pfSense Firewall Lab Simulation: DoS Detection & Mitigation in a Virtual Environment

Author: Rishav Kumar Thapa

Platform: Oracle VirtualBox

Victim OS: Ubuntu

Server OS: Pfsense Firwall

Attacker Os : kali

Date : 6/17/2025

1. Introduction

This lab guide provides a comprehensive walkthrough for setting up a pfSense-based firewall environment using Oracle VirtualBox. It simulates a basic two-zone security setup with an internal LAN and an external WAN to demonstrate firewall configuration and DoS mitigation techniques. The lab is particularly beneficial for cybersecurity students and professionals aiming to understand perimeter defense using pfSense.

Through this setup, users will deploy pfSense as the firewall, Kali Linux as the attacker machine on the WAN side, and Ubuntu Desktop as the victim on the LAN. Using tools like hping3 and Wireshark, the guide walks through launching and detecting a simulated DoS attack and mitigating it using pfSense firewall rules**.**

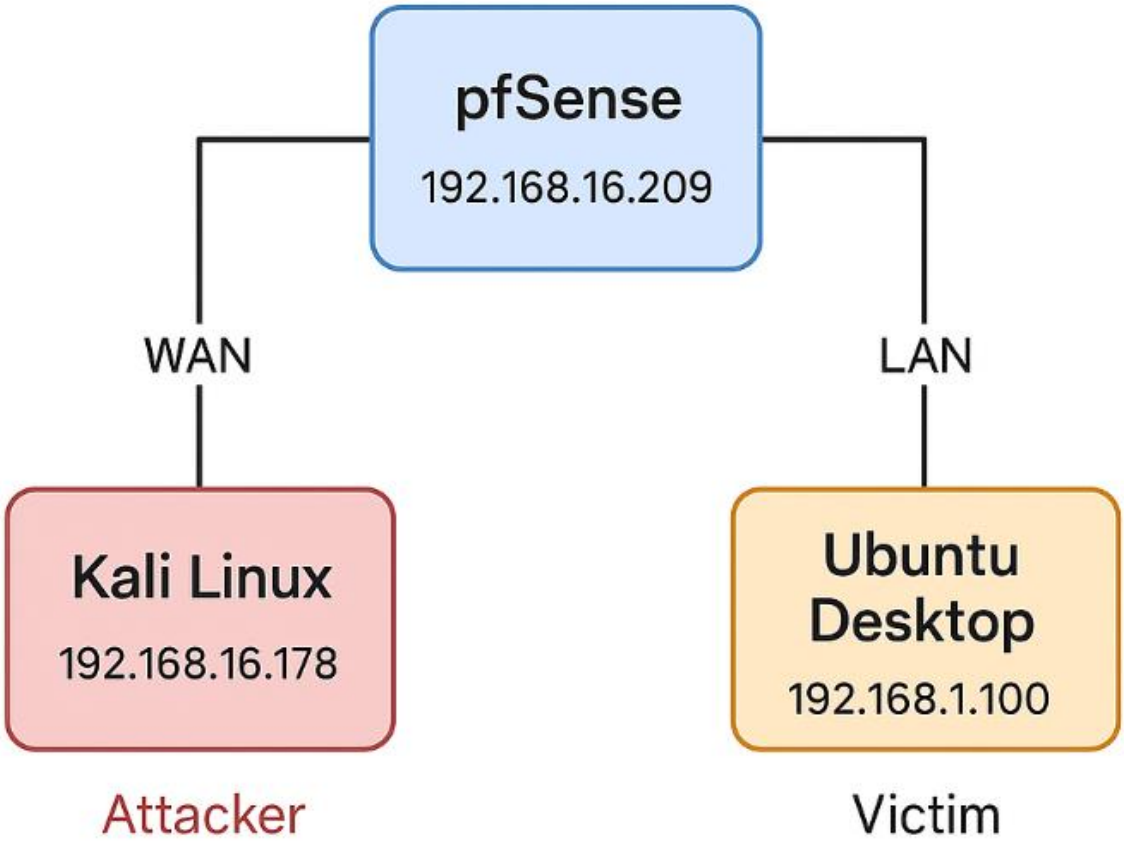
2. Objectives

- Deploy a pfSense firewall in VirtualBox with separate WAN (bridged) and LAN (internal) segments.
- Simulate an external attack from Kali Linux on the WAN interface.
- Configure Ubuntu Desktop as the internal victim machine.
- Demonstrate a DoS attack using hping3.
- Use pfSense firewall rules to mitigate and block the attack.

3. Network Topology & IP Plan

Device	Network Adapter	IP Address	Role
pfSense (WAN	Bridged	192.168.16.209/24	Edge Firewall
Kali Linux (Attacker)	Bridged	192.168.16.178/24	External Threat Source
pfSense (LAN)	Internal (intnet)	x.x.x.x/24	Default GW + DHCP
Ubuntu (Victim)	Internal (intnet)	192.168.1.100/24	Internal Host

4. Network Architecture Overview



5. Prerequisites

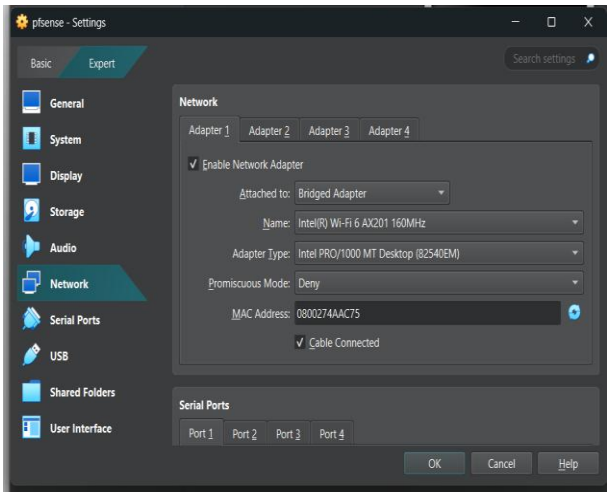
- Oracle VirtualBox (v7 or above)
- ISO files:
    - pfSense-CE-2.7.x-amd64.iso
    - kali-linux-current-amd64.iso
    - ubuntu-22.04-desktop-amd64.iso
- Minimum System Requirements:
    - 8 GB RAM
    - 40 GB disk space
    - Admin rights on host machine

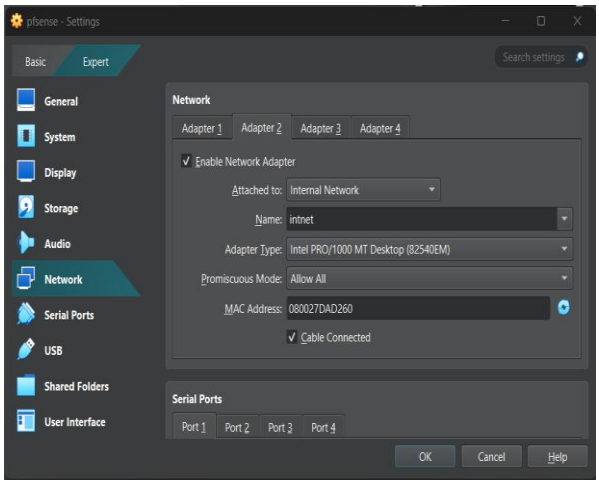
6. Lab Setup Overview

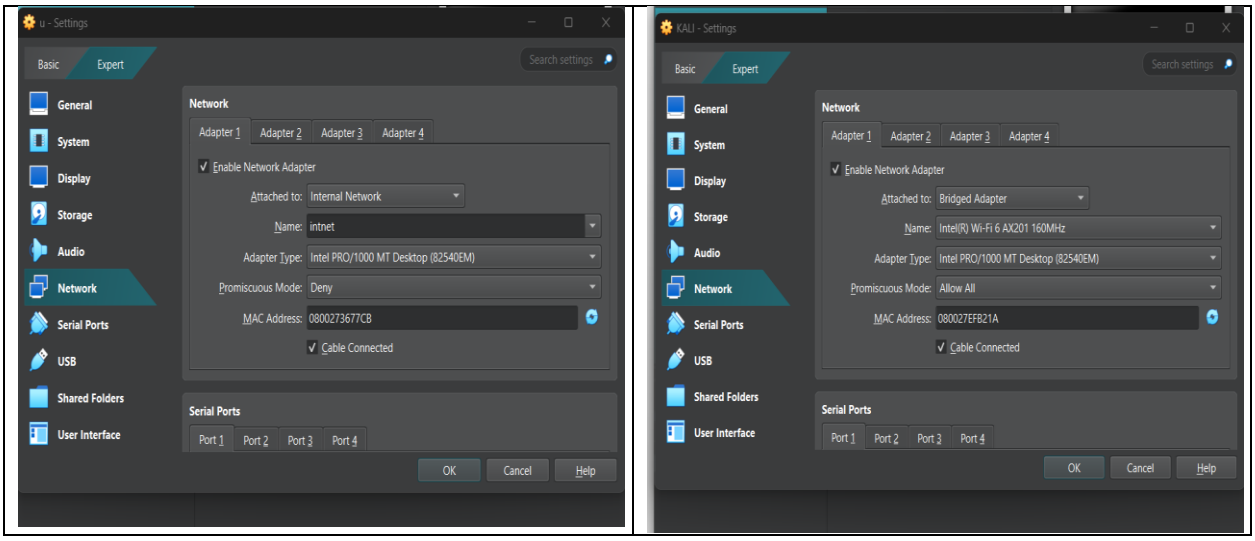
1. Create pfSense VM
    - OS Type: FreeBSD 64-bit
    - Network:
        - Adapter 1: Bridged (WAN)
        - Adapter 2: Internal (LAN / LabNet)
2. Create Kali VM (Attacker)
    - OS Type: Debian 64-bit
    - Network:
        - Adapter 1: Bridged
3. Create Ubuntu VM (Victim)
    - OS Type: Ubuntu 64-bit
    - Network:
        - Adapter 1: Internal (LabNet)

7. Firewall Lab Setup

This section outlines the step-by-step process of building a virtualized firewall environment using pfSense within Oracle VirtualBox. The lab setup is designed to replicate a real-world network infrastructure featuring an external (WAN) attacker and an internal (LAN) victim, protected by a pfSense firewall. Each virtual machine is carefully configured to simulate network segmentation, allowing for practical experimentation with firewall rules, attack simulations, and defense strategies. This foundational setup ensures that all components are correctly positioned for demonstrating and mitigating a Denial-of-Service (DoS) attack in a controlled and observable environment.





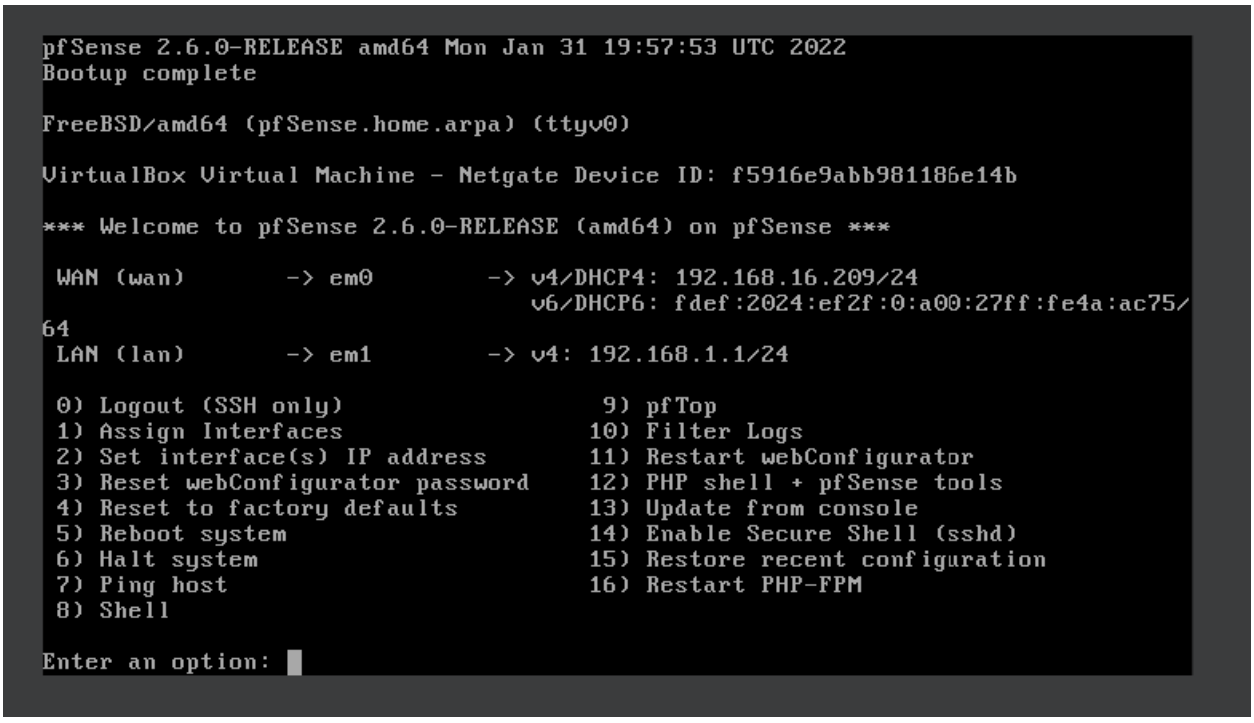


Step 1: Virtual Machine Network Configuration in VirtualBox

To simulate a real-world network with segmented zones, I configured three virtual machines within Oracle VirtualBox as follows:

