

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

*Author: Rishav Kumar Thapa*

*Platform: Oracle VirtualBox*

*Victim OS: Ubuntu*

*Server OS: Pfsense Firewall*

*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.

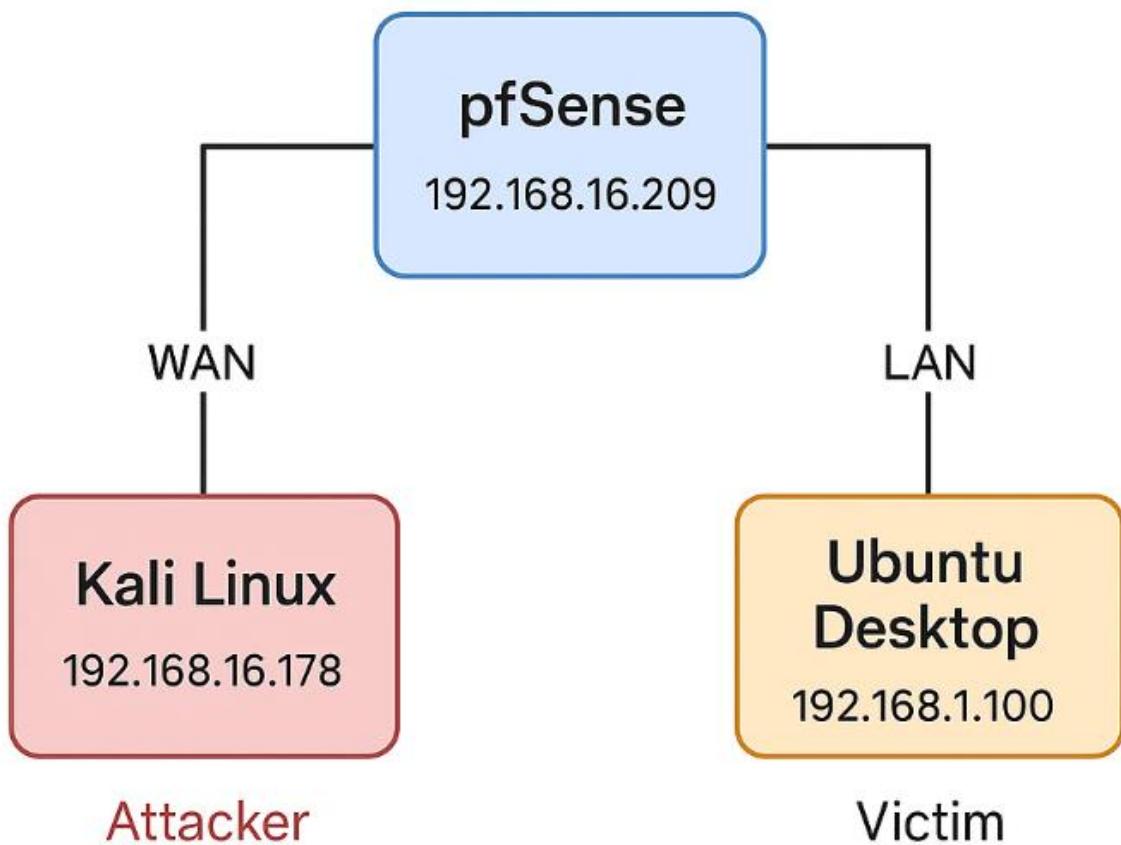
## **2. Objectives**

- 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)

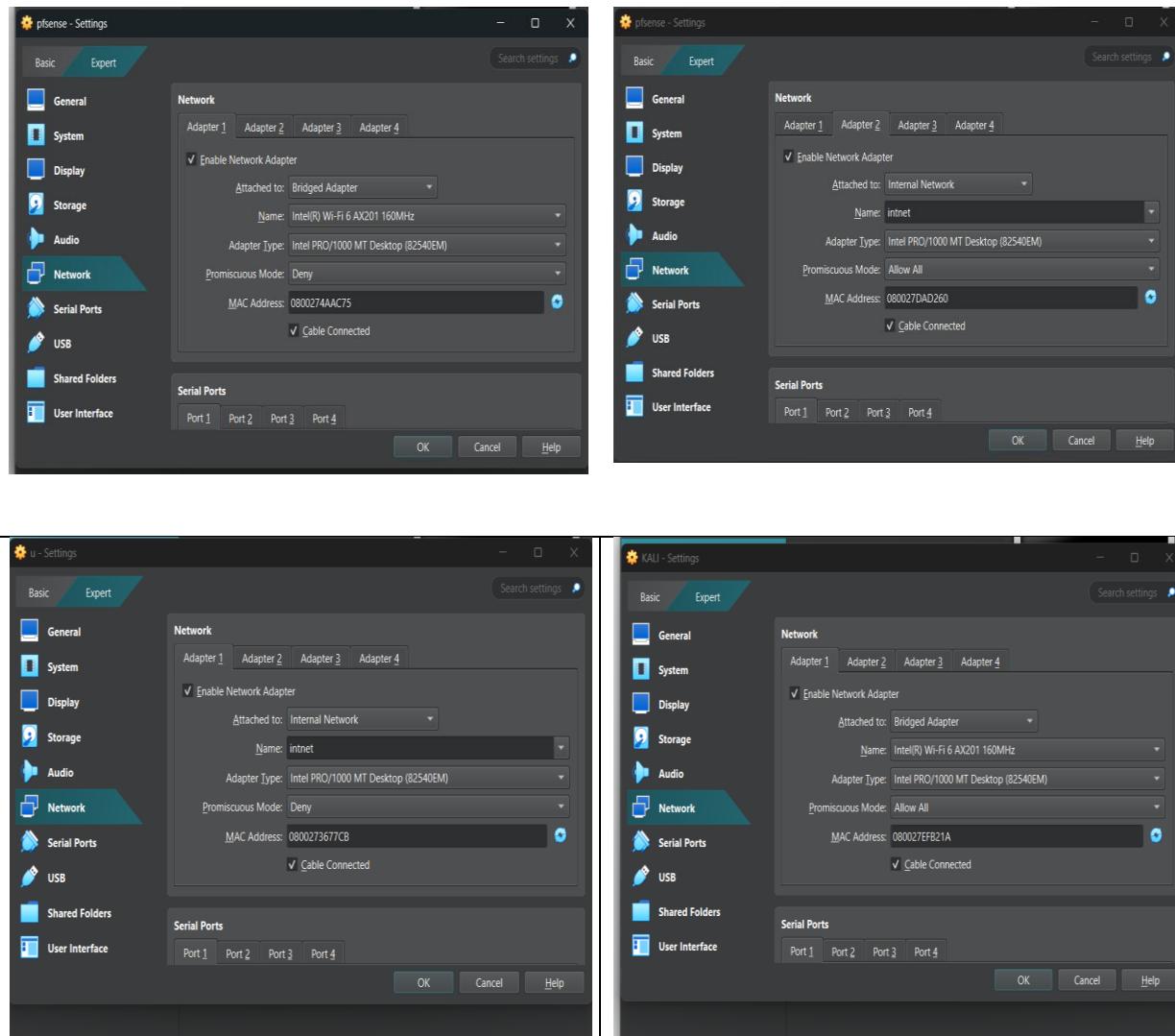
- OS Type: Debian 64-bit
- Network:
  - Adapter 1: Bridged

### 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.

```

pfSense 2.6.0-RELEASE amd64 Mon Jan 31 19:57:53 UTC 2022
Bootup complete

FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: f5916e9abb981186e14b

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***

WAN (wan)      -> em0      -> v4/DHCP4: 192.168.16.209/24
                           v6/DHCP6: fdef:2024:ef2f:0:a00:27ff:fe4a:ac75/
64
LAN (lan)      -> em1      -> v4: 192.168.1.1/24

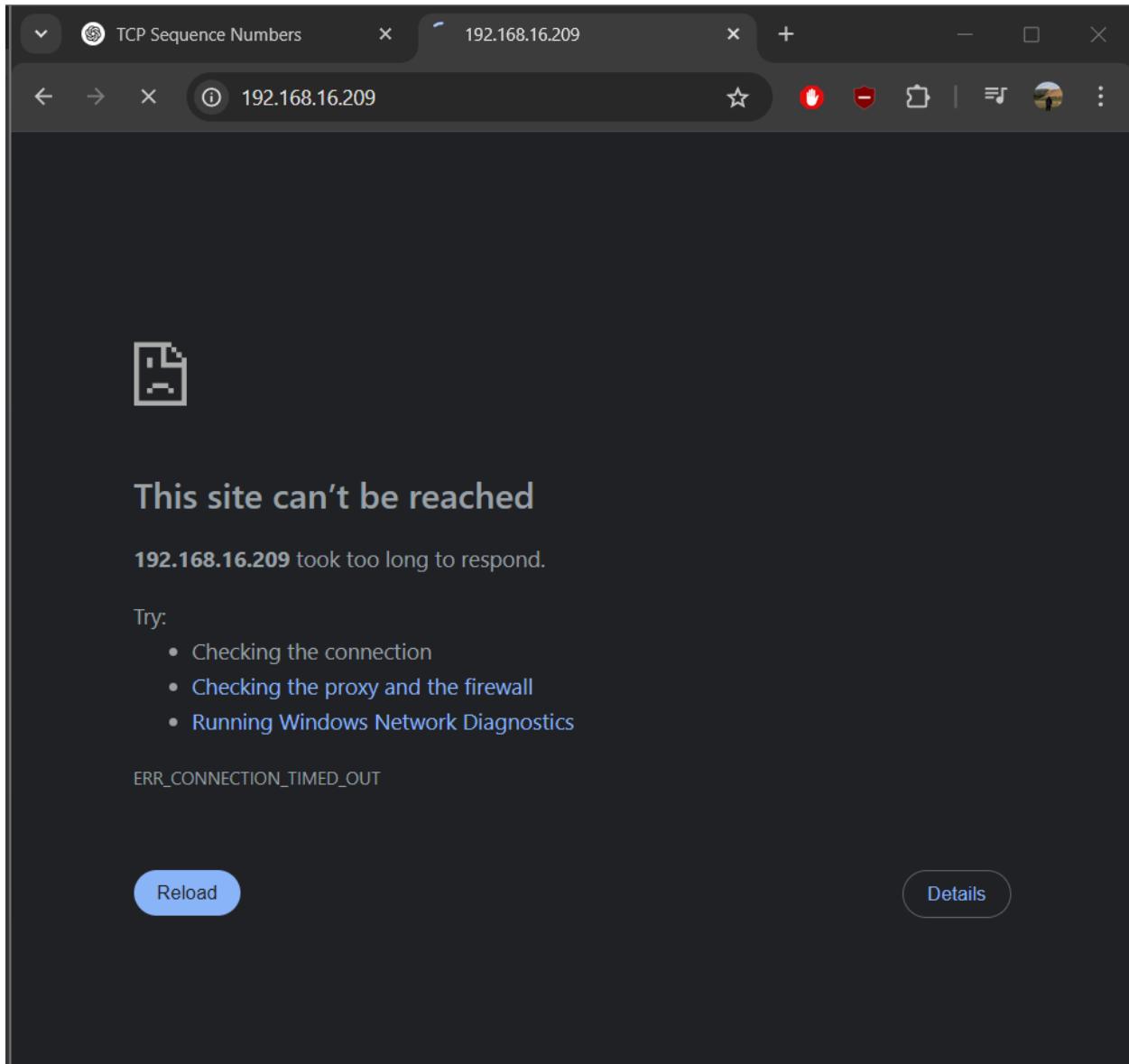
0) Logout (SSH only)          9) pfTop
1) Assign Interfaces          10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) PHP shell + pfSense tools
4) Reset to factory defaults   13) Update from console
5) Reboot system               14) Enable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

Enter an option: 

