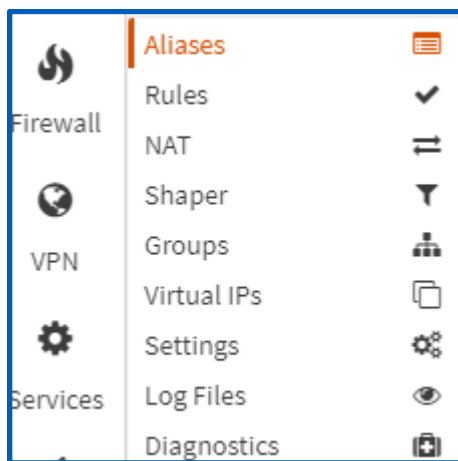


## OPNsense Firewall: Blocking IP's, Ports, and Protocols

In this project, I explored the OPNsense firewall, a tool crucial for network security. The focus was on understanding and implementing firewall rules and aliases to manage and secure networks effectively. The goal was to grasp practical insights into blocking specific IPs, protocols, and ports while optimizing organizational efficiency through alias usage.

Beginning with the firewall interface overview, I navigated to the firewall tab on the left-hand side of OPNsense. This section encompasses various options, with a focus on Aliases, Rules, Settings, and Log Files for this module.



Before delving into specific settings, I went into Settings > Advanced to disable IPv6, ensuring a streamlined configuration. The process involved turning off Allow IPv6 and adjusting Firewall Optimization under Miscellaneous to the normal setting. These configurations were saved to implement the changes.

OPNsense

Lobby  
Reporting  
System  
Interfaces  
**Firewall** 1  
Aliases  
Categories  
Groups  
NAT  
Rules  
Shaper  
**Settings** 2  
**Advanced** 3  
Normalization  
Schedules  
Log Files  
Diagnostics  
VPN

### Firewall: Settings: Advanced

IPv6 Options

Allow IPv6 ☐ Allow IPv6 4

Network Address Translation

Reflection for port forwards ☐

Reflection for 1:1 ☐

Automatic outbound NAT for Reflection ☐

Bogon Networks

Update Frequency Monthly

Gateway Monitoring

Skip rules ☐ Skip rules when gateway is down

**Miscellaneous**

Firewall Optimization normal

Bind states to interface ☐

Disable Firewall ☐ Disable all packet filtering.

Firewall Adaptive Timeouts start end

Dynamic state reset ☐ Reset all states when a dynamic IP address changes.

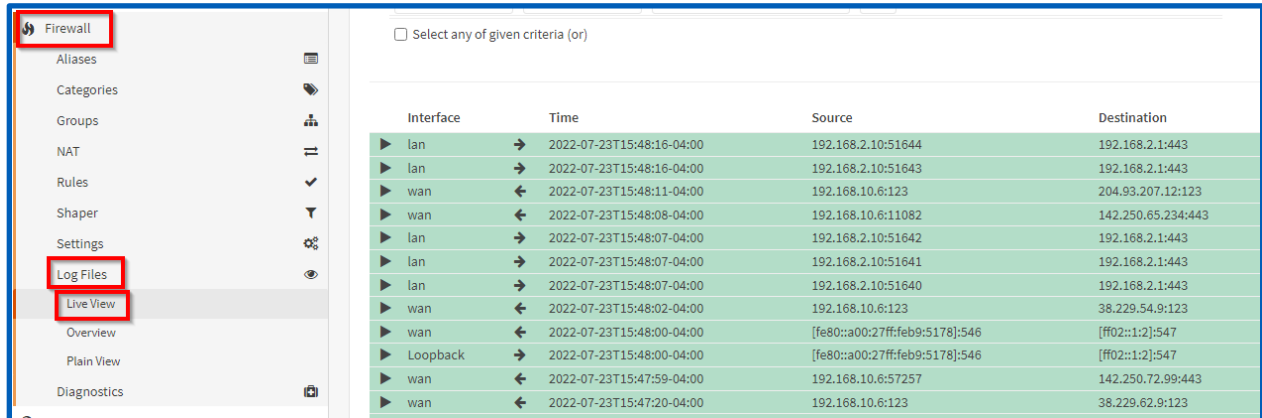
Anti DDOS

Enable syncookies never (default)