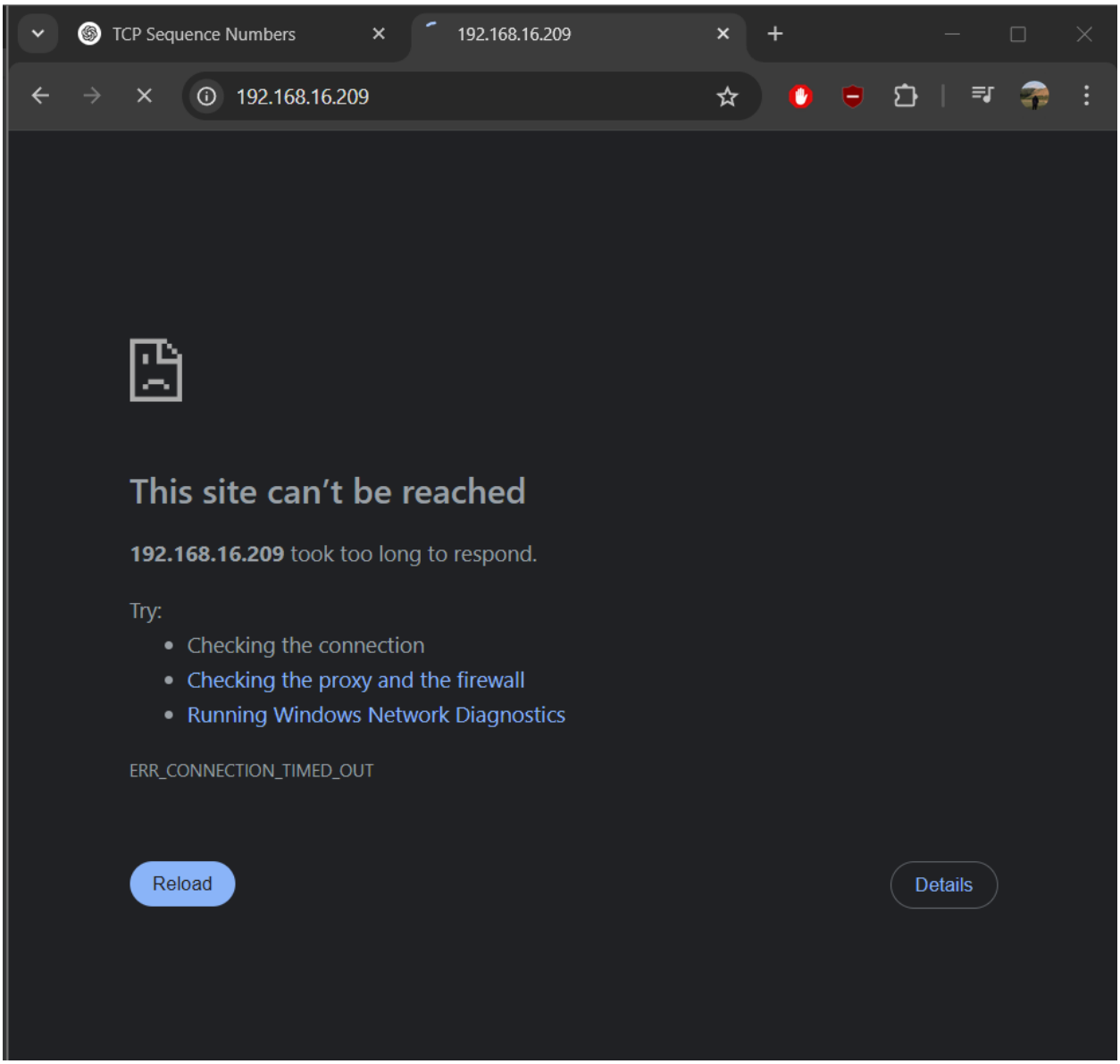
- **Kali Linux** was assigned to the **Bridged Adapter** mode to simulate an external attacker connected to the public/WAN side of the network.
- **Ubuntu Desktop** was connected to an **Internal Network** named intnet, representing a secure internal LAN environment where the victim machine resides.
- **pfSense Firewall** was configured with two network adapters:
    - **Adapter 1 (WAN):** Set to **Bridged Adapter**, enabling pfSense to receive an IP address from the host's physical network and act as the network edge.
    - **Adapter 2 (LAN):** Set to **Internal Network (intnet)**, connecting it to the same LAN segment as the Ubuntu machine, allowing it to serve as the internal gateway and firewall.

This network segmentation allows controlled communication between the external attacker (Kali), the internal victim (Ubuntu), and the security appliance (pfSense), forming the foundation for simulating and analyzing a Denial-of-Service (DoS) attack scenario.



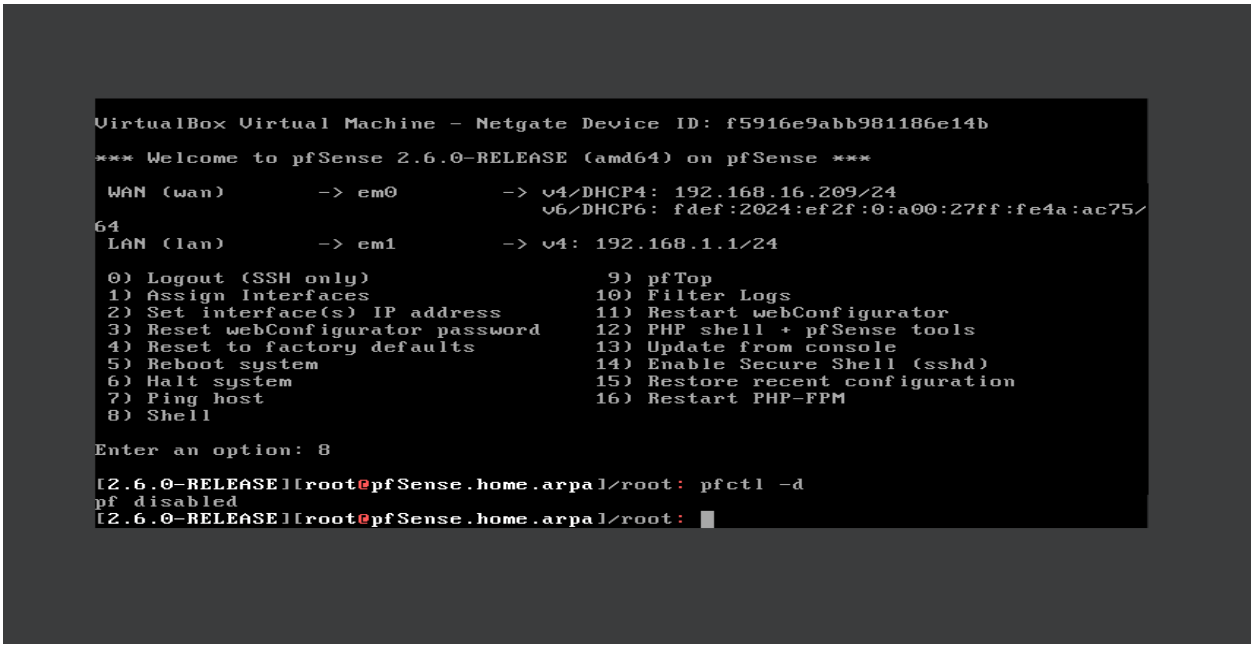
Step 2: Accessing the pfSense Terminal and Initial Setup

After booting the pfSense virtual machine, I accessed the built-in terminal interface to perform the initial configuration.



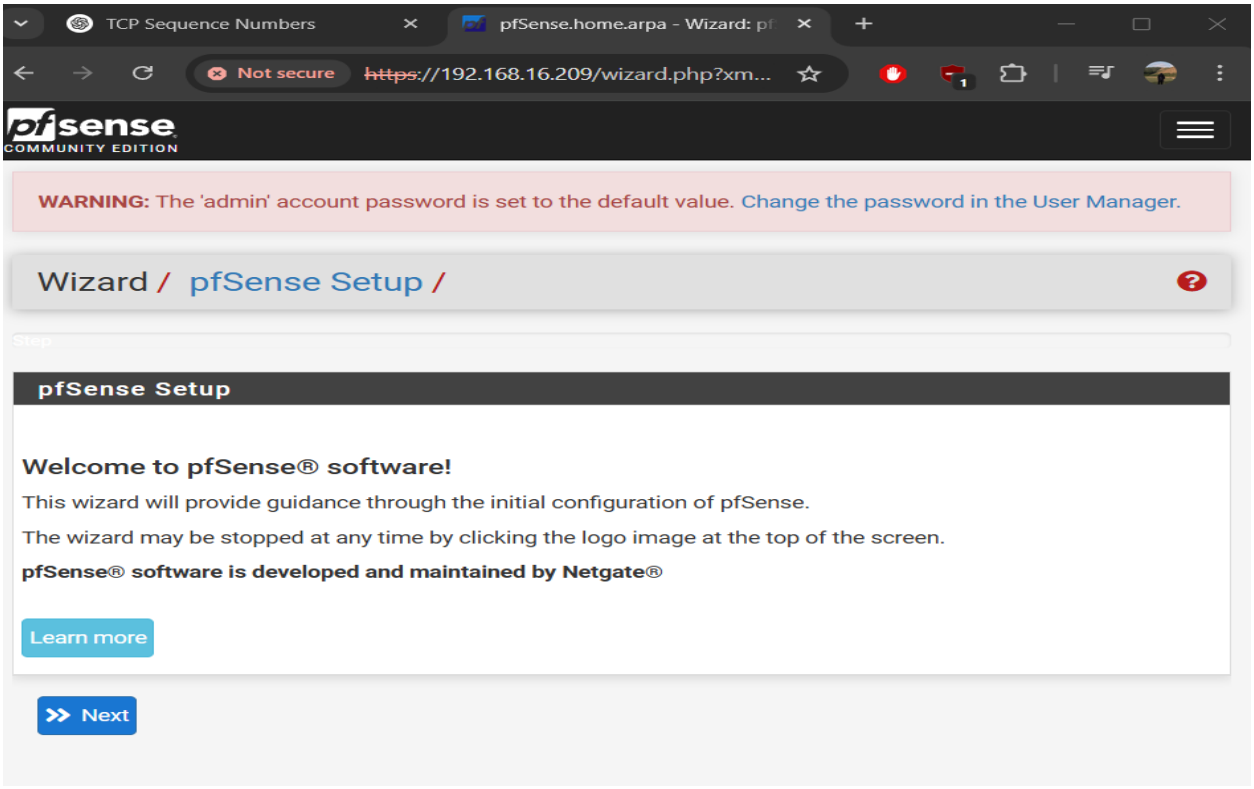
Step 3: Troubleshooting Connection to IP 192.168.16.209

While attempting to access the device at IP address **192.168.16.209** via a web browser,



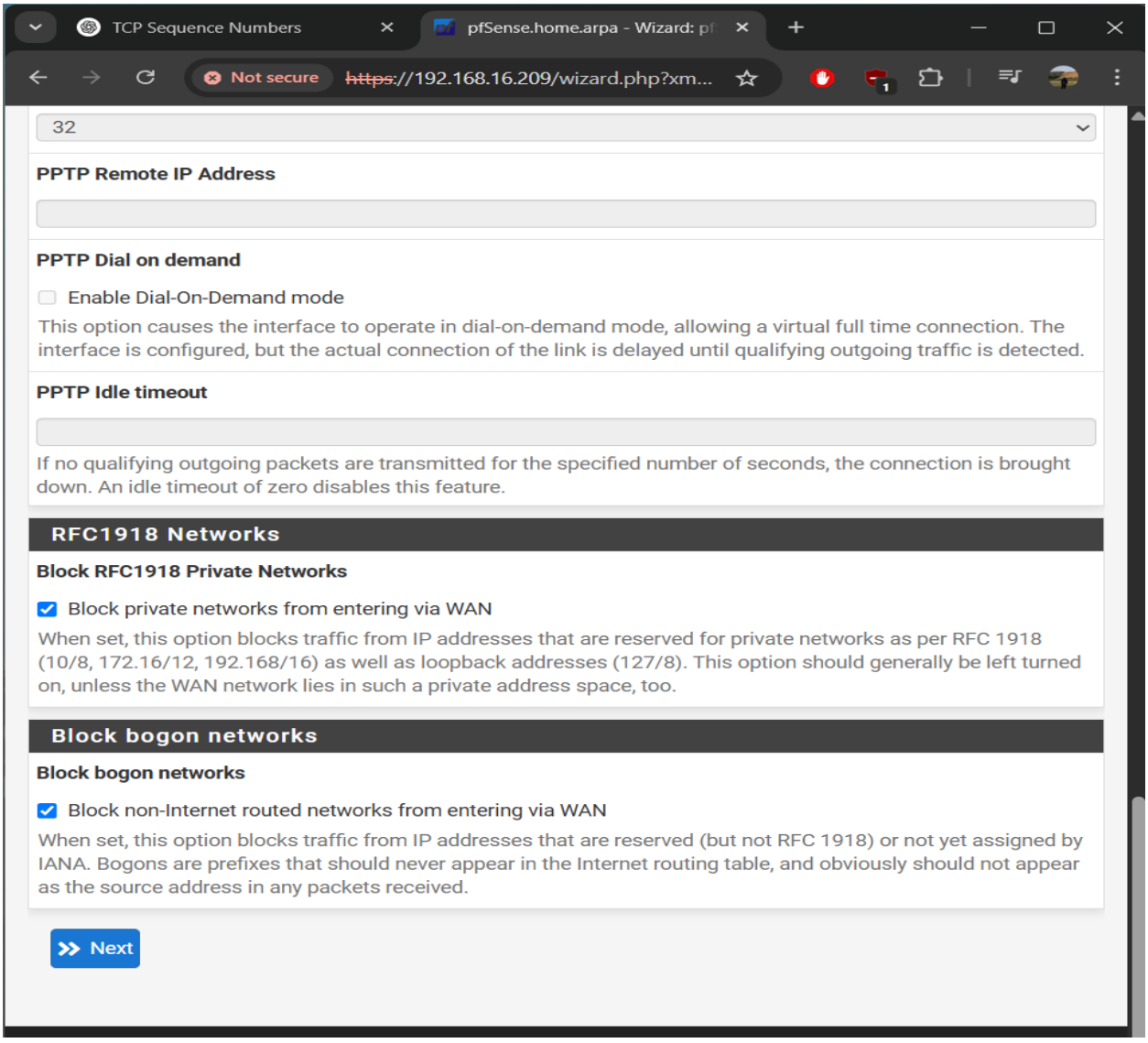
Step 4: Disable pfSense Firewall Temporarily

From the pfSense console shell, I ran the command pfctl -d to temporarily disable the firewall. This allowed me unrestricted access to the pfSense WebGUI for initial configuration without being blocked by firewall rules. This step is only temporary to avoid lockout during setup.