```

## 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

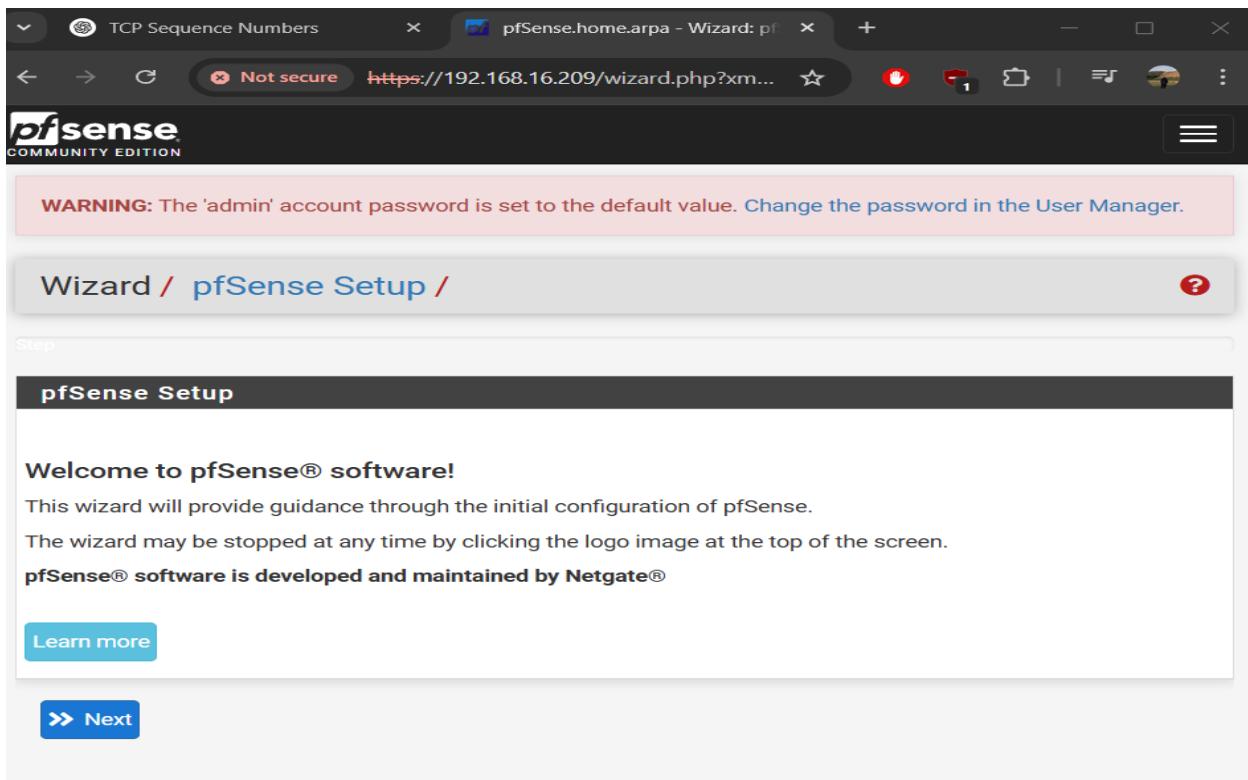
While attempting to access the device at IP address **192.168.16.209** via a web browser,

```
VirtualBox Virtual Machine - Netgate Device ID: f5916e9abb981186e14b
*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***
WAN (wan)      -> em0      -> v4/DHCP4: 192.168.16.209/24
64 LAN (lan)    -> em1      -> v4: 192.168.1.1/24
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]# pfctl -d
pf disabled
[2.6.0-RELEASE][root@pfSense.home.arpa]#
```

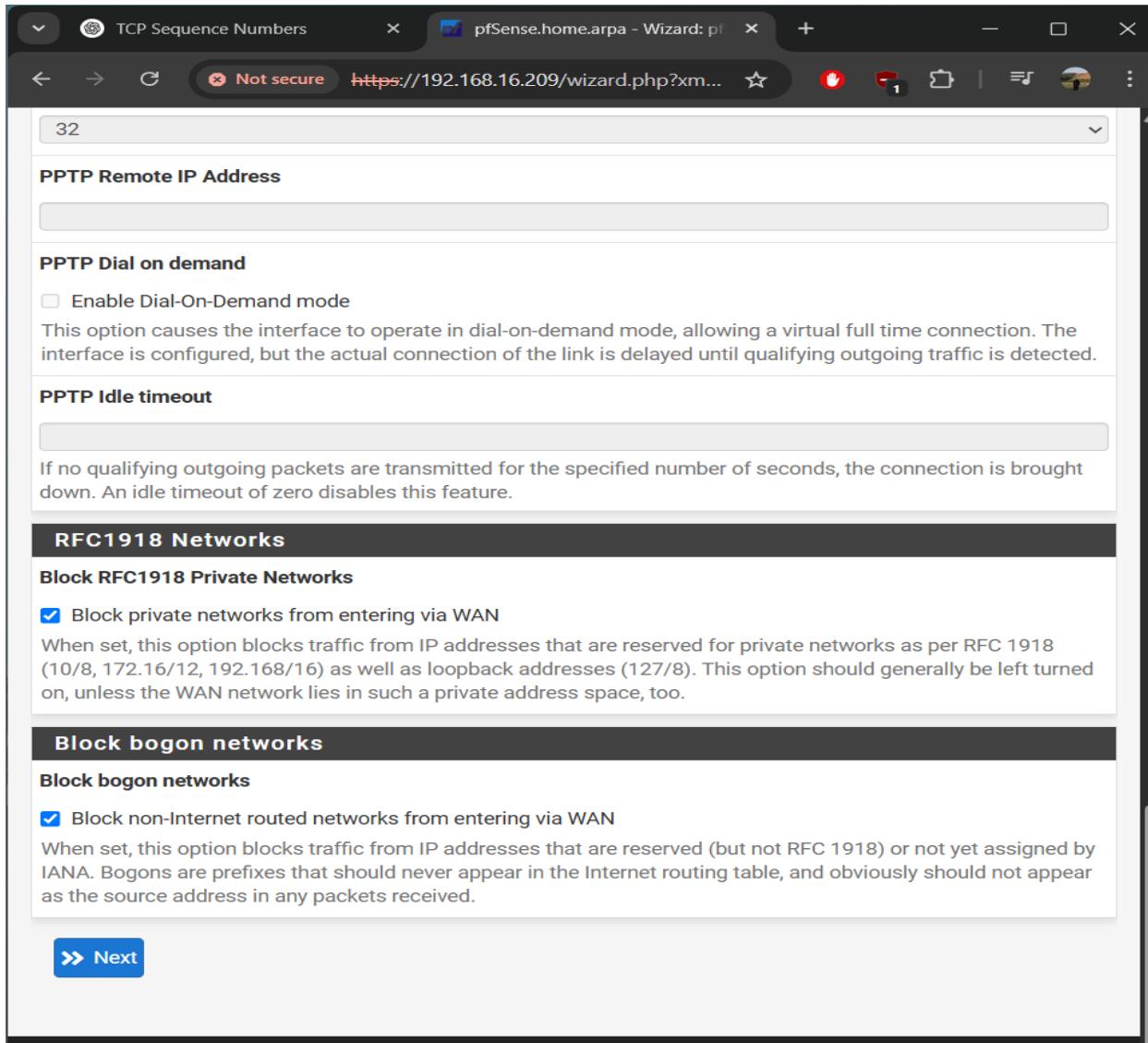
#### Step 4: Disable pfSense Firewall Temporarily

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

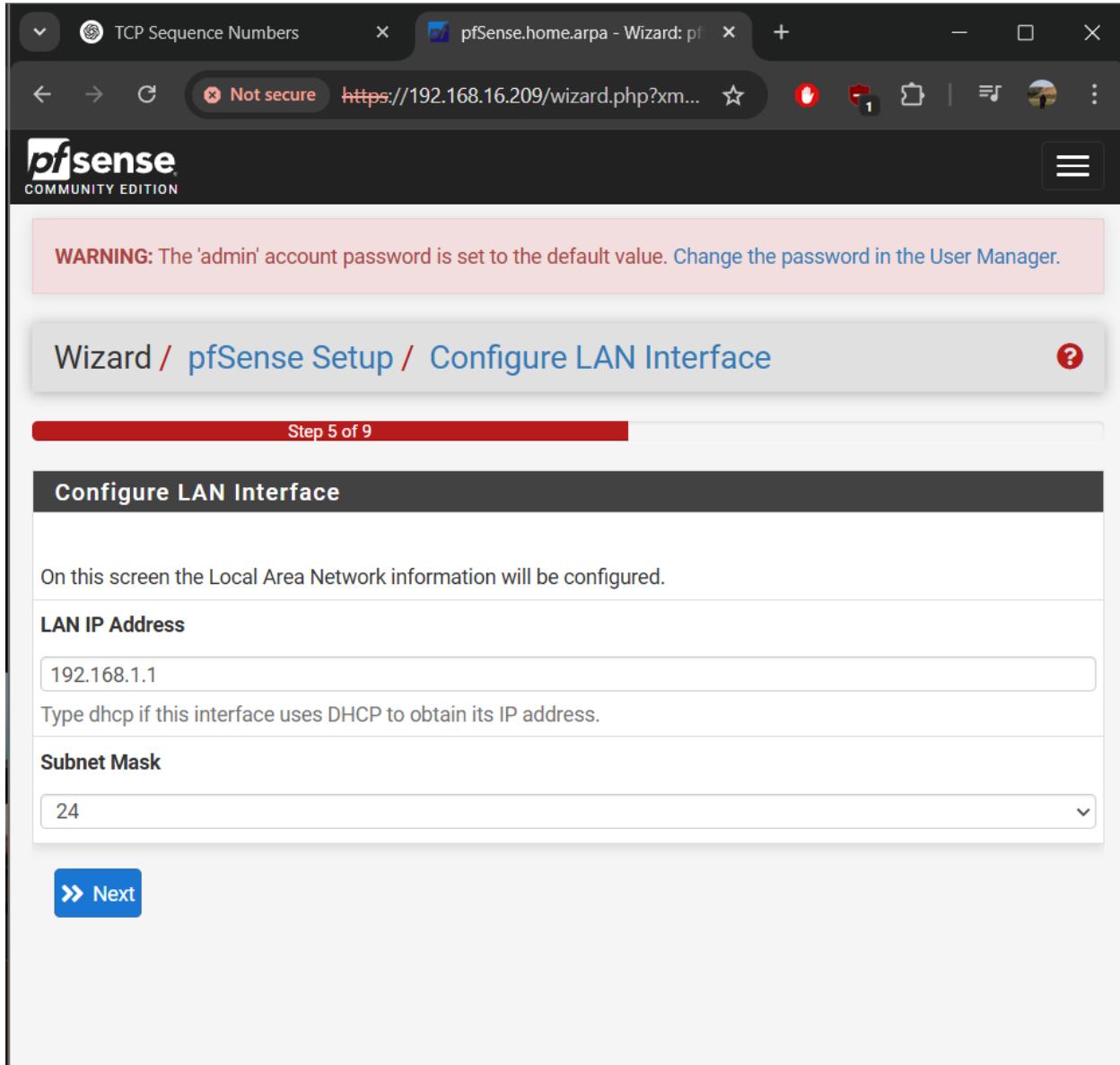
The WAN interface had the options to block RFC 1918 private networks and bogon networks enabled by default. Despite these settings being active, legitimate web pages loaded successfully, indicating that normal internet traffic was allowed while unwanted or suspicious packets were filtered automatically.