**Save**

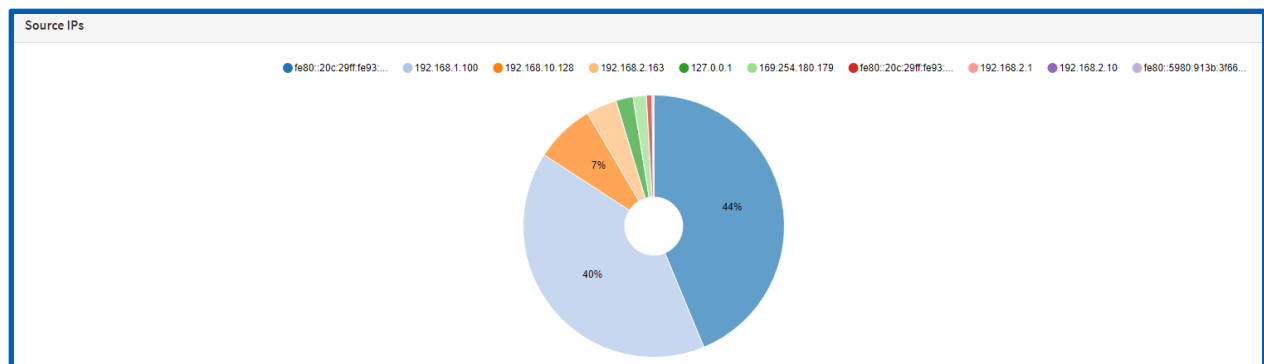
In the second part, I focused on configuring aliases, an essential aspect of firewall management. Aliases allow for the grouping of IP addresses, networks, and ports under a single, easy-to-reference name. This simplifies rule creation and management.

Moving on to real-time monitoring, I accessed Log Files > Live View to gain insights into the firewall's current activities. This provided a dynamic view of the firewall's actions, offering valuable feedback.



Interface	Time	Source	Destination
lan	2022-07-23T15:48:16-04:00	192.168.2.10:51644	192.168.2.1:443
lan	2022-07-23T15:48:16-04:00	192.168.2.10:51643	192.168.2.1:443
wan	2022-07-23T15:48:11-04:00	192.168.10.6:123	204.93.207.12:123
wan	2022-07-23T15:48:08-04:00	192.168.10.6:11082	142.250.65.234:443
lan	2022-07-23T15:48:07-04:00	192.168.2.10:51642	192.168.2.1:443
lan	2022-07-23T15:48:07-04:00	192.168.2.10:51641	192.168.2.1:443
lan	2022-07-23T15:48:07-04:00	192.168.2.10:51640	192.168.2.1:443
wan	2022-07-23T15:48:02-04:00	192.168.10.6:123	38.229.54.9:123
wan	2022-07-23T15:48:00-04:00	[fe80::a00:27ff:feb9:5178]:546	[ff02::1:2]:547
Loopback	2022-07-23T15:48:00-04:00	[fe80::a00:27ff:feb9:5178]:546	[ff02::1:2]:547
wan	2022-07-23T15:47:59-04:00	192.168.10.6:57257	142.250.72.99:443
wan	2022-07-23T15:47:20-04:00	192.168.10.6:123	38.229.62.9:123

Additionally, the Overview section provided a graphical representation of firewall activity.