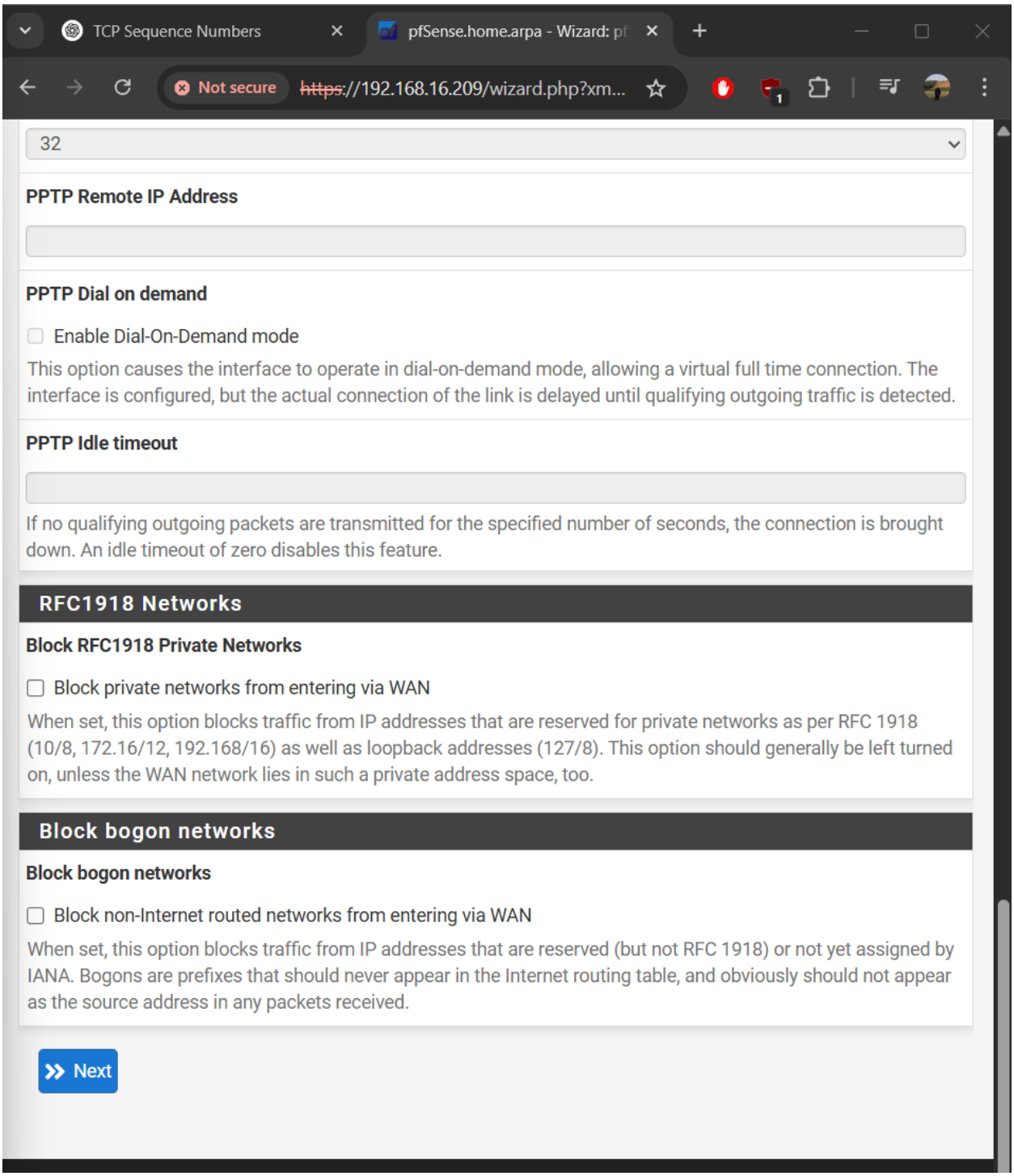
Step 5: Initial pfSense Setup Wizard

Upon accessing the pfSense WebGUI, the Setup Wizard welcomed me to the system and prompted for initial configuration. The wizard provides step-by-step guidance to configure essential settings such as hostname, domain, DNS servers, and admin password. It also issues a warning that the default admin password should be changed immediately for security purposes.



Step 6: Observing Auto-Enabled WAN Firewall Settings

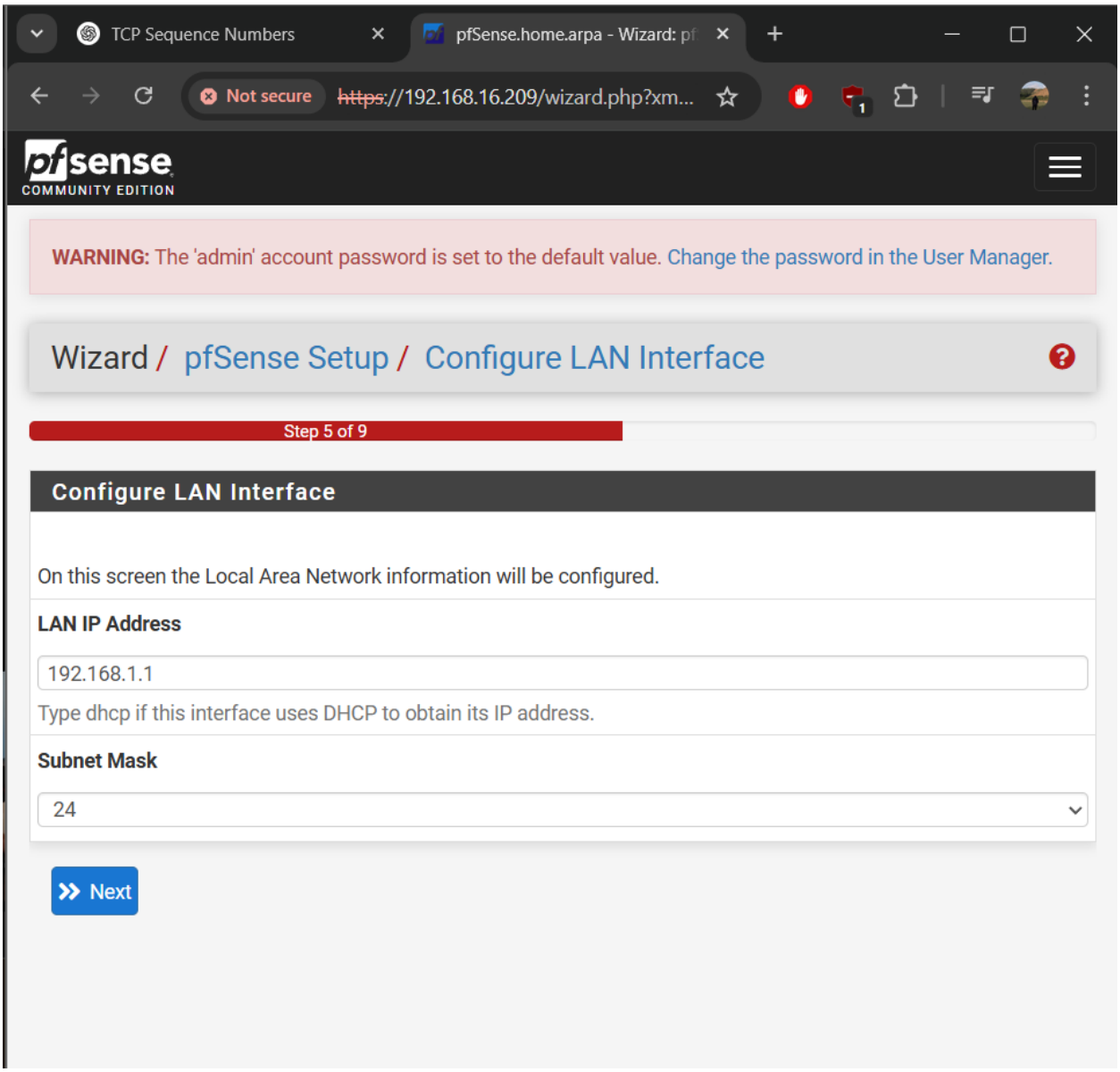
The WAN interface had the options to block RFC 1918 private networks and bogon networks enabled by default. Despite these settings being active, legitimate web pages loaded successfully, indicating that normal internet traffic was allowed while unwanted or suspicious packets were filtered automatically.



Step 7: Disabling WAN Firewall Restrictions to Enable Web Access

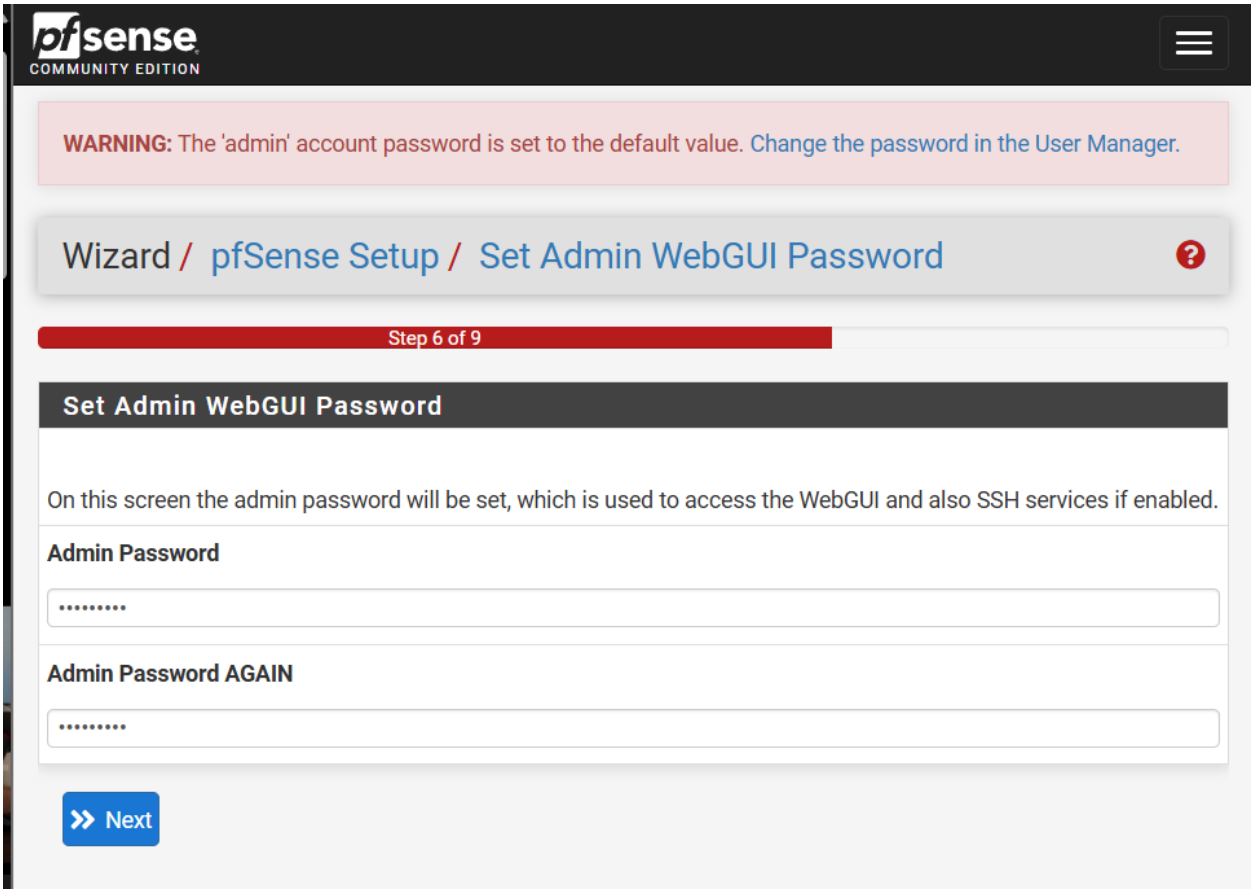
I unticked the options to block RFC 1918 private networks and bogon networks on the WAN interface. This change allows traffic from private IP ranges and reserved IP blocks, which is necessary when the WAN itself uses such private addressing. Disabling these blocks enabled successful web page access through the WAN interface.

Additionally, the Dial-OnDemand mode remained disabled, ensuring the WAN connection stayed active continuously.



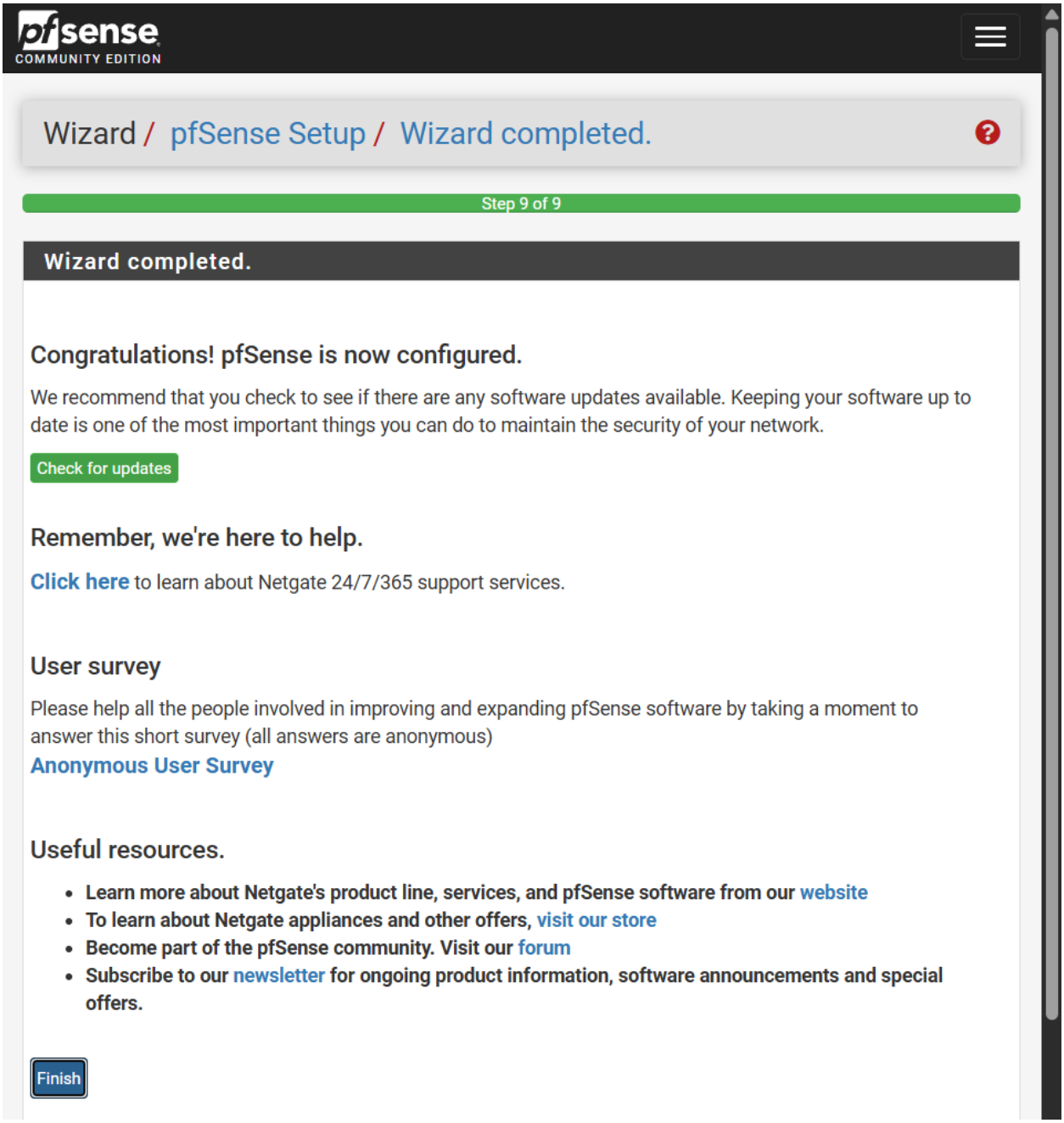
Step 8: Configuring the LAN Interface

In the pfSense setup wizard, I configured the LAN interface with the default IP address **192.168.1.1** and subnet mask **/24**. Since this interface uses a static IP, I left the DHCP option unchecked. I proceeded with these default settings without changing to ensure proper internal network configuration.