The screenshot shows a web-based configuration interface for pfSense. The title bar indicates it's a 'Wizard' step, specifically 'pf'. The address bar shows the URL <https://192.168.16.209/wizard.php?xm...>. The main content area has a header '32' and is titled 'PPTP Remote IP Address'. Below this is a large empty input field. The next section is titled 'PPTP Dial on demand' with a checkbox labeled 'Enable Dial-On-Demand mode'. A descriptive text follows: 'This option causes the interface to operate in dial-on-demand mode, allowing a virtual full time connection. The interface is configured, but the actual connection of the link is delayed until qualifying outgoing traffic is detected.' The next section is titled 'PPTP Idle timeout' with a large empty input field. A descriptive text follows: 'If no qualifying outgoing packets are transmitted for the specified number of seconds, the connection is brought down. An idle timeout of zero disables this feature.' The final section is titled 'RFC1918 Networks' with a header 'Block RFC1918 Private Networks'. It contains a checkbox 'Block private networks from entering via WAN' and a descriptive text: 'When set, this option blocks traffic from IP addresses that are reserved for private networks as per RFC 1918 (10/8, 172.16/12, 192.168/16) as well as loopback addresses (127/8). This option should generally be left turned on, unless the WAN network lies in such a private address space, too.' The next section is titled 'Block bogon networks' with a header 'Block bogon networks'. It contains a checkbox 'Block non-Internet routed networks from entering via WAN' and a descriptive text: 'When set, this option blocks traffic from IP addresses that are reserved (but not RFC 1918) or not yet assigned by IANA. Bogons are prefixes that should never appear in the Internet routing table, and obviously should not appear as the source address in any packets received.' At the bottom left is a blue button labeled '» Next'.

### Step 7: Disabling WAN Firewall Restrictions to Enable Web Access

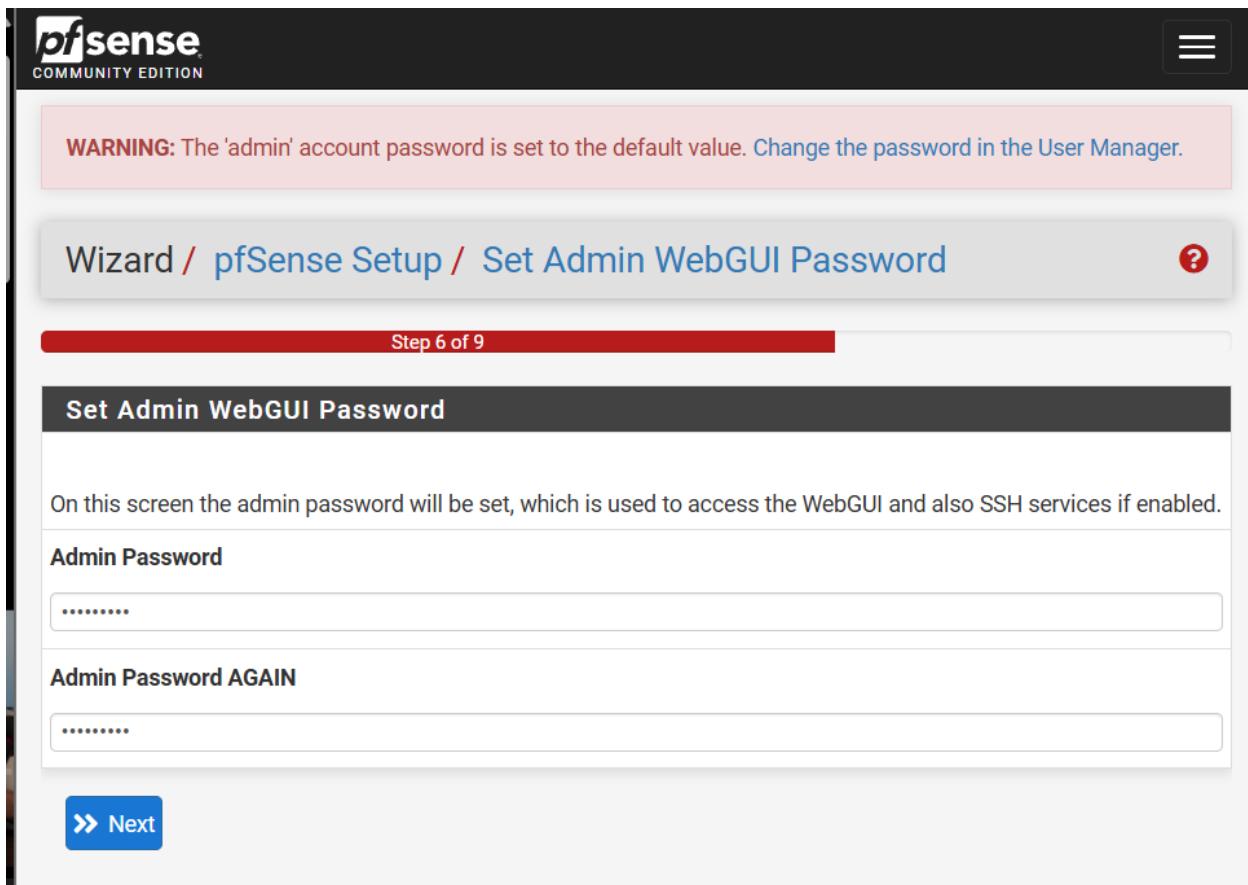
I unticked the options to block RFC 1918 private networks and bogon networks on the WAN interface. This change allows traffic from private IP ranges and reserved IP blocks, which is necessary when the WAN itself uses such private addressing. Disabling these blocks enabled successful web page access through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

The screenshot shows the final step of the pfSense Setup Wizard. The title bar reads "Wizard / pfSense Setup / Wizard completed." A green progress bar at the top indicates "Step 9 of 9". Below the bar, a dark banner displays the text "Wizard completed.". The main content area starts with "Congratulations! pfSense is now configured." followed by a recommendation to check for software updates. It also features a "Check for updates" button and a section titled "Remember, we're here to help." with a link to support services. Further down, there's a "User survey" section encouraging users to answer a short survey, with a link to the "Anonymous User Survey". The final section, "Useful resources.", lists four items: learning about Netgate's product line, visiting the store, joining the community forum, and subscribing to the newsletter. A blue "Finish" button is located at the bottom left.

Wizard / pfSense Setup / Wizard completed.

Step 9 of 9

Wizard completed.

Congratulations! pfSense is now configured.

We recommend that you check to see if there are any software updates available. Keeping your software up to date is one of the most important things you can do to maintain the security of your network.

[Check for updates](#)

Remember, we're here to help.

[Click here](#) to learn about Netgate 24/7/365 support services.

User survey

Please help all the people involved in improving and expanding pfSense software by taking a moment to answer this short survey (all answers are anonymous)

[Anonymous User Survey](#)

Useful resources.

- Learn more about Netgate's product line, services, and pfSense software from [our website](#)
- To learn about Netgate appliances and other offers, [visit our store](#)
- Become part of the pfSense community. Visit [our forum](#)
- Subscribe to our [newsletter](#) for ongoing product information, software announcements and special offers.

