracle VirtualBox. This setup allows us to simulate a secure network environment where firewall rules can be configured and tested in isolation. The goal is to assess the behavior and effectiveness of both UFW and pfSense in real-world scenarios.

The lab will include two Ubuntu Desktop virtual machines. One acts as the firewall (Firewall VM), and the other functions as a client (Internal VM). The Firewall VM will be configured with two network adapters. The first is set NAT to simulate external traffic, and the second uses the Internal Network setting to simulate private internal traffic. The Internal VM will be connected to the same internal network to represent an internal user.

We will manually assign static IP addresses to both VMs for consistent communication. The Firewall VM will use 192.168.10.1, and the Internal VM will use 192.168.10.2. Tools such as ufw, nmap, curl, and net-tools will be installed to support testing. These tools will allow us to analyze connectivity, scan ports, observe firewall behavior, and validate traffic control.

The virtual environment will allow us to safely apply firewall configurations, test for misconfigurations, and validate rules without affecting any real systems or networks. It also will provide a controlled setting to explore different firewall behaviors using real commands and tools.

B. Firewall Configuration

The first tool we plan to test is UFW, also known as Uncomplicated Firewall, on the Ubuntu Firewall VM. UFW will be reset to remove any prior configurations. We will then apply a default policy that blocks all incoming traffic and allows all outgoing traffic. This default deny policy is a widely recommended best practice that reduces the system's exposure to unknown threats.

Logging will be enabled so we can track how traffic is handled. Specific allow rules will be added for services such as SSH and HTTP. This will let us observe how UFW manages exceptions and ensures only approved traffic is permitted.

To verify the rules, we will use the ufw status verbose command. This will display the active rules and confirm that our default policies and exceptions are in effect. Screenshots of the command output will be taken to document the configuration and serve as evidence of the correct setup.

This setup provides a foundation to test whether the firewall behaves as expected. It also allows us to evaluate how UFW supports best practices in a practical environment.

1) Uncomplicated Firewall (UFW)

To examine UFW's behavior in a controlled lab environment, we will perform a series of tests using the Internal VM to simulate typical network interactions. These tests are intended to help us understand how UFW handles both allowed and unauthorized traffic based on its default and custom rules.

The first test will involve using the ping command from the Internal VM to the Firewall VM. This will help verify that both machines are properly connected through the internal network and that basic communication between them is possible.

Next, we will use the curl command to attempt an HTTP request to the firewall's IP address on port 80. Since we do not plan to run a web service or allow HTTP traffic by default, this request is expected to fail. We will also run a nmap scan from the Internal VM to identify which, if any, ports are open on the Firewall VM. This will allow us to confirm whether UFW is effectively blocking unknown or unapproved traffic.

These tests will be used to confirm how UFW enforce a secure configuration by default. With a deny-all policy in place for incoming traffic, we expect the firewall to prevent access to any services unless explicitly allowed. This process will also demonstrate how tools like ping, curl, and nmap can be used to validate firewall behavior and verify whether best practices are being followed.

Following these tests, we will compare UFW's performance and configuration flexibility with pfSense by replicating similar scenarios. This will help us determine which firewall solution is more effective or better suited to different network needs.

2) PfSense

Following the implementation of UFW, pfSense will be used as a more advanced platform to extend our firewall testing. PfSense is an open-source firewall and router solution based on FreeBSD, and it is widely adopted in both business and home networking environments. The plan is to install pfSense as a virtual machine in VirtualBox, where it will serve as the firewall system within our lab network. This setup will allow us to explore how pfSense handles firewall rule configuration, and traffic logging, in comparison to UFW.

The pfSense virtual machine will be configured with two network interfaces. One will be connected to a NAT adapter to simulate internet access, and the other will be assigned to an internal network to communicate with the Ubuntu test machine. Once the VM is installed and booted, we will access the pfSense web interface through a browser and complete the setup wizard. The goal is to establish a secure baseline by applying a default deny configuration and then manually adding rules to allow specific traffic, such as SSH, ICMP, and HTTP.