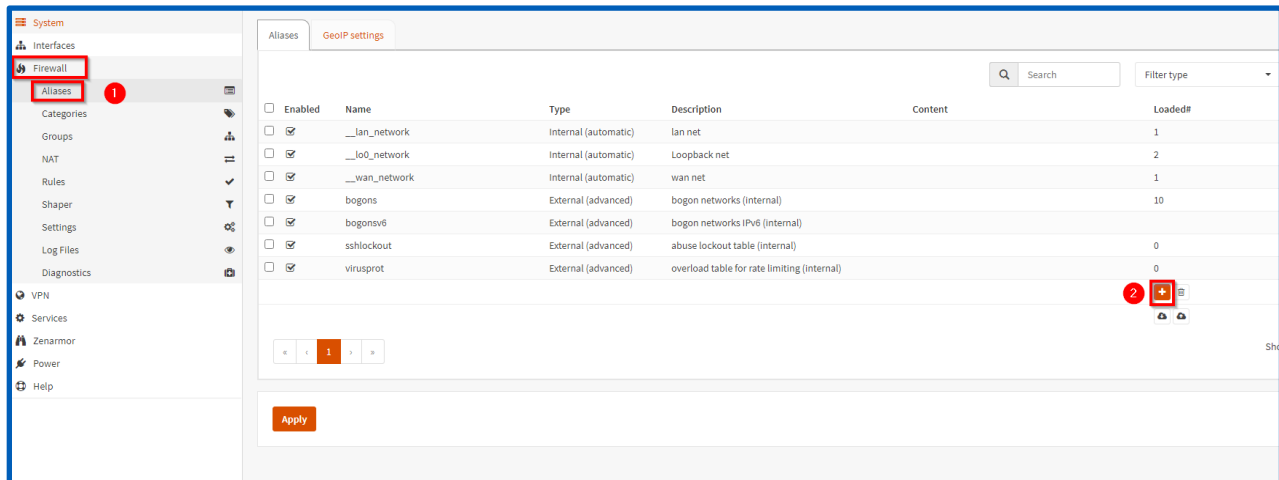


## Part 2 - Configure Aliases

Aliases play a pivotal role in organizing and simplifying administration tasks. They serve as placeholders for actual hosts, networks, or ports, allowing for streamlined management. In this demonstration, I meticulously configured aliases for the three devices on the network: Firewall, PC1, and PC2. This strategic use of aliases reduces the need to manually input IP addresses for each rule, enhancing efficiency.

I navigated to the Firewall section and accessed Aliases, an essential step to manage and organize entities effectively.

To create a new Alias, I clicked the plus sign, initiating the process of defining a placeholder for various network entities.



Make sure to check the IP address first by running ipconfig. For my case, it is 192.168.2.11. Before creating the Alias for PC1, I verified its IP address by running 'ipconfig,' noting it as 192.168.2.11.

I created an Alias for the firewall by entering the following:

**Name: Firewall\_Def\_Gateway.**

**Type: Host(s).**

**Content: 192.168.2.1**

**Description: OPNsense Default Gateway.**

**Click Save.**

I created an Alias named "**Firewall\_Def\_Gateway**" representing the OPNsense Default Gateway, specifying its type, content, and providing a description for clarity.

**Edit Alias** full help ☒

**Enabled** 1 ☒ Enable this alias

**Name** 2   
The name must start with a letter or single underscore, be less than 32 characters and only consist of alphanumeric characters or underscores. Aliases can be nested using this name.

**Type** 3

**Categories**

**Content** 4   
Clear All Copy Paste

**Statistics** ☐ Maintain a set of counters for each table entry

**Description** 5   
You may enter a description here for your reference (not parsed).

6 Save Cancel

I replicated the process to create an Alias named "AdminPC" for PC1, ensuring efficient management by using aliases rather than individual IP addresses.

Make sure to check the IP address first by running ipconfig. For my case, it is 192.168.2.11. Before creating the Alias for PC1, I verified its IP address by running 'ipconfig,' noting it as 192.168.2.11.

```
C:\Users\instructor>ipconfig /release 1
Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::3ce1:f02f:f0bb:37f7%13
    Default Gateway . . . . . : 

C:\Users\instructor>ipconfig /renew 2
Windows IP Configuration

An error occurred while renewing interface Ethernet : The name specified in the netv
The NCB is the data.

C:\Users\instructor>ipconfig 3
Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : localdomain
    Link-local IPv6 Address . . . . . : fe80::3ce1:f02f:f0bb:37f7%13
    IPv4 Address. . . . . : 192.168.2.11
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.2.1
```

Name: AdminPC  
Type: Host(s)  
Content: 192.168.2.11  
Description: PC1 (Admin)

The 'Edit Alias' dialog box contains the following fields and controls:

- Enabled:** A checkbox that is checked.
- Name:** A text input field containing 'AdminPC' (annotated with 1).
- Type:** A dropdown menu set to 'Host(s)' (annotated with 2).
- Content:** A text input field containing '192.168.2.11' (annotated with 3). Below the field are buttons for 'Clear All', 'Copy', and 'Paste'.
- Statistics:** An unchecked checkbox.
- Description:** A text input field containing 'PC1 (Admin)' (annotated with 4).
- Buttons:** 'Cancel' and 'Save' buttons at the bottom right, with 'Save' annotated with 5.

I established the "AdminPC" Alias with the corresponding details.

Following a similar approach, I created an Alias named "GuestPC" for PC2, obtaining its IP address as 192.168.2.10.

**Description: PC1 (Admin)**

### Edit Alias

Enabled

☒

Enable this alias

Name

1

The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and \_". Aliases can be nested using this name.

Type

Host(s)

2

Content

192.168.2.10

×

3

✖ Clear All

Statistics

☐

Maintain a set of counters for each table entry

Description

4

You may enter a description here for your reference (not parsed).

5

Save

Aliases

GeolIP settings

Search

<input type="checkbox"/> Enabled	Name	Type	Description	Content
<input checked="" type="checkbox"/>	AdminPC	Host(s)	PC1 (Admin)	192.168.2.11
<input checked="" type="checkbox"/>	Firewall	Host(s)	Opnsense Firewall	192.168.2.1
<input checked="" type="checkbox"/>	GuestPC	Host(s)	PC2 (Guest)	192.168.2.10

7

**This time, we will add an alias for all web traffic, so that we will not have to make separate rules for HTTP and HTTPS every time:** I created an Alias encompassing all web traffic, streamlining rule creation for both HTTP and HTTPS.