Step 9: Setting the Admin WebGUI Password

I set a strong admin password to secure access to the pfSense WebGUI and SSH services (if enabled). This password protects the firewall's management interface from unauthorized access.After entering and confirming the new password, I proceeded to the next configuration step.



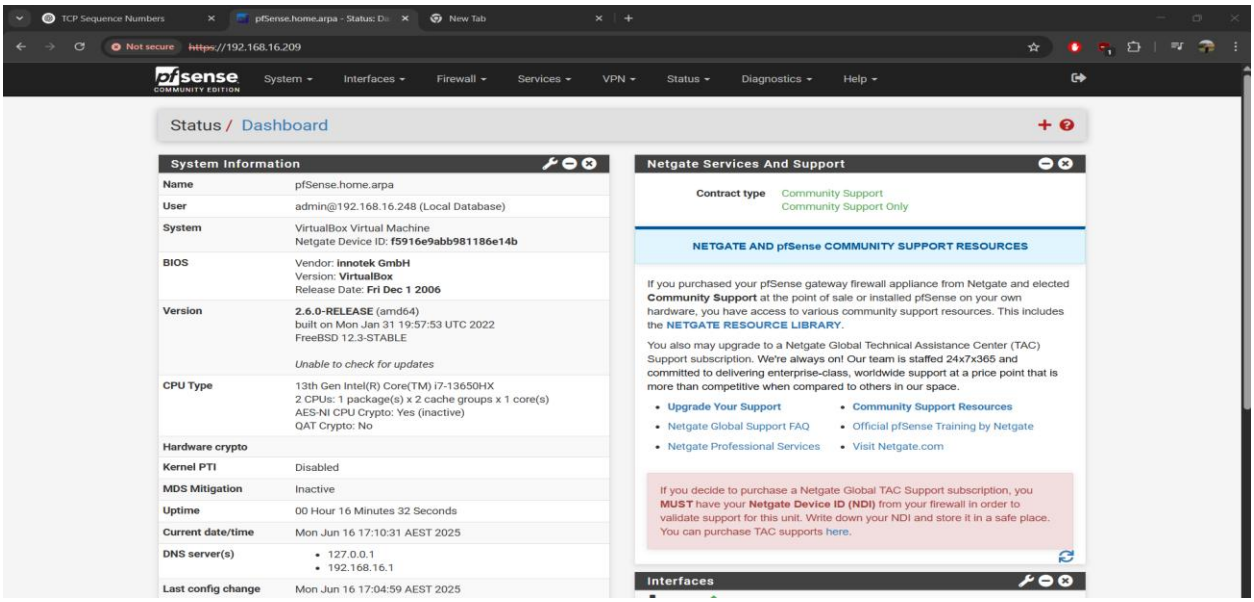
Step 10: Completing the pfSense Setup Wizard

After completing all configuration steps, the pfSense Setup Wizard confirms that the firewall is now configured. It recommends checking for software updates to ensure the latest security and feature improvements. The wizard also offers links to support resources, community forums, and product information. Finally, I clicked **Finish** to exit the wizard and start using the pfSense firewall.

```
 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option: 8

[2.6.0-RELEASE][root@pfSense.home.arpa]/root: pfctl -d
pf disabled
[2.6.0-RELEASE][root@pfSense.home.arpa]/root:
Message from syslogd@pfSense at Jun 16 07:00:24 ...
php-fpm[359]: /index.php: Successful login for user 'admin' from: 192.168.16.248
 (Local Database)

[2.6.0-RELEASE][root@pfSense.home.arpa]/root: pfctl -d
pf disabled
[2.6.0-RELEASE][root@pfSense.home.arpa]/root: pfctl -e
pf enabled
[2.6.0-RELEASE][root@pfSense.home.arpa]/root:
```

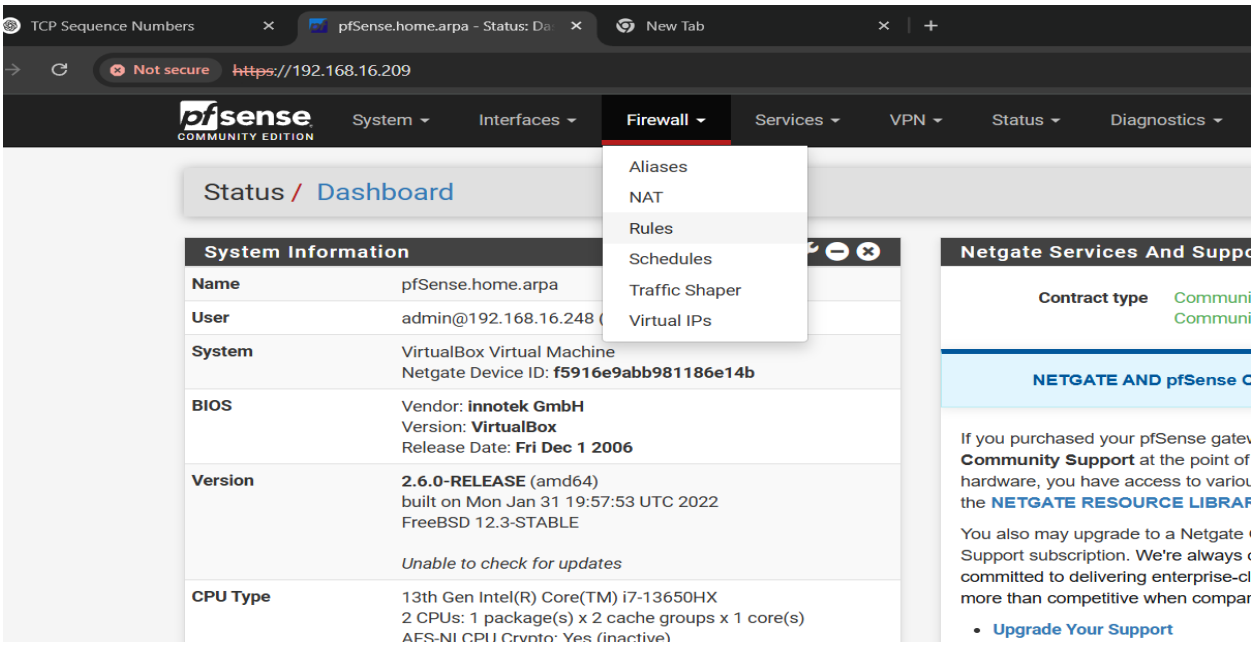
Step 11: Temporarily Disable the pfSense Firewall via Console

From the pfSense console menu, I chose option 8 to access the shell. Then, I ran the command pfctl -d to disable the firewall temporarily, allowing unrestricted access to the WebGUI for initial configuration. The system confirmed that the packet filter (pf) was disabled. After completing the setup, I re-enabled the firewall by running pfctl -e to secure the system again.



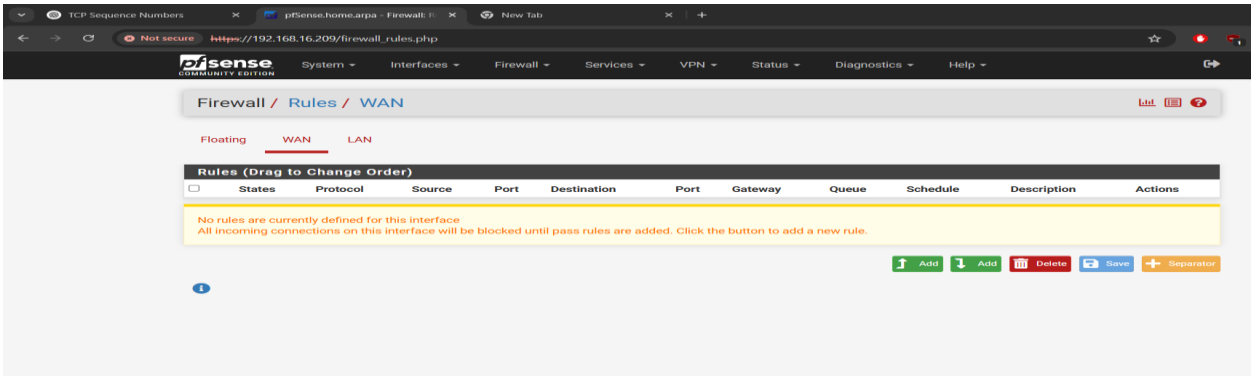
Step 12: Accessing the pfSense Dashboard

After disabling the firewall temporarily, I opened a web browser and navigated to the pfSense WAN IP address (e.g., https://192.168.16.209). The pfSense login page appeared. I logged in using the default credentials and was directed to the **Status Dashboard**, confirming successful access to the pfSense WebGUI for further configuration.



Step 13: Navigating to Firewall Rules

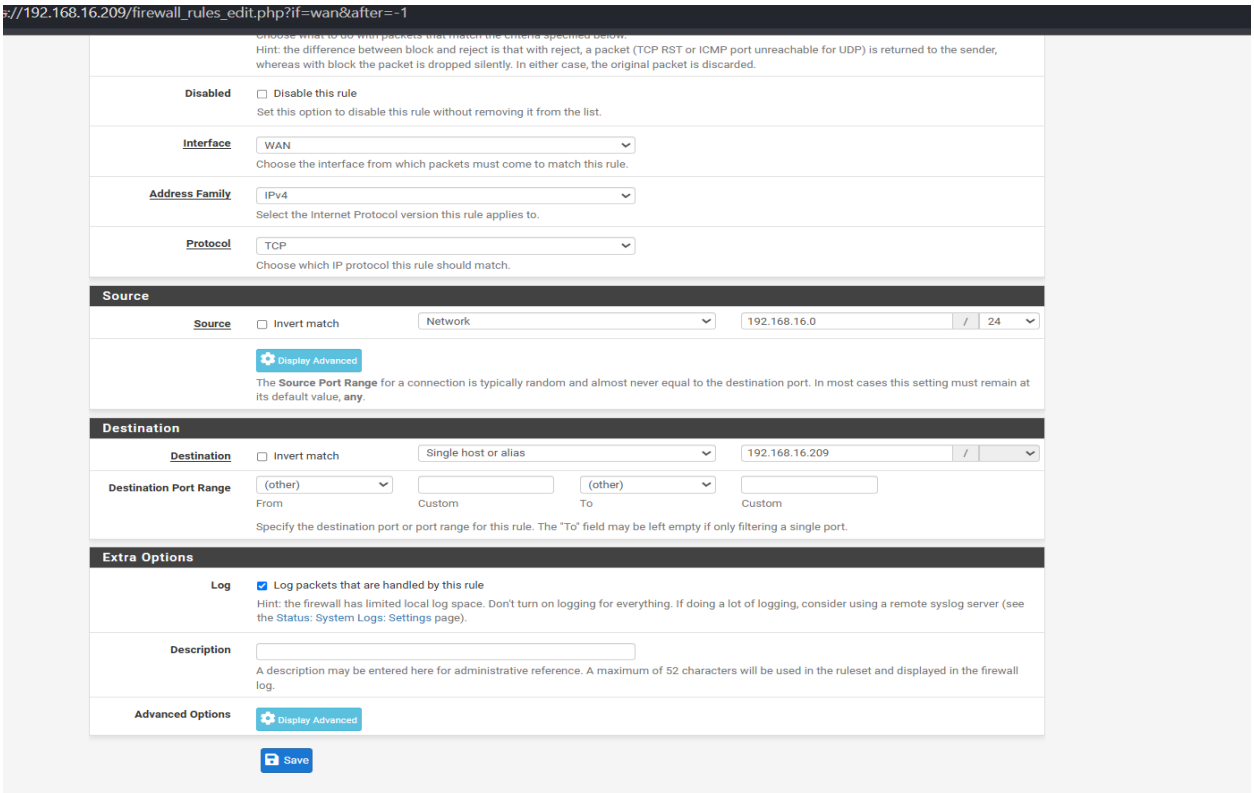
After logging into the pfSense WebGUI, I accessed the **Firewall** menu from the dashboard sidebar. Under this menu, I selected **Rules** to view and manage the firewall rules. This section allows configuring which traffic is allowed or blocked on various interfaces, such as WAN or LAN.



Step 14: Adding Firewall Rules on WAN Interface

In the pfSense WebGUI, I went to **Firewall > Rules > WAN** to add new rules. These rules allow

secure remote management by permitting HTTPS (port 443) traffic from trusted IPs on the WAN side. This helps control and restrict incoming connections while maintaining secure access.