[Finish](#)

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

Enter an option: 8

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

```

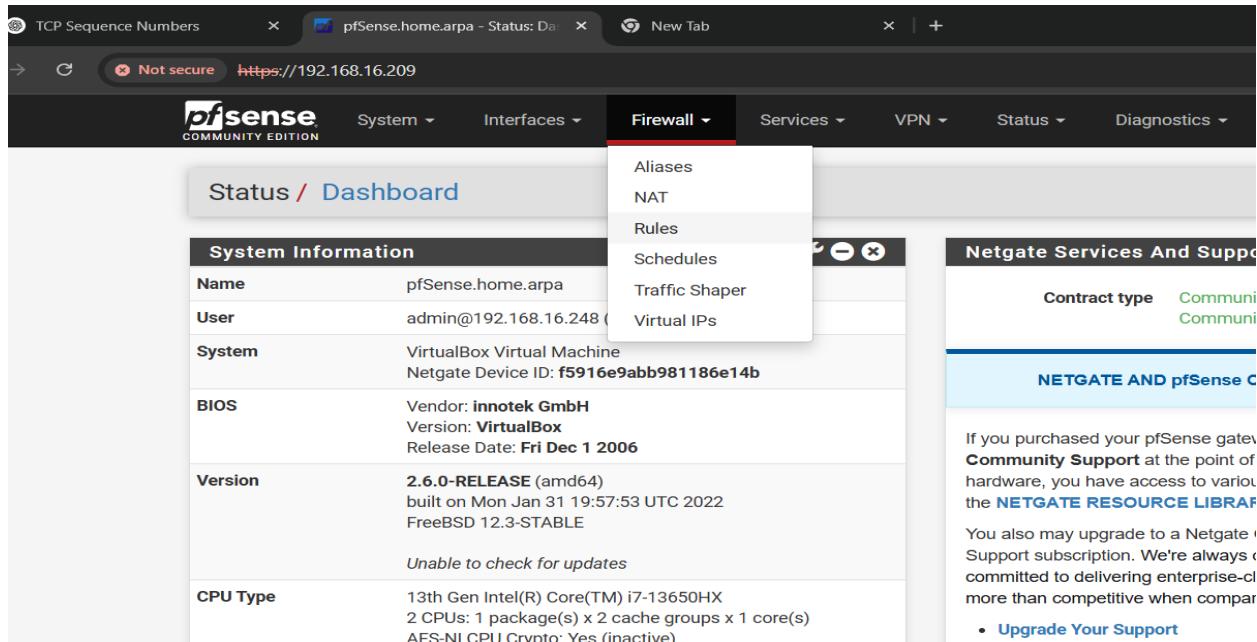
## Step 11: Temporarily Disable the pfSense Firewall via Console

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.

The screenshot shows the pfSense Status Dashboard. On the left, there's a sidebar with links like System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. The main area has tabs for Status and Dashboard. Under Status, there's a "System Information" table with details such as Name (pfSense.home.arp), User (admin@192.168.16.248 (Local Database)), System (VirtualBox Virtual Machine, Netgate Device ID: f5916e9abb981186e14b), BIOS (Vendor: innoteK GmbH, Version: VirtualBox, Release Date: Fri Dec 1 2006), Version (2.6.0-RELEASE (arm64), built on Mon Jan 31 19:57:53 UTC 2022, FreeBSD 12.3-STABLE), CPU Type (13th Gen Intel(R) Core(TM) i7-13650HX, 2 CPUs: 1 package(s) x 2 cache groups x 1 core(s), AES-NI CPU Crypto: Yes (inactive), QAT Crypto: No), and Hardware crypto (Kernel PTI: Disabled, MDS Mitigation: Inactive). To the right, there's a "Netgate Services And Support" section with a "Community Support" contract type, a "NETGATE AND pfSense COMMUNITY SUPPORT RESOURCES" section with links to upgrade support, community support resources, and professional services, and a note about purchasing TAC support. At the bottom, there's an "Interfaces" tab.

## Step 12: Accessing the pfSense Dashboard

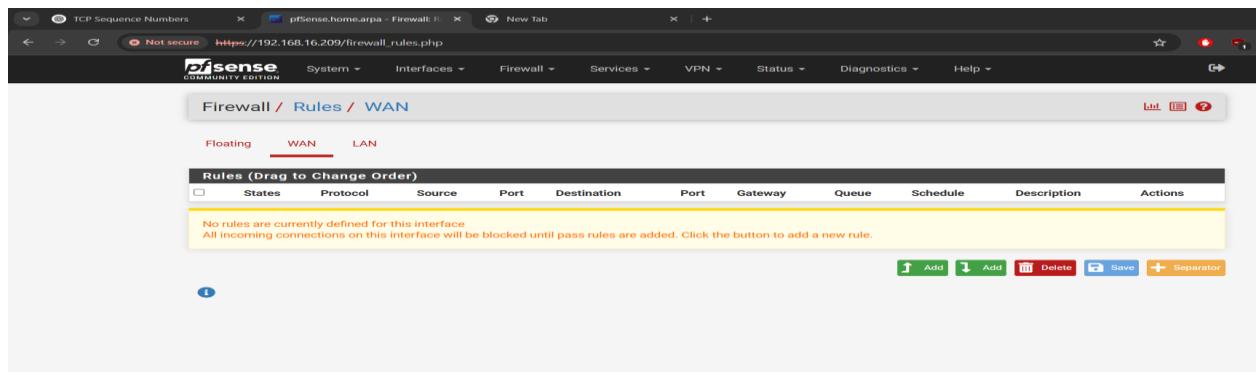
After disabling the firewall temporarily, I opened a web browser and navigated to the pfSense WAN IP address (e.g., <https://192.168.16.209>). The pfSense login page appeared. I logged in using the default credentials and was directed to the **Status Dashboard**, confirming successful access to the pfSense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



The screenshot shows the pfSense Status Dashboard. The Firewall menu is open, displaying options: Aliases, NAT, Rules, Schedules, Traffic Shaper, and Virtual IPs. The main dashboard area shows system information such as Name (pfSense.home.arpa), User (admin@192.168.16.248), System (VirtualBox Virtual Machine, Netgate Device ID: f5916e9abb981186e14b), BIOS (Vendor: innotek GmbH, Version: VirtualBox, Release Date: Fri Dec 1 2006), Version (2.6.0-RELEASE (amd64) built on Mon Jan 31 19:57:53 UTC 2022 FreeBSD 12.3-STABLE), and CPU Type (13th Gen Intel(R) Core(TM) i7-13650HX, 2 CPUs: 1 package(s) x 2 cache groups x 1 core(s), AFS-NI CPU Crypto: Yes (inactive)). On the right, there's a sidebar for Netgate Services And Support and a message about purchasing support.

## 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.



The screenshot shows the Firewall > Rules > WAN interface. The WAN tab is selected. The main area displays a table titled "Rules (Drag to Change Order)" with columns: States, Protocol, Source, Port, Destination, Port, Gateway, Queue, Schedule, Description, and Actions. A message at the top states: "No rules are currently defined for this interface. All incoming connections on this interface will be blocked until pass rules are added. Click the button to add a new rule." At the bottom, there are buttons for Add, Add, Delete, Save, and Separator.

## 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.

The screenshot shows a web-based configuration interface for a firewall rule. The URL is `/192.168.16.209/firewall_rules_edit.php?if=wan&after=-1`. The page is titled "Choose what to do with packets that match the criteria specified below". It includes a hint about the difference between block and reject actions. The configuration fields are as follows:

- Disabled:**  Disable this rule. Set this option to disable this rule without removing it from the list.
- Interface:** WAN. Choose the interface from which packets must come to match this rule.
- Address Family:** IPv4. Select the Internet Protocol version this rule applies to.
- Protocol:** TCP. Choose which IP protocol this rule should match.

**Source:**

- Source:** Network 192.168.16.0/24. Includes an "Invert match" checkbox and a "Display Advanced" button.

**Destination:**

- Destination:** Single host or alias 192.168.16.209. Includes an "Invert match" checkbox.
- Destination Port Range:** (other). From Custom To Custom. Specify the destination port or port range for this rule. The "To" field may be left empty if only filtering a single port.

**Extra Options:**

- Log:**  Log packets that are handled by this rule. Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the Status: System Logs: Settings page).
- Description:** A description may be entered here for administrative reference. A maximum of 52 characters will be used in the ruleset and displayed in the firewall log.
- Advanced Options:** Includes a "Display Advanced" button.

**Save** button at the bottom.

## 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.

The screenshot shows the pfSense Firewall Rules interface. The URL is https://192.168.16.209/firewall\_rules.php. The navigation bar includes System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. The main menu shows Firewall / Rules / WAN. Below this is a sub-menu: Floating, WAN (selected), and LAN. A table titled "Rules (Drag to Change Order)" lists one rule: "2 / 20 KIB" (Protocol: IPv4 TCP, Source: 192.168.16.0/24, Destination: 192.168.16.209). Action buttons include Add, Delete, Save, and Separator.

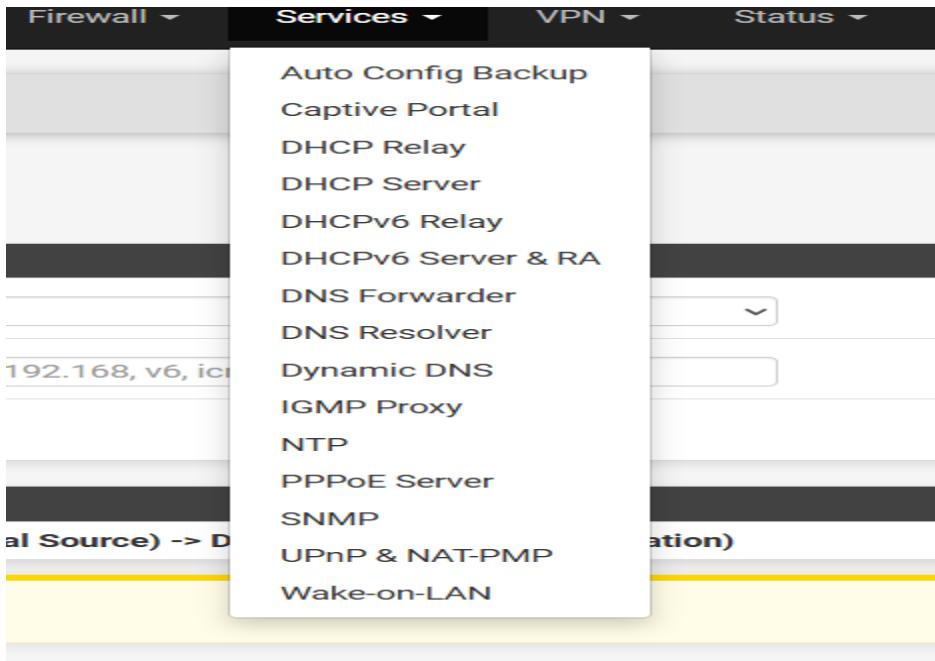
## Step 16: Verifying Firewall Rules on WAN and LAN Interfaces

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.

The screenshot shows the pfSense WAN interface configuration page for interface em0. The title is "Interfaces / WAN (em0)". The "General Configuration" section includes fields for Enable (checked), Description (WAN), IPv4 Configuration Type (DHCP), IPv6 Configuration Type (DHCP6), MAC Address (XXXX:XXXX:XXXX:XX), MTU (blank), MSS (blank), and Speed and Duplex (Default). The "DHCP Client Configuration" section has options for Advanced Configuration (unchecked) and Configuration Override (unchecked).

## 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

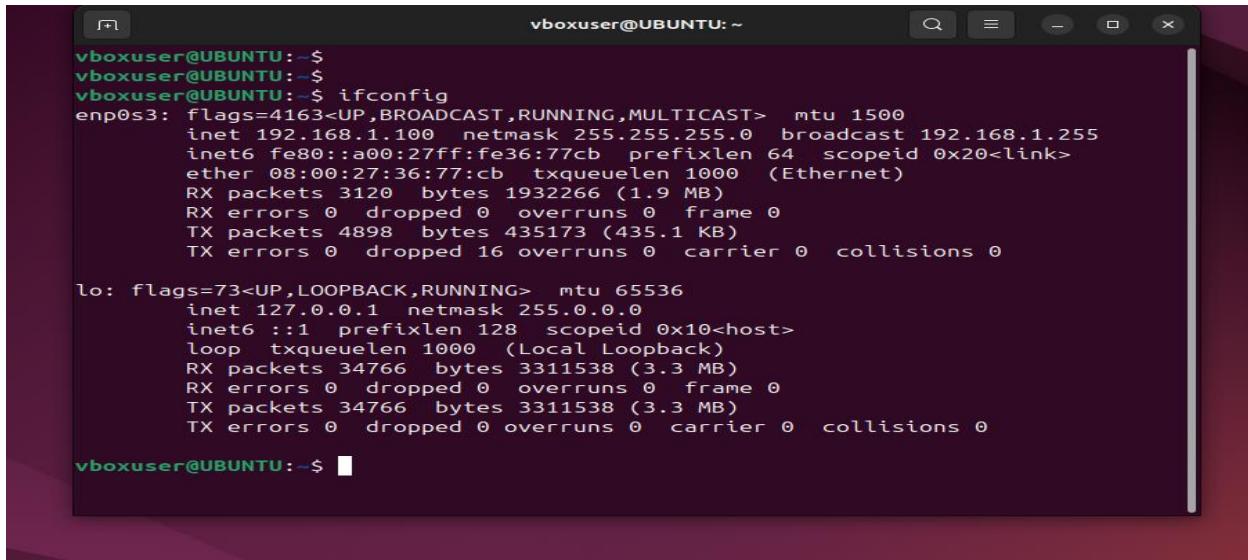
By enabling DHCP on the LAN interface, devices like the Ubuntu machine connected to the internal network will automatically receive an IP address. This allows the Ubuntu system to communicate on the LAN without manual IP configuration, simplifying network setup and management.

ip.php

<small>This option may be useful when a client can dual boot using different client identifiers but the same hardware (MAC) address. Note that the resulting server behavior violates the official DHCP specification.</small>				
<b>Subnet</b>	192.168.1.0			
<b>Subnet mask</b>	255.255.255.0			
<b>Available range</b>	192.168.1.1 - 192.168.1.254			
<b>Range</b>	192.168.1.10 From: _____ To: _____			
<b>Additional Pools</b>				
<input type="button" value="Add pool"/> <span style="color: green;">+ Add pool</span>				
<small>If additional pools of addresses are needed inside of this subnet outside the above Range, they may be specified here.</small>				
<b>Pool Start</b>	192.168.1.245			
<b>Pool End</b>	192.168.1.254			
<b>Description</b>				
<b>Servers</b>				
<b>WINS servers</b>	WINS Server 1 WINS Server 2			
<b>DNS servers</b>	DNS Server 1 DNS Server 2 DNS Server 3 DNS Server 4 DNS Server 5			
<small>Leave blank to use the system default DNS servers this Interface IP if DNS Forwarder or Resolver is enabled, otherwise the servers configured on the System / General Setup page.</small>				
<b>CMAPI</b>				
<b>CMAPI Port</b>	CMAPI Port <small>Set the port that CMAPI will listen on. The default port is 7911, leave blank to disable. Only the first CMAPI configuration is used.</small>			
<b>CMAPI Key</b>	CMAPI Key <small>Enter a key matching the selected algorithm. To secure connections to the CMAPI endpoint.</small>			
<b>Key Algorithm</b>	HMAC-SHA256 (current binds default) <small>Set the algorithm that CMAPI key will use.</small>			
<b>Other Options</b>				
<b>Gateway</b>	<small>The default is to use the IP on this interface of the firewall as the gateway. Specify an alternate gateway here if this is not the correct gateway for the network. Type "none" for no gateway assignment.</small>			
<b>Domain name</b>	<small>The default is to use the domain name of this system as the default domain name provided by DHCP. An alternate domain name may be specified here.</small>			
<b>Domain search list</b>	<small>The DHCP server can optionally provide a domain search list. Use the semicolon character as separator.</small>			
<b>Default lease time</b>	<small>This is used for clients that do not ask for a specific expiration time. The default is 7200 seconds.</small>			
<b>Maximum lease time</b>	<small>This is the maximum lease time for clients that ask for a specific expiration time. The default is 60000 seconds.</small>			
<b>Relayover peer IP</b>	<small>Leave blank to disable. Enter the interface IP address of the other machine. Machines must be using CARB Interface's algorithm whether the DHCP process is Primary or Secondary. Ensure one machine's adminstrator + 20 (and the other - 20).</small>			
<b>Static ARP</b>	<input type="checkbox"/> Create Static ARP entries <small>This option persists even if DHCP server is disabled. Only the machines listed below will be able to communicate with the firewall on this interface.</small>			
<b>Time format change</b>	<input type="checkbox"/> Change DHCP display lease time from UTC to local time <small>By default DHCP leases are displayed in UTC time. By checking this box DHCP lease time will be displayed in local time and set to the time zone selected. This will be used for all DHCP interfaces lease time.</small>			
<b>Statistics graphs</b>	<input type="checkbox"/> Create RRD statistics graphs <small>Enable this to add DHCP leases statistics to the RRD graphs. Disabled by default.</small>			
<b>Ping check</b>	<input type="checkbox"/> Disable ping check <small>When enabled dhclient sends a ping to the address being assigned, and if no response has been heard, it assigns the address. Enabled by default.</small>			
<b>Dynamic DNS</b>	<input checked="" type="checkbox"/> Dynamic DNS <input checked="" type="checkbox"/> Display Addresses			
<b>MAC address control</b>	<input checked="" type="checkbox"/> MAC Address Control <input checked="" type="checkbox"/> Display Addresses			
<b>NTP</b>	<input checked="" type="checkbox"/> NTP <input checked="" type="checkbox"/> Display Addresses			
<b>TFTP</b>	<input checked="" type="checkbox"/> TFTP <input checked="" type="checkbox"/> Display Addresses			
<b>LDAP</b>	<input checked="" type="checkbox"/> LDAP <input checked="" type="checkbox"/> Display Addresses			
<b>Network Booting</b>	<input checked="" type="checkbox"/> Network Booting <input checked="" type="checkbox"/> Display Addresses			
<b>Additional BOOTP/DHCP Options</b>	<input checked="" type="checkbox"/> Additional Options <input checked="" type="checkbox"/> Display Addresses			
<input type="button" value="Save"/> <span style="color: green;">+ Add</span>				
<b>DHCP Static Mappings for this Interface</b>				
<b>Static ARP</b>	<b>MAC address</b>	<b>IP address</b>	<b>Hostname</b>	<b>Description</b>

## Step 19: Configure LAN Subnet and DNS Settings

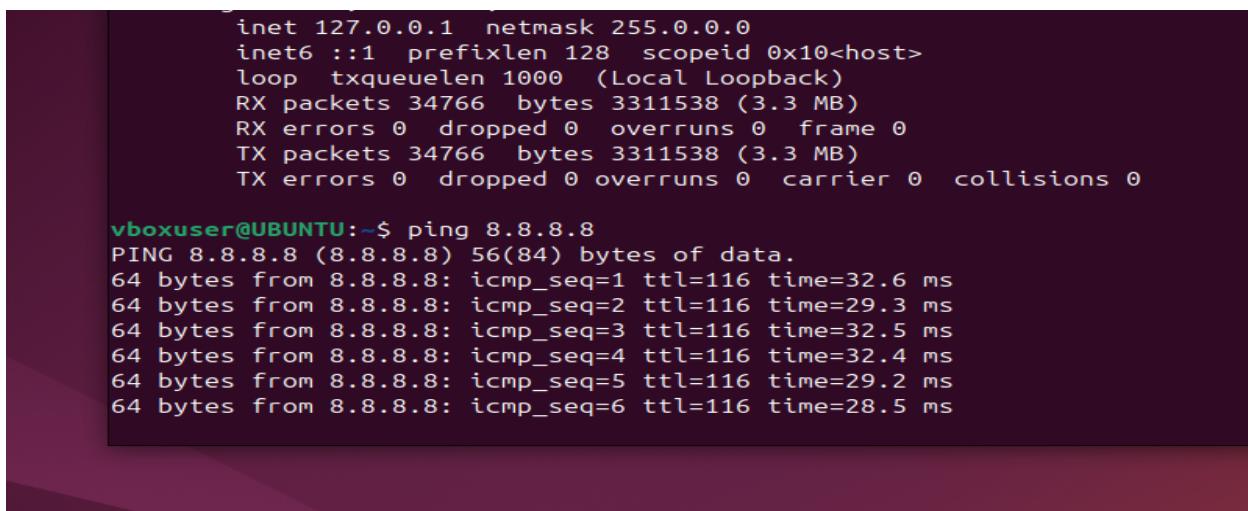
We set the LAN subnet to **192.168.1.0** with a subnet mask of **255.255.255.0**, providing an IP range from **192.168.1.1** to **192.168.1.254** for devices on the network. For DNS, we use the home router's DNS server to resolve domain names, ensuring proper internet connectivity and name resolution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