After the initial rule set is in place, we plan to test traffic flow using tools like ping, curl, and nmap from the Ubuntu machine. These tests will help us observe how pfSense manages traffic across different ports and protocols, and whether the firewall behaves as expected when enforcing a least privilege policy. Screenshots of the dashboard and rule table will be captured to document the interface and the logic applied during configuration. In the end, pfSense will give us a clearer view of how traffic is handled. how logs are recorded and how firewall rules are processed. These are key aspects we intend to compare directly against our earlier UFW results.

C. Traffic Testing and Validation

To better understand the impact of our firewall configurations on network behavior, we performed a series of traffic-based tests using both UFW and pfSense. These tests were designed to mimic common activities such as checking connectivity, scanning for open ports, and attempting to access web services. Each action allowed us to observe how the firewall responded to typical types of traffic and whether it applied the intended rules.

In the UFW setup, we used two Ubuntu virtual machines, assigning one as the internal user system and the other as the firewall. Both were configured with static IP addresses on the same internal network. Using basic tools like ping, curl, and nmap, we tested traffic flow from the internal machine to the firewall. The first test involved sending a ping to the firewall to confirm communication across the internal network. The firewall responded without any issues, which showed that the connection between the two machines was working as expected.

Next, we attempted to connect to port 80 on the firewall using the curl command. This request failed, which aligned with our configuration since there was no web server running, and the firewall was set to deny unauthorized incoming traffic. A port scan using nmap also showed that all scanned ports were closed. This result confirmed that the firewall was not exposing unnecessary services.

Once pfSense setup is complete, we plan to apply the same traffic testing process. After adding rules through its graphical interface, we will use the internal Ubuntu machine to run connectivity checks and scan for open ports. This will allow us to compare the performance and response of pfSense against what we observed with UFW.

D. Results and Observations

The outcomes of our UFW testing supported the security goals outlined in our configuration plan. Ping responses confirmed that the firewall and internal system were able to communicate as intended. This was important because it showed that our network setup was correct and ready for further testing.

The attempt to reach port 80 using curl failed, which matched our expectations. Since we did not install a web server and our firewall was configured to block incoming traffic by default, there was no reason for this request to succeed. When we followed up with a scan using nmap, the tool reported that all ports were closed. This supported the conclusion that the firewall was actively blocking unauthorized traffic and not exposing any unused or vulnerable services.

These results showed that the UFW rules were applied correctly and that the firewall followed a default deny configuration. It only allowed the types of traffic we specified and blocked the rest. This behavior is consistent with best practices in network security and helped reduce unnecessary exposure to potential threats.

E. Comparative Analysis

PfSense and UFW are both open-source tools used to manage network traffic, but they differ quite a bit in how they're designed and what they offer. Based on our testing, pfSense provided a more complete environment for working with firewall rules, tracking real-time connections, and reviewing logs. The web-based interface made it easier to see how traffic moved through the network and to fine-tune rule sets when needed. This kind of control is especially helpful in setups where more advanced security is required.

UFW, on the other hand, is a simpler tool that's often used by default on Ubuntu systems. It's built to be easy to use and works well for basic firewall configurations like opening or closing specific ports. Since it's managed through the command line, UFW is great for quick changes but doesn't offer the same kind of visibility or features as pfSense. There are no built-in tools for analyzing traffic or a graphical dashboard to help with rule management.

What we observed during our testing lines up with what others have reported in recent reviews. PfSense is known for supporting more advanced features like VPN setup, IPv6, and traffic shaping, all of which are not available in UFW

In summary, pfSense is a better fit for people or organizations that need more control and visibility into their firewall. UFW is a good option for basic protection and smaller environments where simplicity matters more than advanced features. Choosing between them really depends on how much security and customization your network needs.

IV. Project Limitations and improvements

Although the project gave us a great insight into how firewall tools like UFW and pfSense work, there were some limits to what we could do. Everything was set up in a virtual lab using VirtualBox, which made it easier to test in a controlled space. Due to the fact that it wasn't a real-world network with multiple users and constant traffic, we didn't get to see how the firewalls would perform under heavier or more unpredictable conditions.