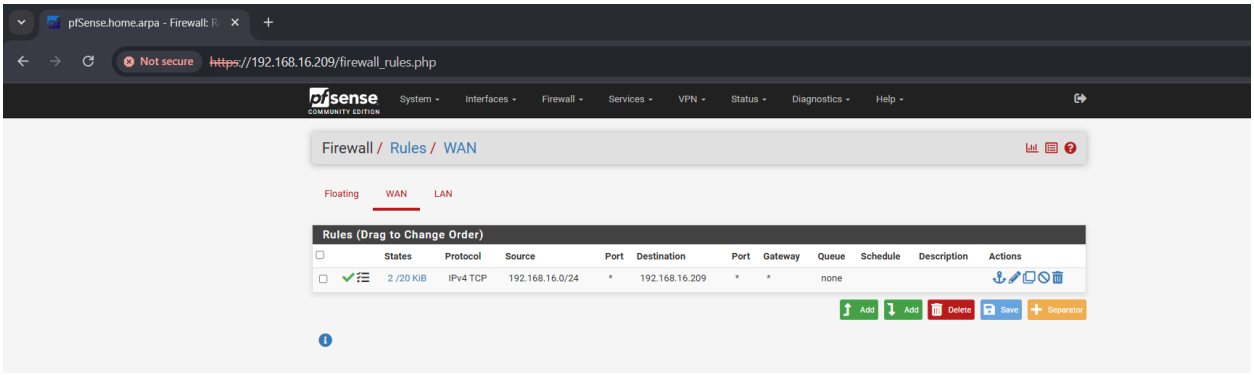


Step 15: Creating a Firewall Rule to Allow Traffic from Source Network to Specific Destination

I configured a firewall rule on the WAN interface to **allow traffic from the source network 192.168.16.0/24 to the specific destination IP 192.168.16.209**. The rule was set to **pass** matching packets to permit communication. Key settings included:

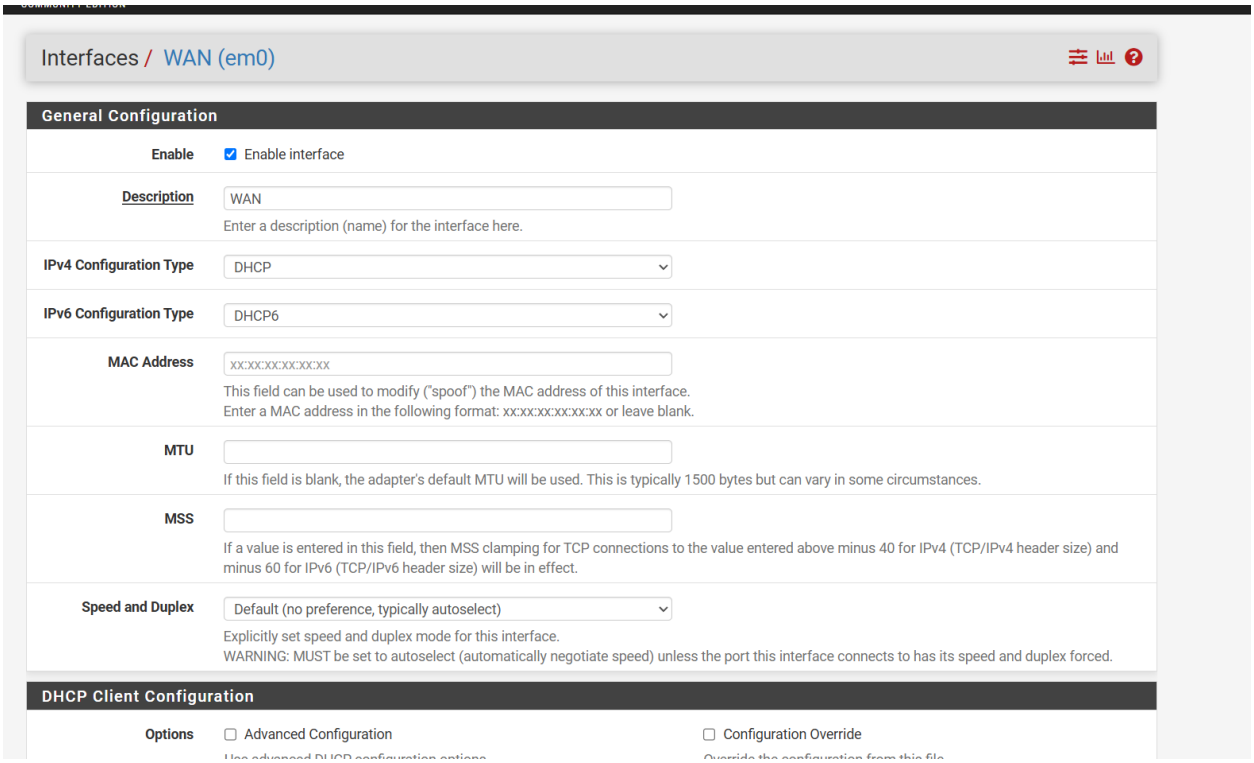
- Interface: WAN
- Protocol: Any (or specify if needed)
- Source: Network 192.168.16.0/24
- Destination: Single host 192.168.16.209
- Port Range: Default (any)
- Action: Pass

The rule was enabled to ensure traffic filtering applied correctly. This setup allows controlled access from the trusted local subnet to the designated IP address.



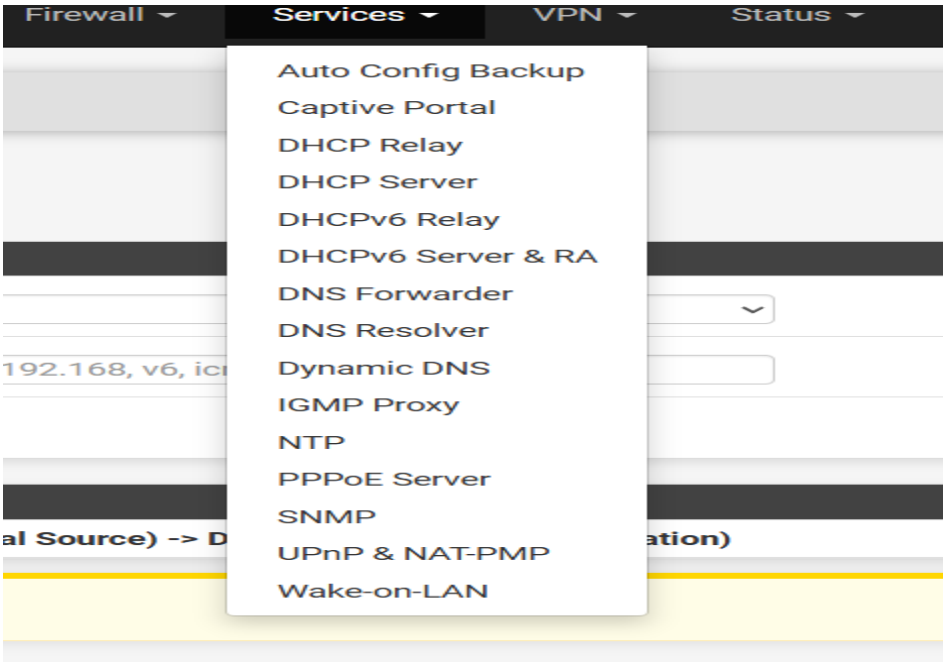
Step 16: Verifying Firewall Rules on WAN and LAN Interfaces

I navigated to **pfSense > Firewall > Rules** and confirmed that the new rule allowing traffic from the LAN network (192.168.16.0/24) to the destination IP (192.168.16.209) was successfully created under the **WAN** interface rules. This ensures proper filtering and access control for incoming traffic.



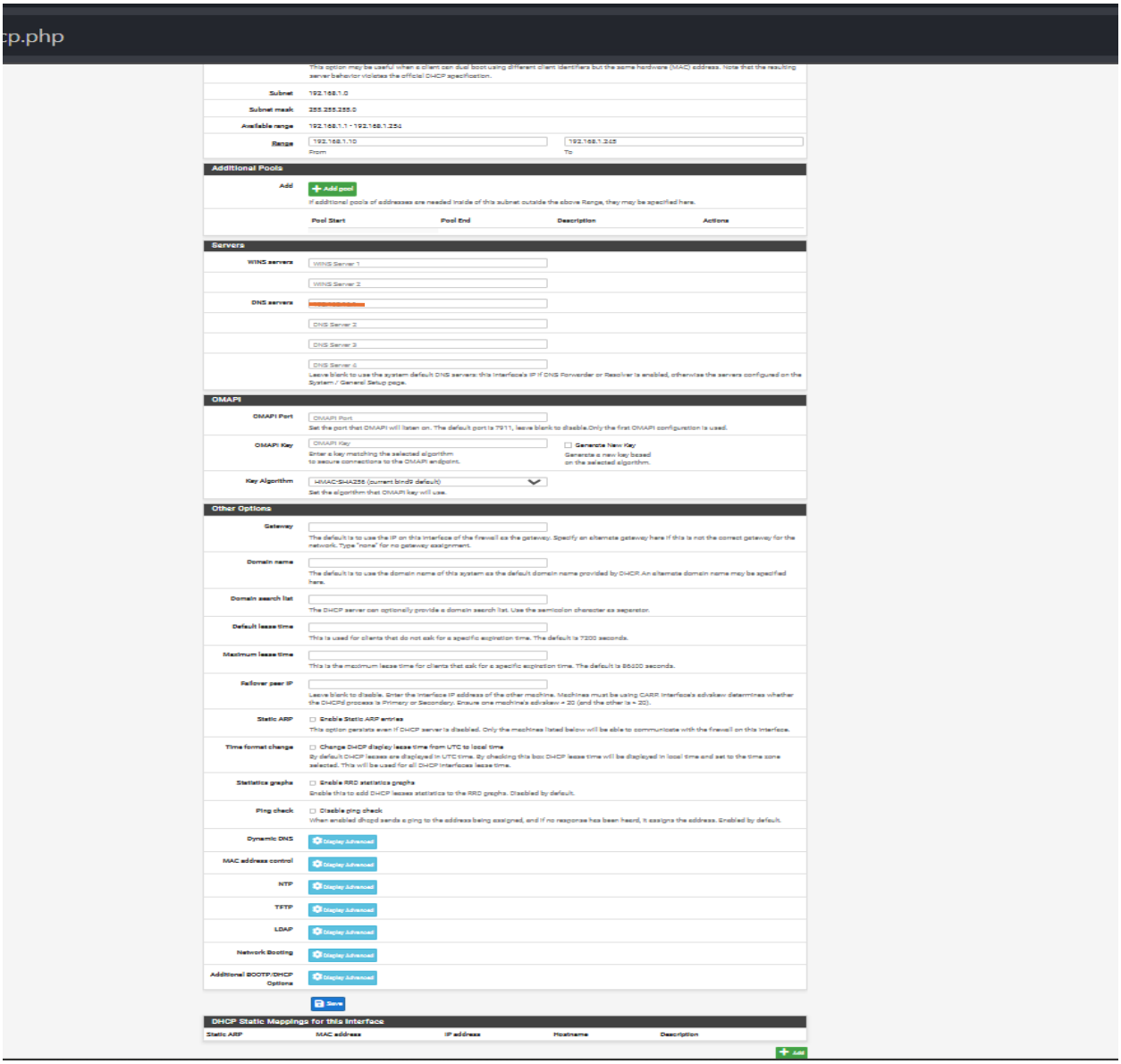
Step 17: Configuring WAN Interface Firewall Rules

I accessed **Firewall > Rules > WAN** in pfSense WebGUI and created a rule to allow traffic from the source network 192.168.16.0/24 to the destination IP 192.168.16.209. This rule ensures proper access control and allows the required communication through the WAN interface.



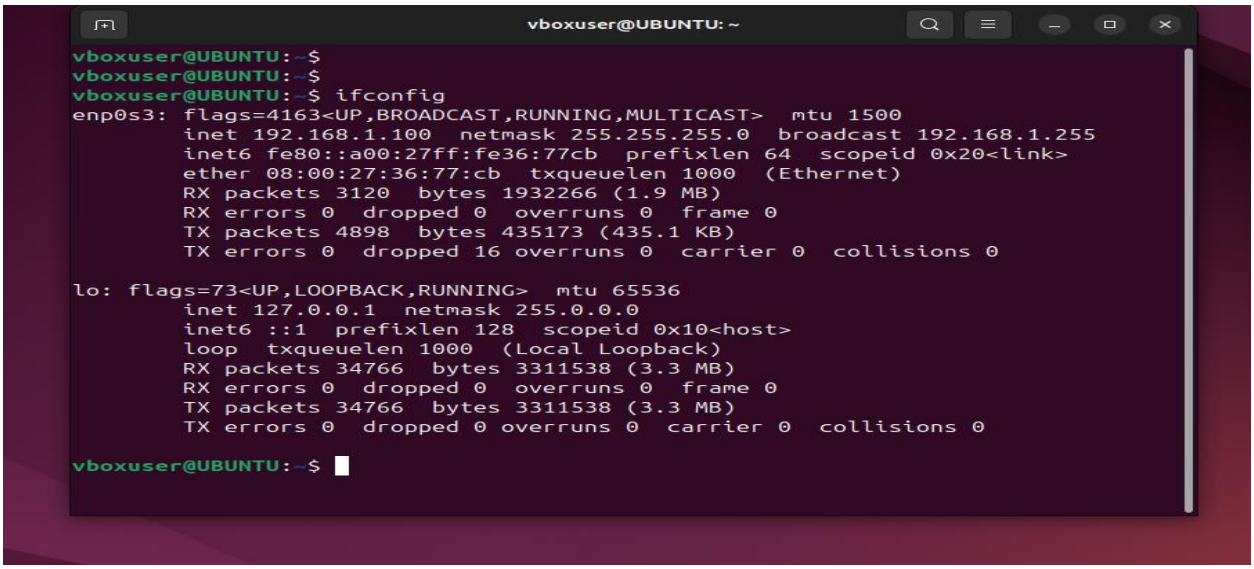
Step 18: Assign IP to LAN Interface Using DHCP

By enabling DHCP on the LAN interface, devices like the Ubuntu machine connected to the internal network will automatically receive an IP address. This allows the Ubuntu system to communicate on the LAN without manual IP configuration, simplifying network setup and management