```
vboxuser@UBUNTU:~$  
vboxuser@UBUNTU:~$  
vboxuser@UBUNTU:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
        inet 192.168.1.100 netmask 255.255.255.0 broadcast 192.168.1.255  
              inet6 fe80::a00:27ff:fe36:77cb prefixlen 64 scopeid 0x20<link>  
                ether 08:00:27:36:77:cb txqueuelen 1000 (Ethernet)  
                  RX packets 3120 bytes 1932266 (1.9 MB)  
                  RX errors 0 dropped 0 overruns 0 frame 0  
                  TX packets 4898 bytes 435173 (435.1 KB)  
                  TX errors 0 dropped 16 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
        inet 127.0.0.1 netmask 255.0.0.0  
              inet6 ::1 prefixlen 128 scopeid 0x10<host>  
                loop txqueuelen 1000 (Local Loopback)  
                  RX packets 34766 bytes 3311538 (3.3 MB)  
                  RX errors 0 dropped 0 overruns 0 frame 0  
                  TX packets 34766 bytes 3311538 (3.3 MB)  
                  TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
vboxuser@UBUNTU:~$
```

## Step 20: Verify Ubuntu Network Configuration

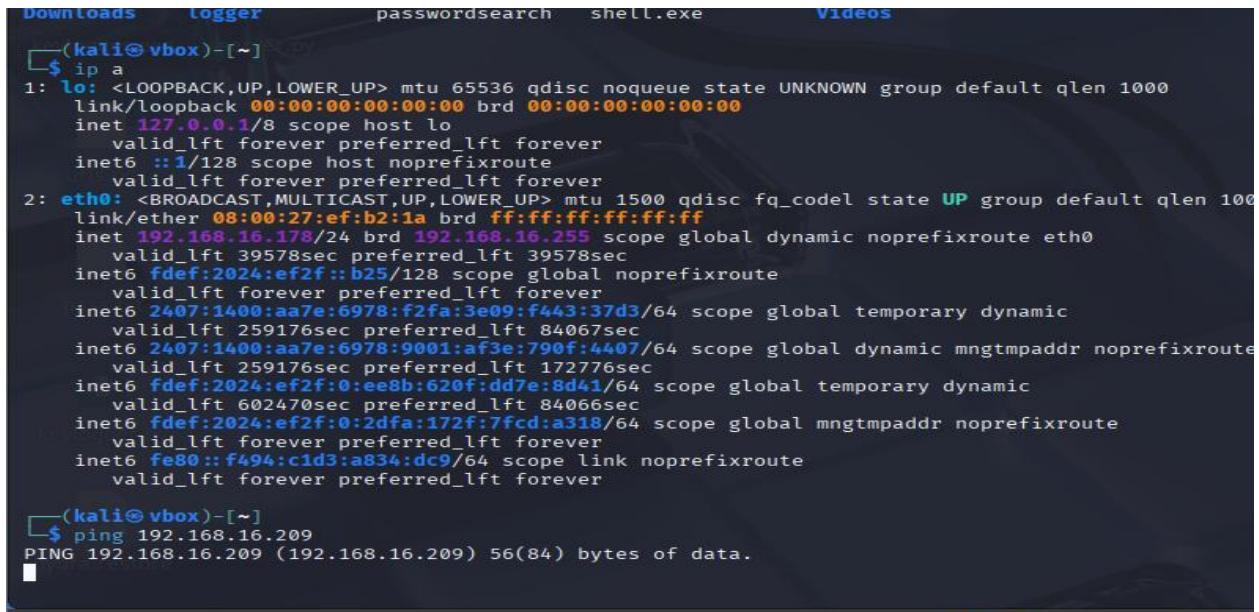
On the Ubuntu VM, the network interface **enp0s3** is configured on the internal network with IP **192.168.1.100** (via DHCP), subnet mask **255.255.255.0**, and is successfully transmitting and receiving packets. This confirms the DHCP service from pfSense is working correctly, assigning IPs in the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