Another limitation was the tools we used for testing. We mainly relied on simple commands like ping, curl, and nmap to check if the firewall rules were working. These tools helped us understand basic traffic behavior, but they didn't give us a full picture of how the firewalls would react to more complex or malicious attacks. Using more advanced tools like Metasploit or a traffic generator could have made the testing more realistic.

If we had more time and resources, we would expand the setup to include more machines, simulate outside attacks, and push more traffic through the network. We'd also look at other firewall systems, like Cisco or IPFire, to compare even more options. In the future, it would be helpful to include automated testing and tools that give a clearer view of what's happening behind the scenes, especially for larger or more complex environments.

V. Conclusion

In conclusion, this project highlighted the significance of applying structured and secure firewall configurations in any networked environment. By testing both UFW and pfSense, we observed that while UFW is lightweight and easy to configure for basic security needs, pfSense provides more advanced control through its graphical interface and detailed logging features. The traffic testing phase confirmed that setting up a default deny configuration, followed by explicitly defined allow rules, is effective in reducing exposure to unwanted access. We also recognized that misconfigured or overly permissive rules can weaken firewall performance, regardless of the platform used. Overall, the experience emphasized that proper firewall management is not just about using the right tool, but also about understanding the environment, setting clear policies, and continuously reviewing and adjusting configurations to adapt to evolving network demands.

References

[1] Ranathunga, D., Roughan, M., Nguyen, H., Kernick, P., & Falkner, N. (2016, December). Case studies of SCADA firewall configurations and the ... IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. https://roughan.info/papers/ieee_trans_man_2016.pdf

[2] Aledhari, M., Mandal, S., Aneja, N., Dautrey, M., Nighot, R., Mantri, P., & Bielby, J. B. (2017, May). Protecting internet traffic: Security challenges and solutions. https://standards.ieee.org/wp-content/uploads/import/documents/other/whitepaper-protecting-internet-traffic-dh-v1.pdf

[3] Anwar, R. W., Abdullah, T., & Pastore, F. (2021, October 2). *Firewall best practices for Securing Smart Healthcare Environment: A Review*. MDPI. https://www.mdpi.com/2076-3417/11/19/9183

[4] Liang, J., & Kim, Y. (2022). *Evolution of firewalls: Toward securer network using Next Generation Firewall | IEEE Conference publication | IEEE Xplore*. 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC). https://ieeexplore.ieee.org/document/9720435

[5] Alsaqour, R., Motmi, A., & Abdelhaq, M. (2021, April). *A systematic study of network firewall and its implementation*. IJCSNS International Journal of Computer Science and Network Security, VOL.21 No.4, 199–206. http://paper.ijcsns.org/07_book/202104/20210424.pdf

[6] Voronkov, A., Iwaya, L. H., Martucci, L. A., & Lindskog, S. (2017). Systematic literature review on usability of firewall

configuration. *ACM Computing Surveys, 50*(6), Article 87, 1–35. https://doi.org/10.1145/3130876

[7] S. Krit and E. Haimoud, "Overview of firewalls: Types and policies: Managing windows embedded firewall programmatically," *IEEE Xplore*, May 01, 2017. https://ieeexplore.ieee.org/document/8273003

[8] N. Susanto, April Firman Daru, and Febrian Wahyu Christanto, "Packet Filtering Gateway and Application Layer Gateway on Mikrotik Router Based Firewalls for Server and Internet Access Restrictions," Dec. 2023, doi: https://doi.org/10.1109/icteca60133.2023.10490754.

[9] I. Kovačević, B. Štengl, and S. Groš, "Systematic review of automatic translation of high-level security policy into firewall rules," *arXiv*, Dec. 2022. doi: https://doi.org/10.23919/MIPRO55190.2022.9803570

[10] A. Voronkov, S. Lindskog, and L. A. Martucci, "Challenges in managing firewalls," in *Secure IT Systems*, Lecture Notes in Computer Science, vol. 191–196, Jan. 2015. doi: http://dx.doi.org/10.1007/978-3-319-26502-5_13

[11] "Misconfiguration in Firewalls and Network Access Controls: Literature Review," *MDPI* (2021). https://doi.org/10.3390/fi13110283