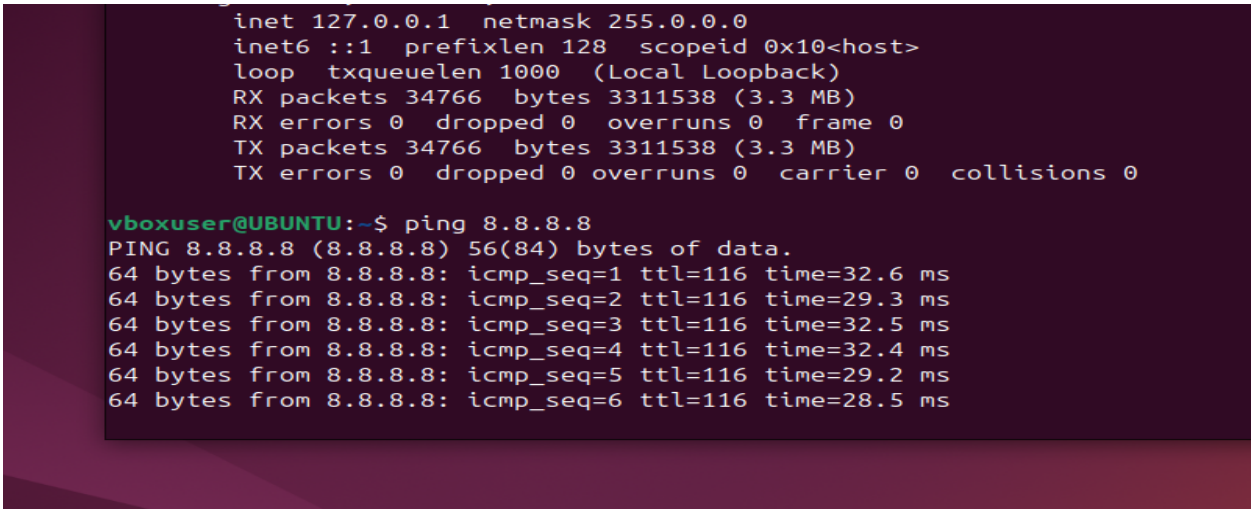
Step 19: Configure LAN Subnet and DNS Settings

We set the LAN subnet to **192.168.1.0** with a subnet mask of **255.255.255.0**, providing an IP range from **192.168.1.1** to **192.168.1.254** for devices on the network. For DNS, we use the home router's DNS server to resolve domain names, ensuring proper internet connectivity and name resolution for devices on the LAN.



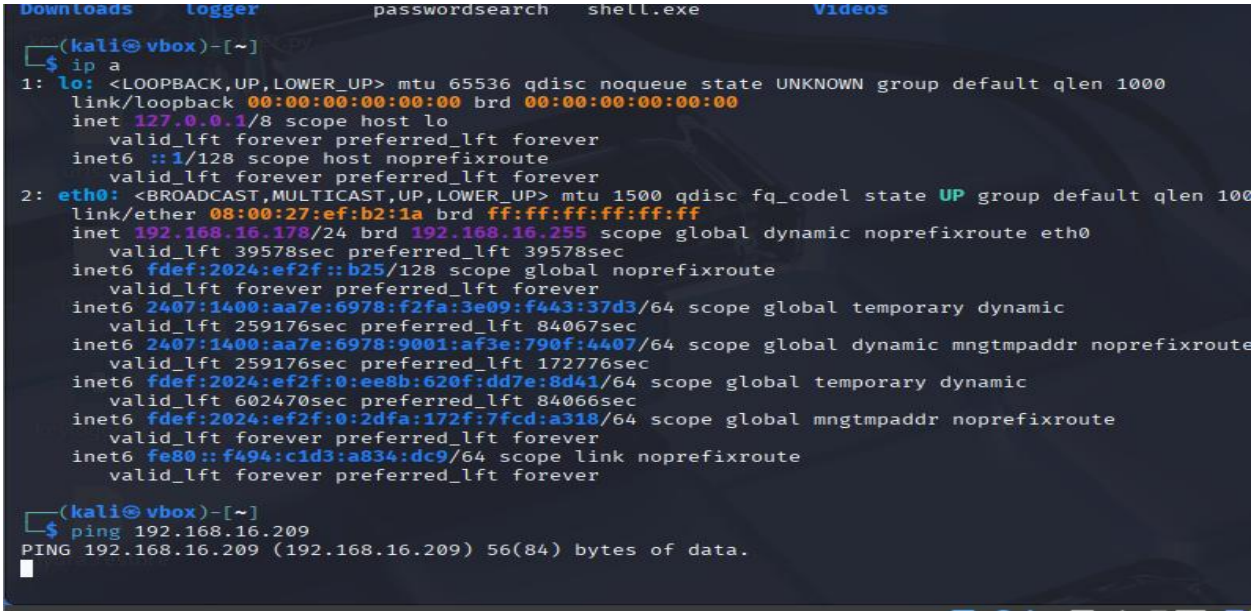
Step 20: Verify Ubuntu Network Configuration

On the Ubuntu VM, the network interface **enp0s3** is configured on the internal network with IP **192.168.1.100** (via DHCP), subnet mask **255.255.255.0**, and is successfully transmitting and receiving packets. This confirms the DHCP service from pfSense is working correctly, assigning IPs in the LAN subnet. The pfSense firewall is connected with two adapters: one bridged (WAN) and one internal (LAN).



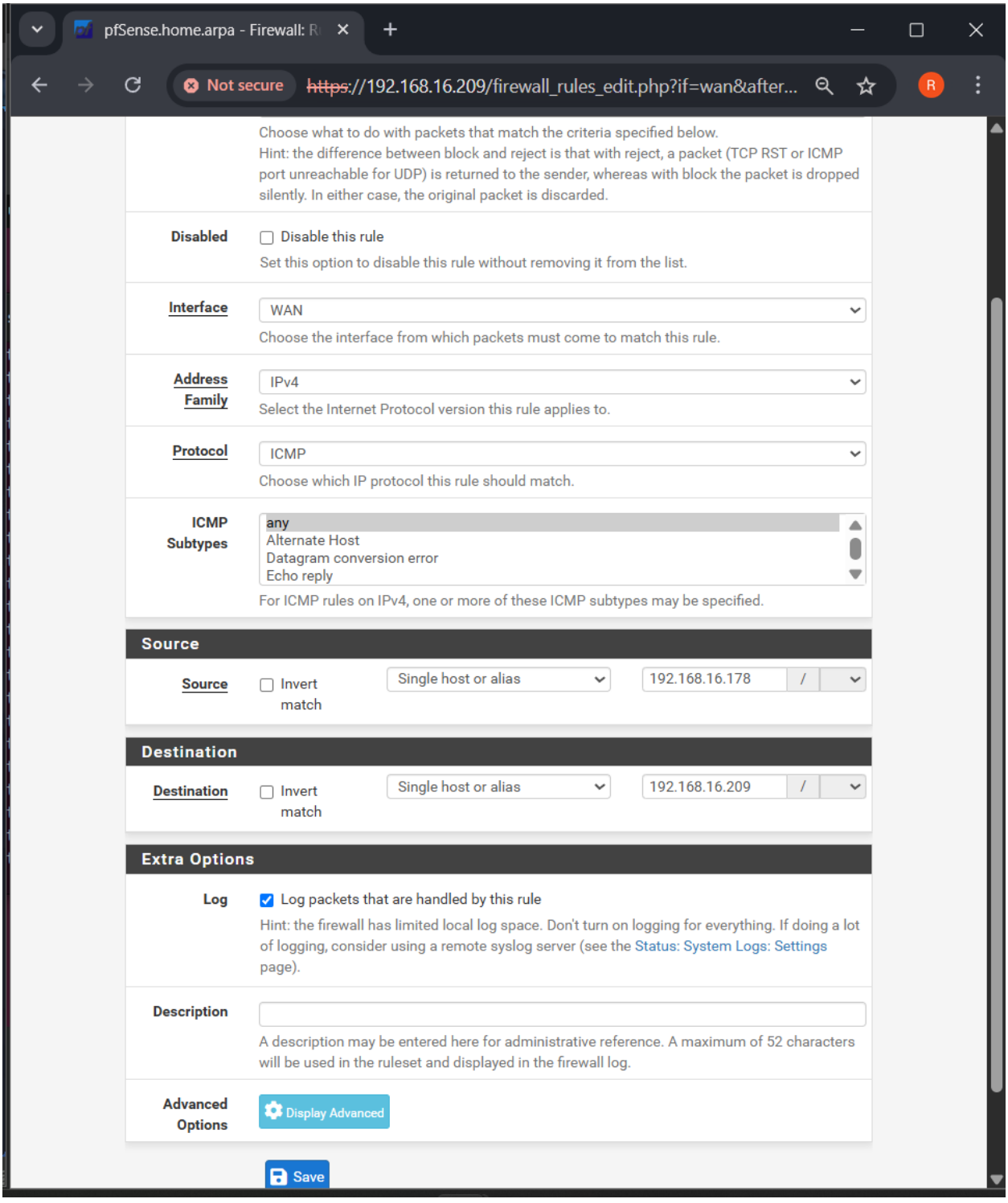
Step 21: Verify Internet Connectivity from Ubuntu VM

The Ubuntu VM successfully pings the external DNS server **8.8.8.8**, showing stable responses with low latency and no packet loss. This confirms that the Ubuntu VM has proper internet access through the pfSense firewall.



Step 22: Kali Linux Unable to Connect to pfSense LAN

The Kali Linux machine cannot reach the pfSense LAN IP (192.168.16.209) because no firewall rule/policy currently allows traffic from Kali's network. This step highlights the need to create appropriate firewall rules to enable communication.



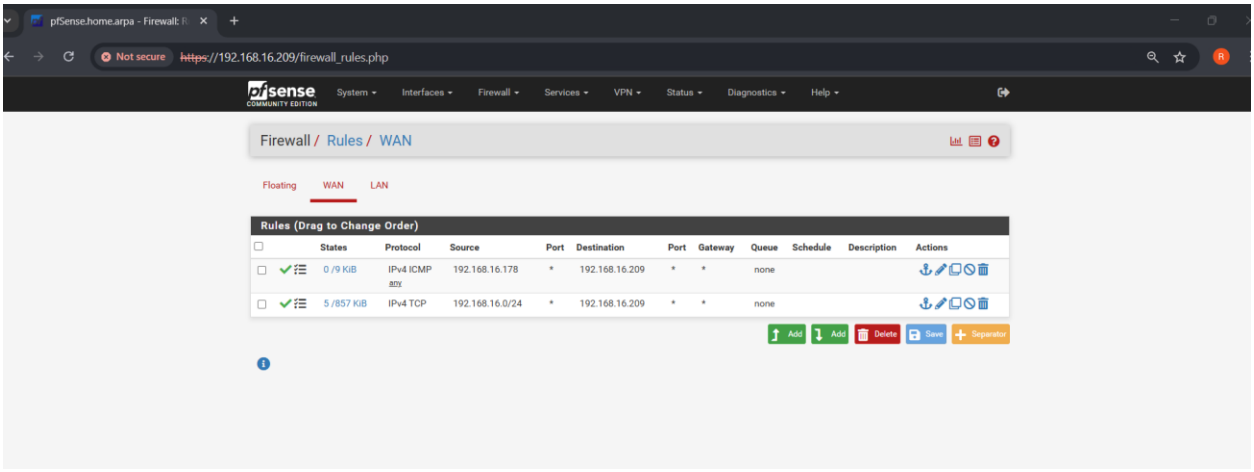
Step 23: Create Firewall Rule to Allow ICMP from Kali to pfSense

Add a new firewall rule on the pfSense WAN or LAN interface to allow ICMP traffic from Kali's IP (192.168.16.178) to pfSense LAN IP (192.168.16.209).

- Protocol: ICMP (IPv4)

- Source: Single host or alias — 192.168.16.178
- Destination: Single host or alias — 192.168.16.209
- Enable logging (optional) to monitor allowed packets
- Add a descriptive note like "Allow ICMP from Kali to pfSense"

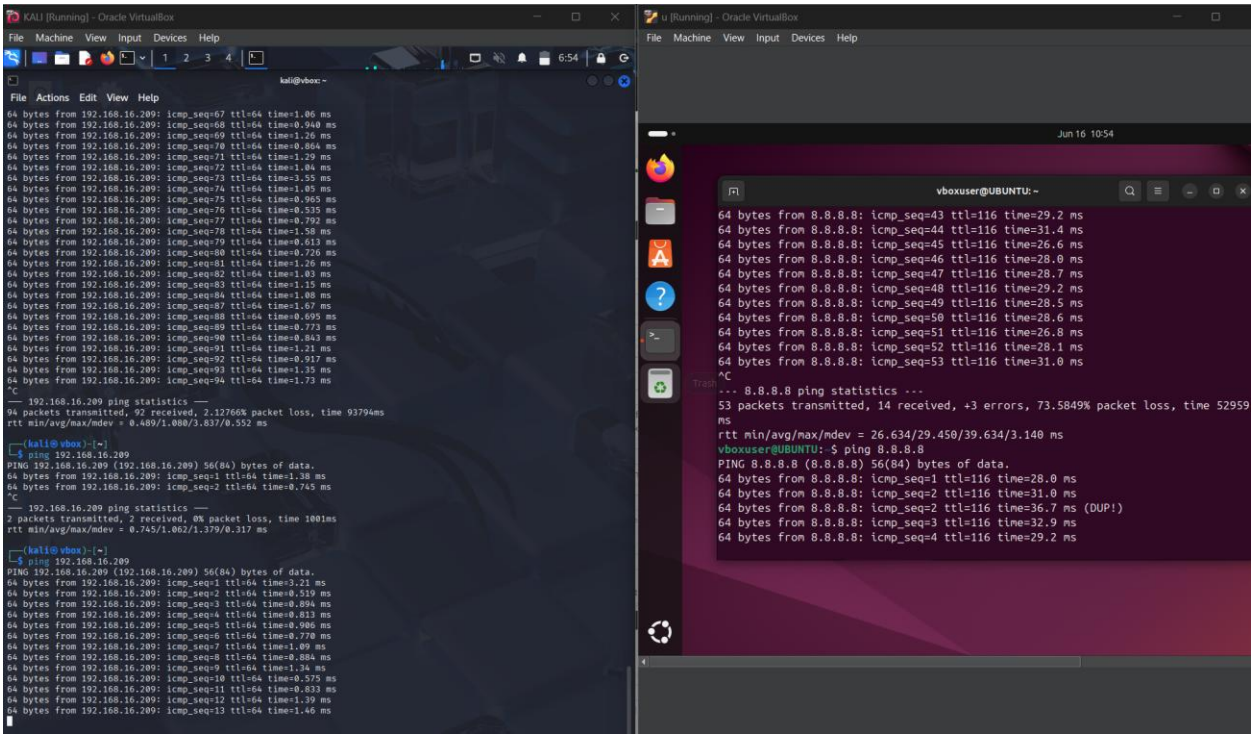
This rule enables ping and network diagnostics between Kali and the pfSense firewall.