```
vboxuser@UBUNTU:~$  
vboxuser@UBUNTU:~$ ping 8.8.8.8  
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.  
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=32.6 ms  
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=29.3 ms  
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=32.5 ms  
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=32.4 ms  
64 bytes from 8.8.8.8: icmp_seq=5 ttl=116 time=29.2 ms  
64 bytes from 8.8.8.8: icmp_seq=6 ttl=116 time=28.5 ms
```

## 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.



(kali㉿vbox) ~ \$ ip a  
1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
 inet 127.0.0.1/8 scope host lo  
 valid\_lft forever preferred\_lft forever  
 inet6 ::1/128 scope host noprefixroute  
 valid\_lft forever preferred\_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UP group default qlen 1000  
 link/ether 08:00:27:ef:b2:1a brd ff:ff:ff:ff:ff:ff  
 inet 192.168.16.178/24 brd 192.168.16.255 scope global dynamic noprefixroute eth0  
 valid\_lft 39578sec preferred\_lft 39578sec  
 inet6 fdef:2024:ef2f::b25/128 scope global noprefixroute  
 valid\_lft forever preferred\_lft forever  
 inet6 2407:1400:aa7e:6978:f2fa:3e09:f443:37d3/64 scope global temporary dynamic  
 valid\_lft 259176sec preferred\_lft 84067sec  
 inet6 2407:1400:aa7e:6978:9001:af3e:790f:4407/64 scope global dynamic mngtmpaddr noprefixroute  
 valid\_lft 259176sec preferred\_lft 172776sec  
 inet6 fdef:2024:ef2f:0:ee8b:620f:dd7e:8d41/64 scope global temporary dynamic  
 valid\_lft 602470sec preferred\_lft 84066sec  
 inet6 fdef:2024:ef2f:0:2dfa:172f:7fcda:318/64 scope global mngtmpaddr noprefixroute  
 valid\_lft forever preferred\_lft forever  
 inet6 fe80::f494:c1d3:a834:dc9/64 scope link noprefixroute  
 valid\_lft forever preferred\_lft forever  
  
(kali㉿vbox) ~ \$ ping 192.168.16.209  
PING 192.168.16.209 (192.168.16.209) 56(84) bytes of data.

## 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.

pfSense.home.arpa - Firewall: Rules

Not secure https://192.168.16.209/firewall\_rules\_edit.php?if=wan&after...

Choose what to do with packets that match the criteria specified below.  
Hint: the difference between block and reject is that with reject, a packet (TCP RST or ICMP port unreachable for UDP) is returned to the sender, whereas with block the packet is dropped silently. In either case, the original packet is discarded.

**Disabled**  Disable this rule  
Set this option to disable this rule without removing it from the list.

**Interface** WAN  
Choose the interface from which packets must come to match this rule.

**Address Family** IPv4  
Select the Internet Protocol version this rule applies to.

**Protocol** ICMP  
Choose which IP protocol this rule should match.

**ICMP Subtypes** any  
Alternate Host  
Datagram conversion error  
Echo reply  
For ICMP rules on IPv4, one or more of these ICMP subtypes may be specified.

**Source**

**Source**  Invert match Single host or alias 192.168.16.178 /

**Destination**

**Destination**  Invert match Single host or alias 192.168.16.209 /

**Extra Options**

**Log**  Log packets that are handled by this rule  
Hint: the firewall has limited local log space. Don't turn on logging for everything. If doing a lot of logging, consider using a remote syslog server (see the [Status: System Logs: Settings](#) page).

**Description**   
A description may be entered here for administrative reference. A maximum of 52 characters will be used in the ruleset and displayed in the firewall log.

**Advanced Options**

The screenshot shows the pfSense Firewall Rules configuration page. A new rule is being created for ICMP traffic. The 'Protocol' is set to ICMP. The 'Source' is set to 'Single host or alias' with the IP 192.168.16.178. The 'Destination' is set to 'Single host or alias' with the IP 192.168.16.209. The 'Log' checkbox is checked. The 'Description' field is empty. The 'Advanced Options' button is visible. The 'Save' button is at the bottom.

### 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.

The screenshot shows the pfSense web interface for managing firewall rules. The URL is https://192.168.16.209/firewall\_rules.php. The navigation bar includes links for System, Interfaces, Firewall, Services, VPN, Status, Diagnostics, and Help. The main title is 'Firewall / Rules / WAN'. Below the title, there are tabs for Floating, WAN (which is selected), and LAN. The main content area displays a table titled 'Rules (Drag to Change Order)'. The table has columns for序号 (Index), States, Protocol, Source, Port, Destination, Port, Gateway, Queue, Schedule, Description, and Actions. There are two entries:

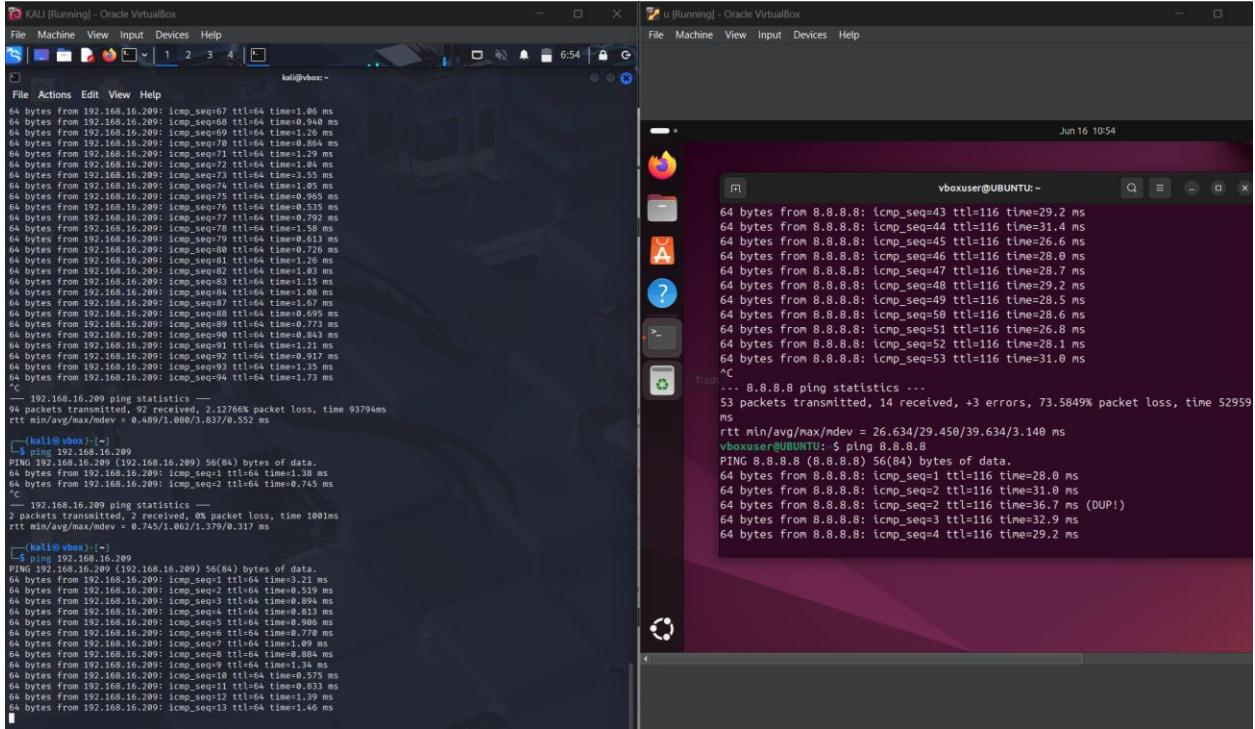
序号	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
1	<input checked="" type="checkbox"/>	IPv4 ICMP	192.168.16.178	*	192.168.16.209	*	*	none		Add, Edit, Delete, Save, Separator	
2	<input checked="" type="checkbox"/>	IPv4 TCP	192.168.16.0/24	*	192.168.16.209	*	*	none		Add, Edit, Delete, Save, Separator	

At the bottom of the table are buttons for Add, Edit, Delete, Save, and Separator.

## 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.

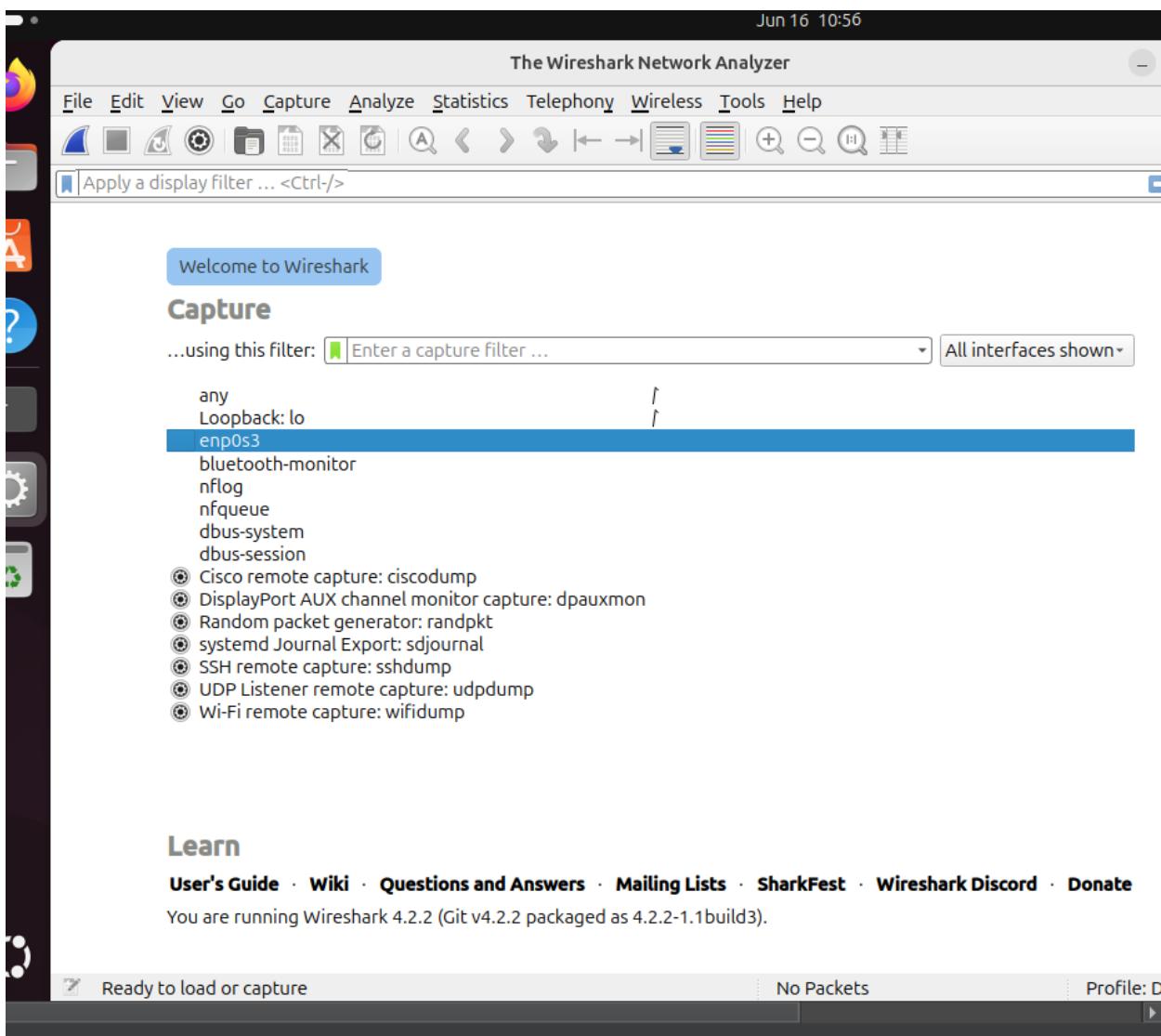


## Step 25: Enable Communication Between Kali, pfSense Firewall, and Ubuntu

- 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

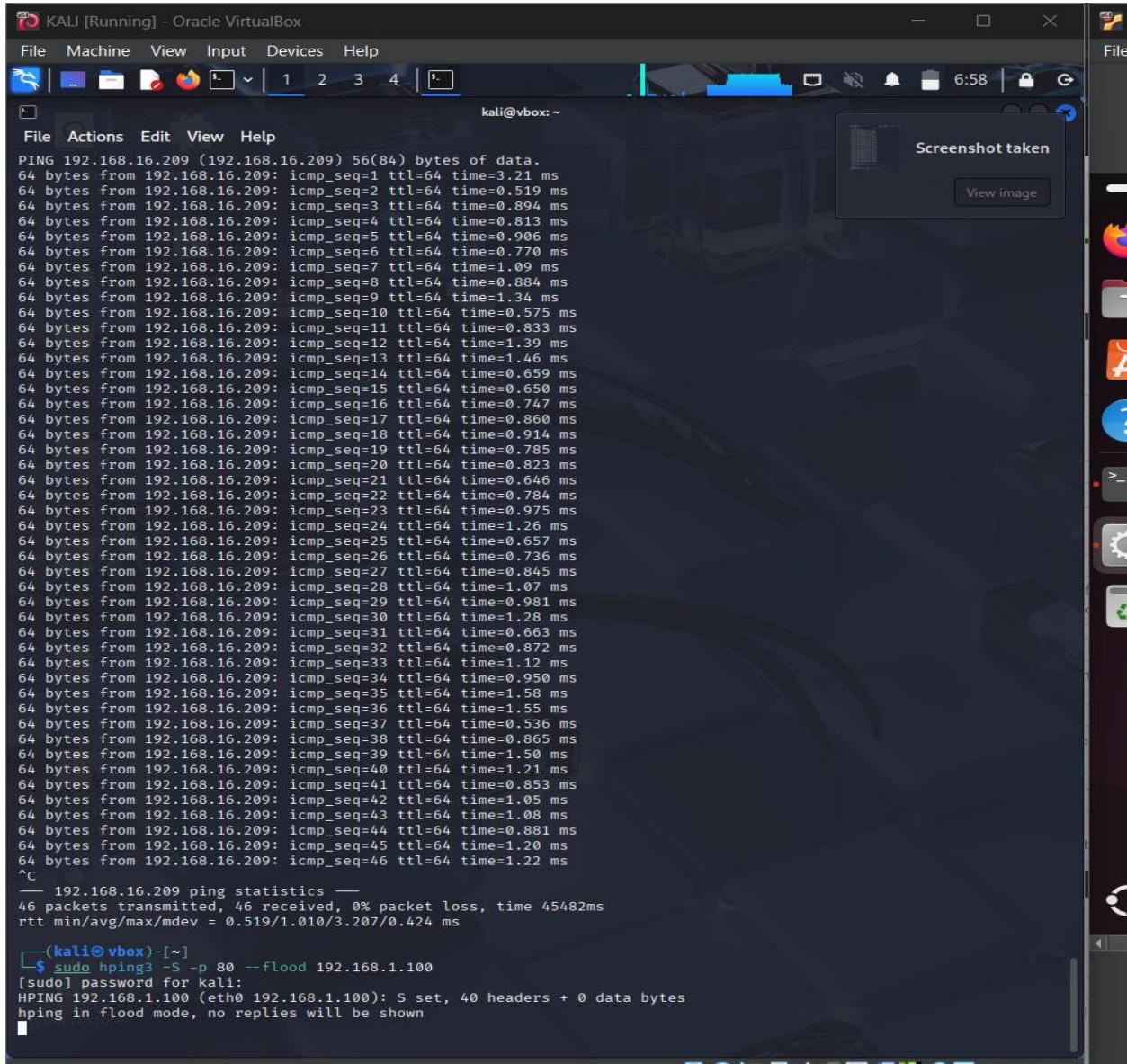
In this step, we demonstrate a simple Denial of Service (DoS) attack within the virtual network setup. We have already deployed a firewall rule allowing ICMP traffic from the Kali attacker to the Ubuntu victim, enabling them to successfully ping each other. This connectivity ensures that Kali can now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



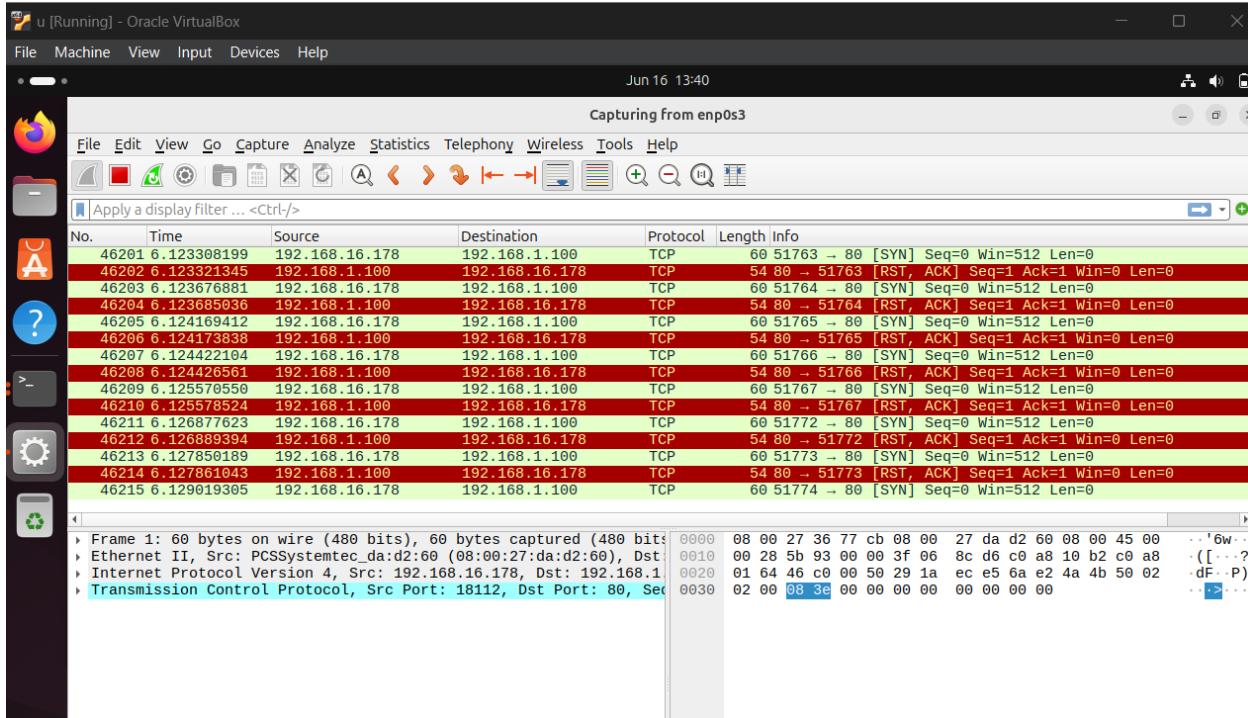
The screenshot shows a Kali Linux terminal window titled "KALI [Running] - Oracle VirtualBox". The terminal displays a ping test from the local host to IP 192.168.16.209, showing a sequence of ICMP echo requests (seq=1 to seq=46) with varying response times. A message "Screenshot taken" is visible in the top right corner. Below the ping output, the user runs the command \$ sudo hping3 -S -p 80 --flood 192.168.1.100, which triggers a SYN flood attack on port 80 of the target host 192.168.1.100.