Step 24: Review and Configure Firewall Rules on pfSense

- Navigate to **Firewall > Rules** and check the rules under **WAN** and **LAN** interfaces.
- Ensure there is a rule allowing traffic from the source IP (e.g., 192.168.16.178 or subnet 192.168.16.0/24) to the destination IP (192.168.16.209 — pfSense LAN).
- Confirm the rules specify correct protocols (e.g., TCP, ICMP) and ports if applicable.
- Use descriptions to clearly identify each rule’s purpose for easier management.
- Adjust rule order if needed, as pfSense processes rules top-down.

This step ensures proper traffic flow through the firewall by configuring and verifying relevant rules.

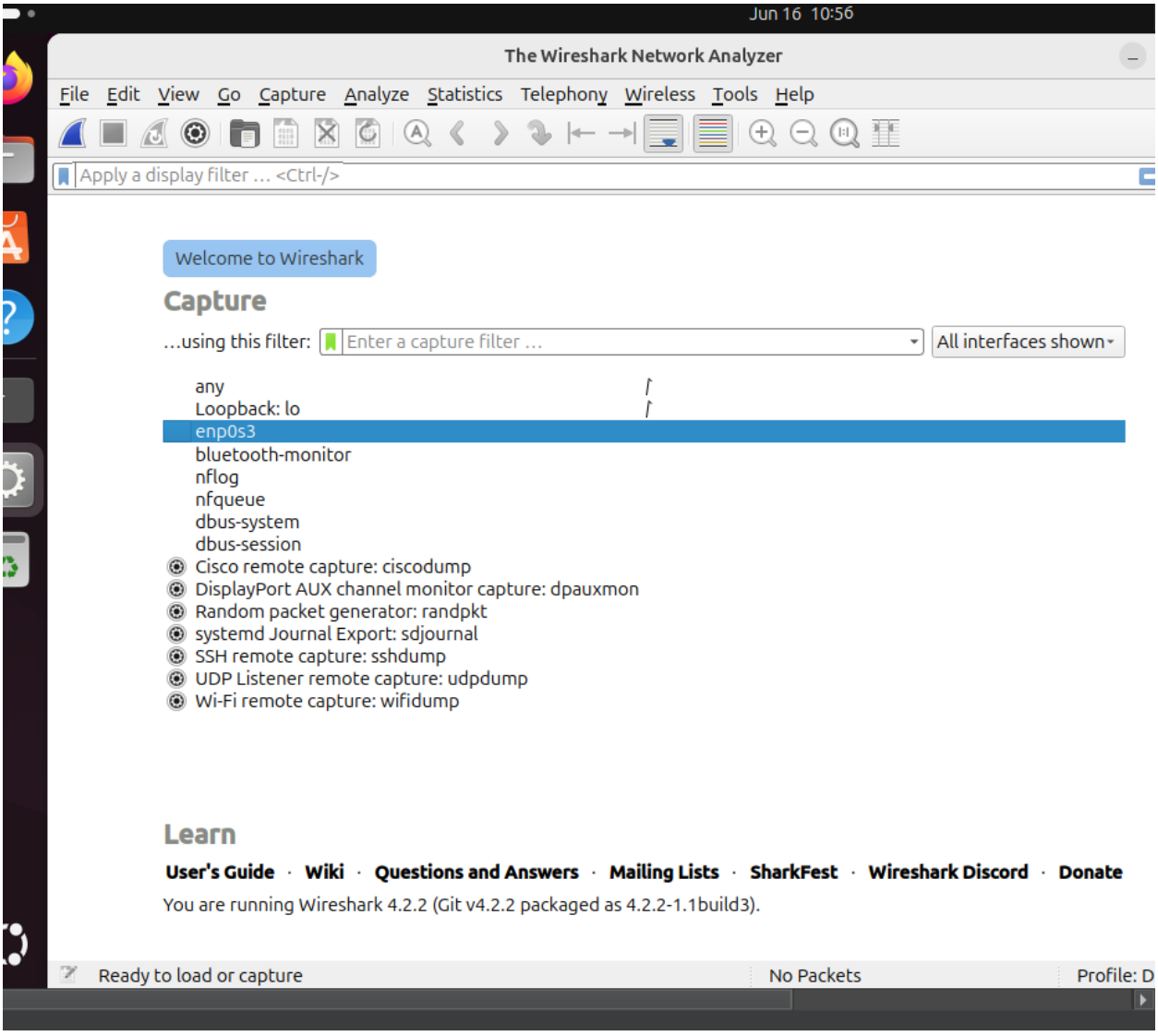


Step 25: Enable Communication Between Kali, pfSense Firewall, and Ubuntu

- Configure firewall rules on pfSense to allow traffic from Kali (e.g., 192.168.16.178) to the pfSense LAN IP (192.168.16.209) and to Ubuntu's IP on the internal network (e.g., 192.168.1.100
).
- Ensure Ubuntu is configured to use DHCP or a static IP within the subnet managed by pfSense.
- Verify Kali and Ubuntu can ping pfSense and each other by allowing ICMP traffic in firewall rules.
- Confirm network interfaces on pfSense (bridge and internal) are properly configured to route traffic between Kali and Ubuntu.

8. Demonstration of a Basic DoS Attack within the Virtual Network Environment

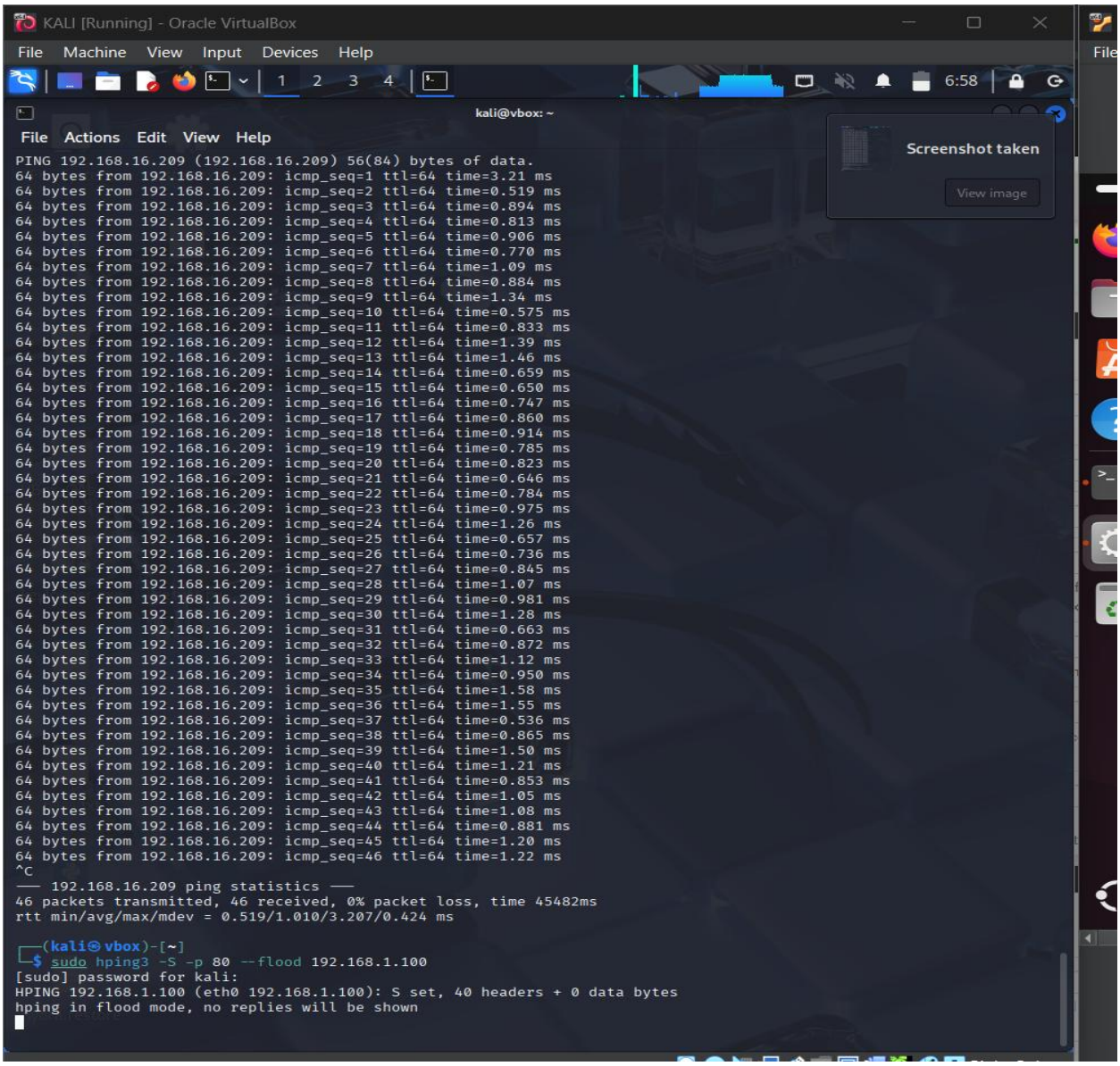
In this step, we demonstrate a simple Denial of Service (DoS) attack within the virtual network setup. We have already deployed a firewall rule allowing ICMP traffic from the Kali attacker to the Ubuntu victim, enabling them to successfully ping each other. This connectivity ensures that Kali can now perform DoS attacks by flooding ICMP requests to the Ubuntu machine, simulating network disruption for testing purposes.



Step 1: Launch Wireshark on Ubuntu

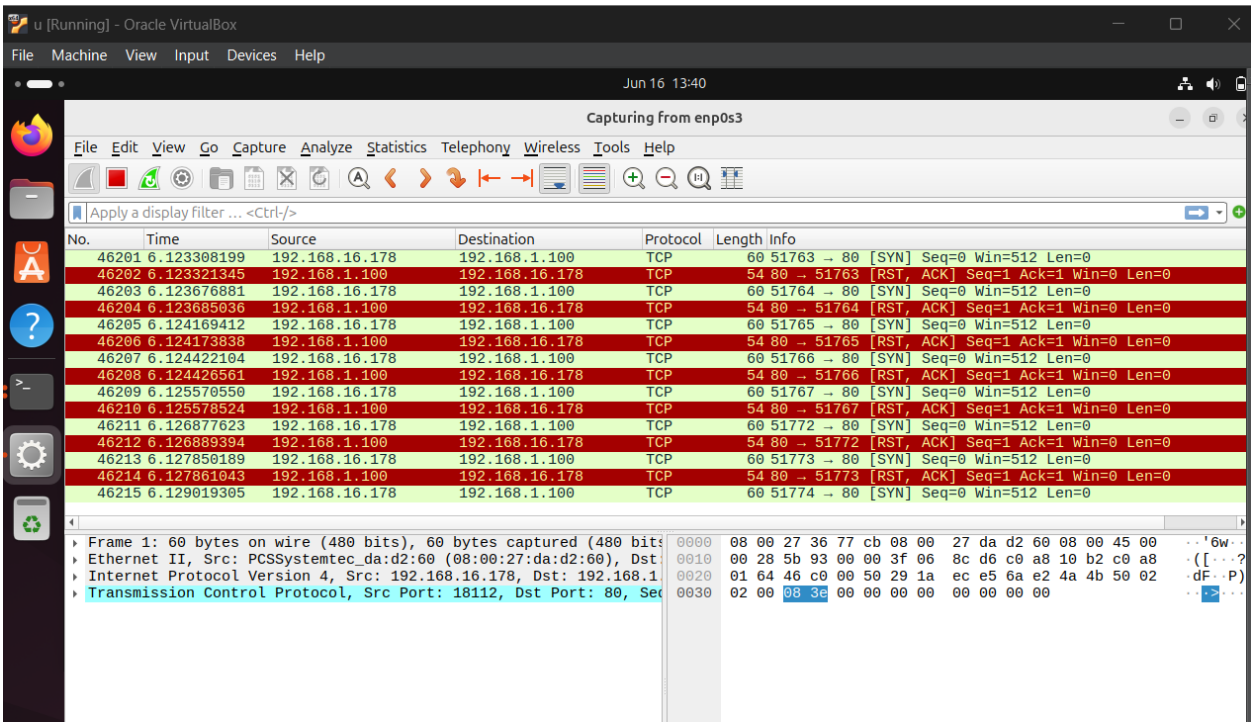
Open Wireshark from the applications menu or by typing wireshark in the terminal. Once

opened, you will see the main Wireshark window displaying available network interfaces. This tool will be used to capture and analyze network traffic during the DoS attack demonstration.