```
PING 192.168.16.209 (192.168.16.209) 56(84) bytes of data.
64 bytes from 192.168.16.209: icmp_seq=1 ttl=64 time=3.21 ms
64 bytes from 192.168.16.209: icmp_seq=2 ttl=64 time=0.519 ms
64 bytes from 192.168.16.209: icmp_seq=3 ttl=64 time=0.894 ms
64 bytes from 192.168.16.209: icmp_seq=4 ttl=64 time=0.813 ms
64 bytes from 192.168.16.209: icmp_seq=5 ttl=64 time=0.906 ms
64 bytes from 192.168.16.209: icmp_seq=6 ttl=64 time=0.770 ms
64 bytes from 192.168.16.209: icmp_seq=7 ttl=64 time=1.09 ms
64 bytes from 192.168.16.209: icmp_seq=8 ttl=64 time=0.884 ms
64 bytes from 192.168.16.209: icmp_seq=9 ttl=64 time=1.34 ms
64 bytes from 192.168.16.209: icmp_seq=10 ttl=64 time=0.575 ms
64 bytes from 192.168.16.209: icmp_seq=11 ttl=64 time=0.833 ms
64 bytes from 192.168.16.209: icmp_seq=12 ttl=64 time=1.39 ms
64 bytes from 192.168.16.209: icmp_seq=13 ttl=64 time=1.46 ms
64 bytes from 192.168.16.209: icmp_seq=14 ttl=64 time=0.659 ms
64 bytes from 192.168.16.209: icmp_seq=15 ttl=64 time=0.650 ms
64 bytes from 192.168.16.209: icmp_seq=16 ttl=64 time=0.747 ms
64 bytes from 192.168.16.209: icmp_seq=17 ttl=64 time=0.860 ms
64 bytes from 192.168.16.209: icmp_seq=18 ttl=64 time=0.914 ms
64 bytes from 192.168.16.209: icmp_seq=19 ttl=64 time=0.785 ms
64 bytes from 192.168.16.209: icmp_seq=20 ttl=64 time=0.823 ms
64 bytes from 192.168.16.209: icmp_seq=21 ttl=64 time=0.646 ms
64 bytes from 192.168.16.209: icmp_seq=22 ttl=64 time=0.784 ms
64 bytes from 192.168.16.209: icmp_seq=23 ttl=64 time=0.975 ms
64 bytes from 192.168.16.209: icmp_seq=24 ttl=64 time=1.26 ms
64 bytes from 192.168.16.209: icmp_seq=25 ttl=64 time=0.657 ms
64 bytes from 192.168.16.209: icmp_seq=26 ttl=64 time=0.736 ms
64 bytes from 192.168.16.209: icmp_seq=27 ttl=64 time=0.845 ms
64 bytes from 192.168.16.209: icmp_seq=28 ttl=64 time=1.07 ms
64 bytes from 192.168.16.209: icmp_seq=29 ttl=64 time=0.981 ms
64 bytes from 192.168.16.209: icmp_seq=30 ttl=64 time=1.28 ms
64 bytes from 192.168.16.209: icmp_seq=31 ttl=64 time=0.663 ms
64 bytes from 192.168.16.209: icmp_seq=32 ttl=64 time=0.872 ms
64 bytes from 192.168.16.209: icmp_seq=33 ttl=64 time=1.12 ms
64 bytes from 192.168.16.209: icmp_seq=34 ttl=64 time=0.950 ms
64 bytes from 192.168.16.209: icmp_seq=35 ttl=64 time=1.58 ms
64 bytes from 192.168.16.209: icmp_seq=36 ttl=64 time=1.55 ms
64 bytes from 192.168.16.209: icmp_seq=37 ttl=64 time=0.536 ms
64 bytes from 192.168.16.209: icmp_seq=38 ttl=64 time=0.865 ms
64 bytes from 192.168.16.209: icmp_seq=39 ttl=64 time=1.50 ms
64 bytes from 192.168.16.209: icmp_seq=40 ttl=64 time=1.21 ms
64 bytes from 192.168.16.209: icmp_seq=41 ttl=64 time=0.853 ms
64 bytes from 192.168.16.209: icmp_seq=42 ttl=64 time=1.05 ms
64 bytes from 192.168.16.209: icmp_seq=43 ttl=64 time=1.08 ms
64 bytes from 192.168.16.209: icmp_seq=44 ttl=64 time=0.881 ms
64 bytes from 192.168.16.209: icmp_seq=45 ttl=64 time=1.20 ms
64 bytes from 192.168.16.209: icmp_seq=46 ttl=64 time=1.22 ms
^C
-- 192.168.16.209 ping statistics --
46 packets transmitted, 46 received, 0% packet loss, time 45482ms
rtt min/avg/max/mdev = 0.519/1.010/3.207/0.424 ms

(kali㉿vbox)-[~]
$ sudo hping3 -S -p 80 --flood 192.168.1.100
[sudo] password for kali:
HPING 192.168.1.100 (eth0 192.168.1.100): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

## 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.

The screenshot shows the pfSense Firewall Rules Edit interface. A new rule is being created with the following settings:

- Action:** Block
- Disabled:** Unchecked
- Interface:** WAN
- Address Family:** IPv4
- Protocol:** TCP
- Source:** Single host or alias: 192.168.16.178
- Destination:** Single host or alias: 192.168.1.100
- Extra Options:**
  - Log:** Checked
  - Description:** (empty)

## 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.

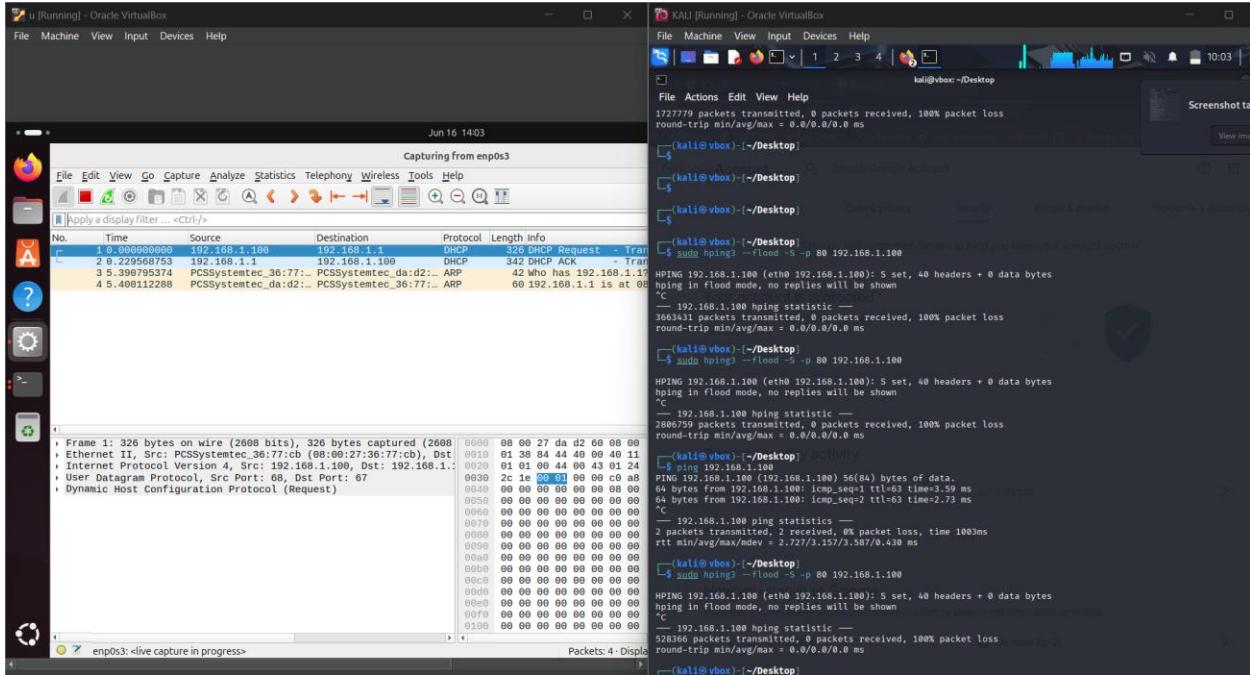
The screenshot shows the pfSense Firewall Rules WAN interface. At the top, a message box states: "The changes have been applied successfully. The firewall rules are now reloading in the background. Monitor the filter reload progress." Below this, there are tabs for Floating, WAN, and LAN, with WAN selected. The main table lists firewall rules:

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description	Actions
<input type="checkbox"/>	0 /11.95 MiB	IPv4 TCP	192.168.16.178	*	192.168.1.100	*	*	*	none		
<input type="checkbox"/>	0 /3 KiB	IPv4 ICMP	any	192.168.16.178	*	192.168.1.100	*	*	none	Allow Kali ICMP to Ubuntu	
<input type="checkbox"/>	0 /43.78 MiB	IPv4 TCP	192.168.16.178	*	192.168.1.100	*	*	*	none		
<input type="checkbox"/>	0 /1008 B	IPv4 ICMP	any	192.168.16.178	*	192.168.16.209	*	*	none		
<input type="checkbox"/>	2 /11.56 MiB	IPv4 TCP	192.168.16.0/24	*	192.168.16.209	*	*	*	none		

At the bottom, there are buttons for Add, Save, and Separator.

## 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)
<b>Setup</b>	-Realistic network segmentation (WAN/LAN). -Isolated VMs (Kali, Ubuntu, pfSense) for safe testing.	-Initial DHCP misconfiguration on LAN. <b>Fix:</b> Enabled DHCP server on pfSense (192.168.1.1/24).
<b>Attack Simulation</b>	-Successfully demonstrated DoS attacks using hping3 and Wireshark for analysis.	-SYN flood bypassed rules initially due to incorrect rule order.
<b>Firewall Rules</b>	-Granular control over traffic (ICMP/TCP blocking, IP-based filtering).	-Rule misplacement caused conflicts (e.g., allow rules overriding blocks).
<b>Connectivity</b>	-Ubuntu accessed the internet via pfSense NAT; Kali bridged to host network.	-Kali lost connectivity when host WiFi changed (required adapter reconfiguration).
<b>Scalability</b>	-Easy to expand (e.g., adding IDS like Suricata or VPN support).	-Manual traffic logging made forensic analysis tedious.
<b>Ease of Use</b>	-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.