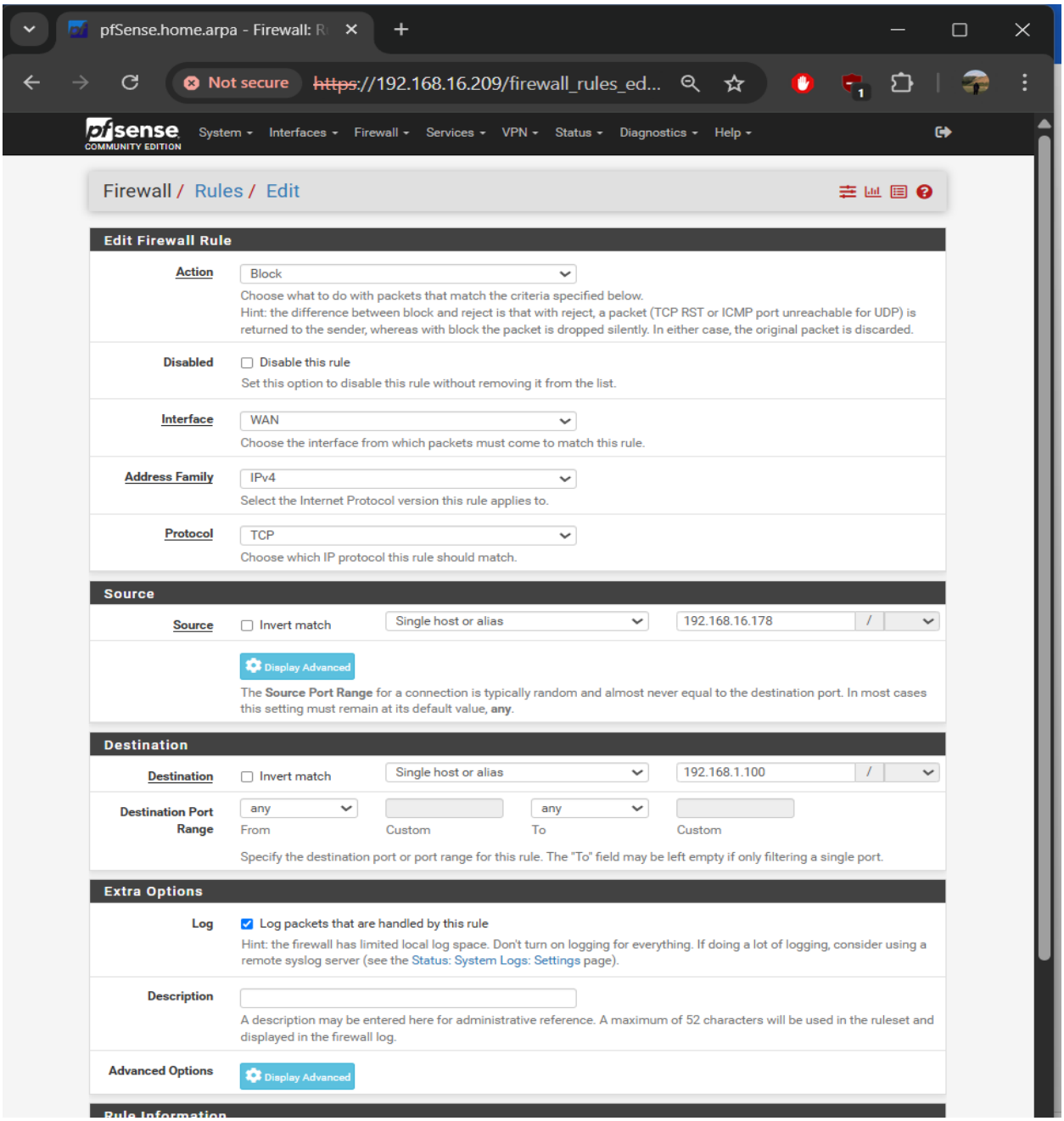
Step 2: A Basic DoS Attack Using Ping and hping3 Flood

In this step, we perform a simple DoS attack by flooding the target IP 192.168.16.209 with ICMP echo requests (ping) and TCP SYN packets using the hping3 tool. First, a normal ping test shows the target's response times and packet statistics. Then, using hping3 with the --flood and -S (SYN flag) options on port 80, a rapid stream of TCP SYN packets is sent to overwhelm the target's resources, simulating a SYN flood attack. This helps illustrate how a DoS attack can disrupt network availability.



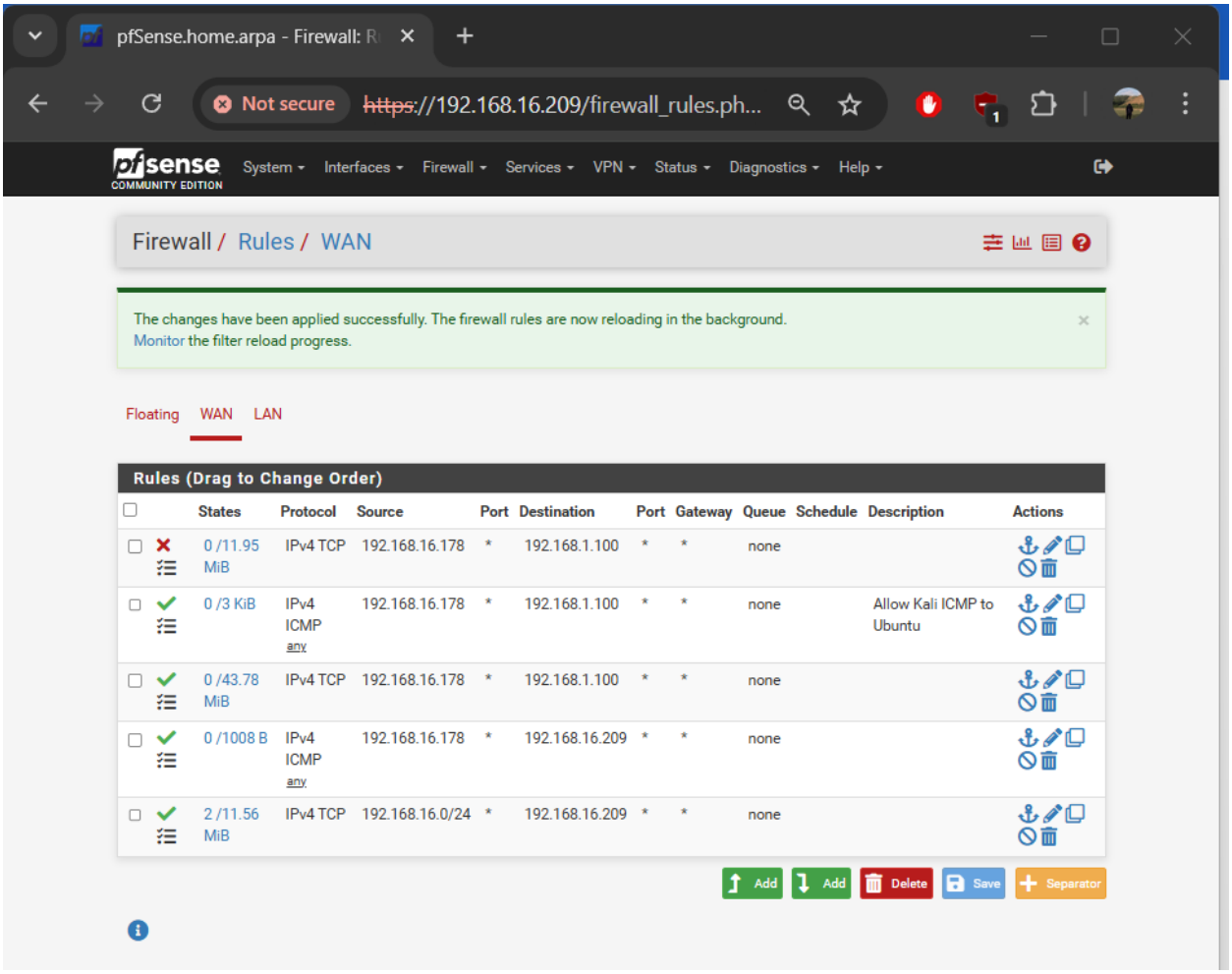
Step3: Capturing Attack Traffic

During this step, we capture the network traffic while the attack is in progress using Wireshark on the relevant interface. The capture reveals TCP packets with SYN flags set, indicating the initiation of multiple connection requests typical in a DoS attack. The detailed packet information includes source and destination IPs, ports, sequence numbers, and protocol headers, allowing us to analyze the attack pattern and its impact on the network.



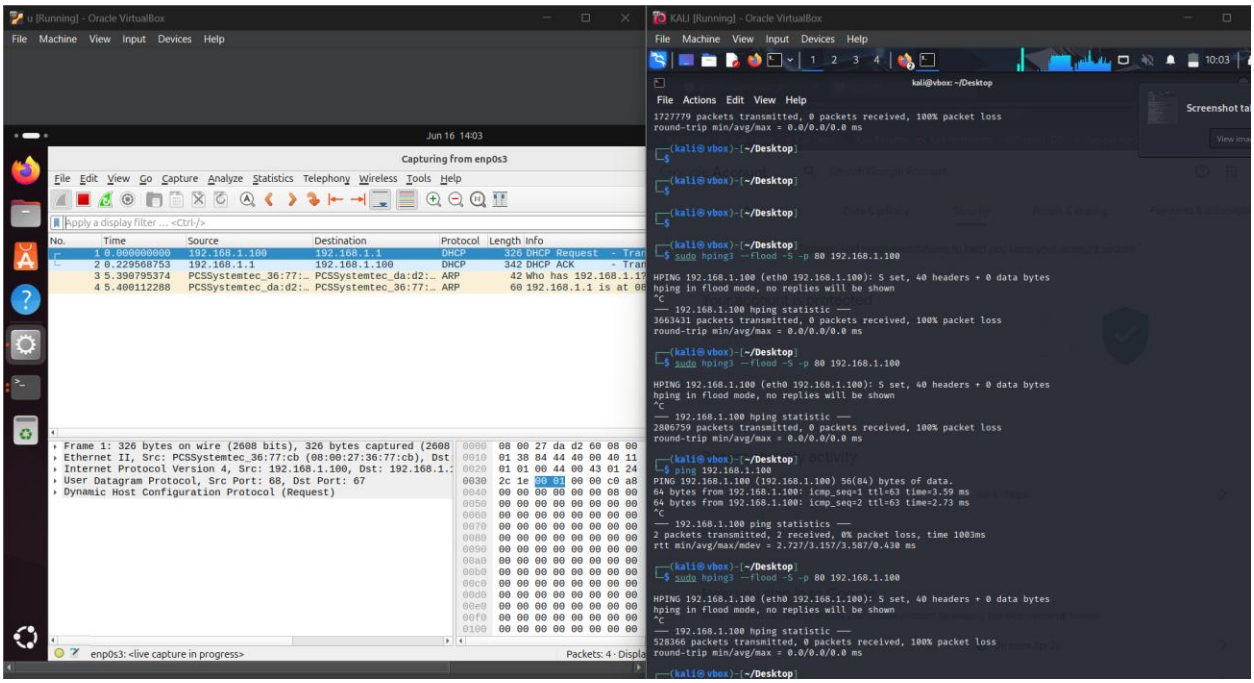
Step 4: Adding a Firewall Rule to Block DoS Attack

In this step, we create a firewall rule on pfSense to block traffic from the Kali Linux attacker IP to the Ubuntu victim IP. The rule is set to block the specific source IP (Kali) targeting the destination IP (Ubuntu) on all ports and protocols, effectively stopping the DoS attack. Optional logging is enabled to monitor blocked packets for further analysis.



Step 5: Deploying and Verifying Firewall Rule on WAN

We deployed a firewall rule on the pfSense WAN interface to block traffic from the attacker IP (192.168.16.178) targeting the victim IP (192.168.16.209). After applying the rule, the firewall successfully reloaded the configuration. Monitoring the rule shows active state entries, confirming the rule is enforced and blocking the specified traffic as intended.



Step 6: Limited Traffic Observed After Adding Firewall Rule

After adding the firewall rule to allow traffic from the Kali attacker to the Ubuntu victim, only minimal network traffic was observed. This indicates that although basic connectivity was established, no significant attack activity or malicious packets were detected on the network at this stage. Further configuration or attack steps may be required to generate noticeable traffic.

9.Pros and Cons

Category	Pros (Strengths)	Cons (Challenges & Fixes)
Setup	-Realistic network segmentation (WAN/LAN). -Isolated VMs (Kali, Ubuntu, pfSense) for safe testing.	-Initial DHCP misconfiguration on LAN. **-Fix**: Enabled DHCP server on pfSense (192.168.1.1/24).
Attack Simulation	-Successfully demonstrated DoS attacks using hping3 and Wireshark for analysis.	-SYN flood bypassed rules initially due to incorrect rule order.
Firewall Rules	-Granular control over traffic (ICMP/TCP blocking, IP-based filtering).	-Rule misplacement caused conflicts (e.g., allow rules overriding blocks).
C**onnectivity**	-Ubuntu accessed the internet via pfSense NAT; Kali bridged to host network.	-Kali lost connectivity when host WiFi changed (required adapter reconfiguration).
Scalability	-Easy to expand (e.g., adding IDS like Suricata or VPN support).	-Manual traffic logging made forensic analysis tedious.
Ease of Use	-pfSense WebGUI is user-friendly for beginner	-Default WAN rules blocked private IPs, requiring manual adjustment.

10. Future Enhancements

1. **Advanced Security Features**
    - Suricata IDS/IPS: Add intrusion detection for deeper attack analysis.
    - GeoIP Blocking: Block traffic from high-risk regions.
    - Rate Limiting: Throttle ICMP/SYN packets to prevent floods.
2. **Network Resilience**
    - Failover WAN: Add a secondary ISP link for redundancy.
    - VPN Integration: Secure remote management (OpenVPN/WireGuard).
3. **Monitoring & Logging**
    - ELK Stack: Centralize logs for attack forensics.
    - Grafana Dashboards: Visualize traffic patterns in real-time.

11. Critical Lessons Learned

- **Rule Order Matters**: pfSense processes rules top-down; place blocks before allows.
- **Test Rules Incrementally**: Avoid lockouts by testing one rule at a time.
- **Document Changes**: Label rules (e.g., "Allow Admin HTTPS") for easier troubleshooting.
- **Backup Configs**: Export pfSense settings before major changes.

12. Conclusion

This lab successfully demonstrated the setup and configuration of a pfSense-based firewall environment within Oracle VirtualBox, simulating a real-world network with distinct WAN and LAN segments. By deploying Kali Linux as the attacker, Ubuntu Desktop as the victim, and pfSense as the firewall, the lab effectively showcased the execution, detection, and mitigation of a Denial-of-Service (DoS) attack using tools like hping3 and Wireshark. The step-by-step configuration of pfSense firewall rules highlighted its robust capabilities in controlling network traffic and securing internal hosts against external threats. Key lessons learned, such as the importance of rule order, incremental testing, and clear documentation, underscore the practical challenges and solutions in firewall management. The lab provided valuable hands-on experience for cybersecurity enthusiasts, reinforcing the critical role of perimeter defense in protecting network infrastructure.