### Edit Alias

---

**Enabled** ☒ Enable this alias

**Name**  1  
The name of the alias may only consist of the characters "a-z, A-Z, 0-9 and \_". Aliases can be nested using this name.













**Type**  2

**Content**  3  
✖ Clear All

**Description**  4  
You may enter a description here for your reference (not parsed).

5 Save

I applied the changes, ensuring the new aliases are active.

Enabled	Name	Type	Descript...	Content	Loaded#	Last upd...	Commands
<input checked="" type="checkbox"/>	Firewall_Def_Gateway	Host(s)	OPNsense ...	192.168.2.1	1	2023-03-12...	  
<input checked="" type="checkbox"/>	AdminPC	Host(s)	PC1 (Admin)	192.168.2.10	1	2023-03-12...	  
<input checked="" type="checkbox"/>	GuestPC	Host(s)	Guest (PC2)	192.168.2.11	1	2023-03-12...	  
<input checked="" type="checkbox"/>	All_Web_Traffic	Port(s)	All Web Tra...	443,80			  
<input checked="" type="checkbox"/>	bogons	External (advanced)	bogon net...		10		
<input checked="" type="checkbox"/>	bogonsv6	External (advanced)	bogon net...				
<input checked="" type="checkbox"/>	virusprot	External (advanced)	overload ta...		0		

Showing 1 to 7 of 11 entries

« < 1 2 > »

Apply

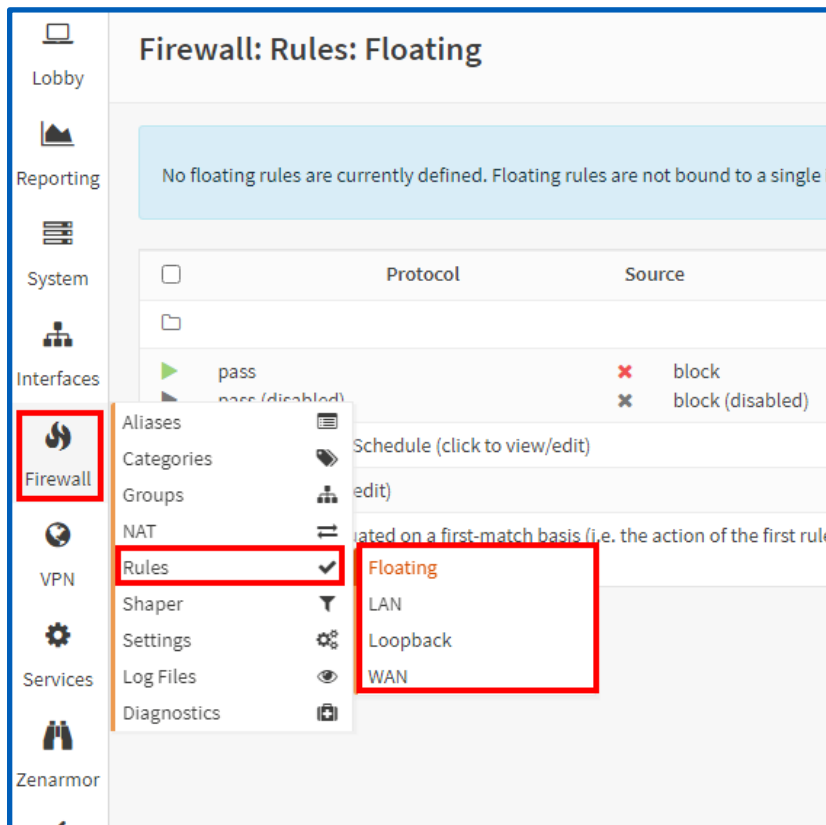


I emphasized the significance of aliases in simplifying configurations and enhancing organization.

With the aliases established, I proceeded to configure rules, leveraging the organized and simplified structure provided by aliases.

### Part 3 - Rules: Blocking IP's

I navigated to the Firewall tab and accessed the Rules section.



**Priority:** The firewall will go from top to bottom when reading rules; the rules on top will have the highest priority: Understanding the significance of rule priority, I ensured a top-to-bottom approach for effective rule processing.

**From top to bottom, the priority is as follows:**

- 1. System Rules:**
  - I acknowledged the existence of OPNsense's built-in rules, realizing their unchangeable and highest-priority nature.
- 2. Floating:**

- Emphasizing the versatility of floating rules, I considered their priority over interface-specific rules.

### 3. Interface Group:

- I highlighted the importance of grouping interface rules for efficient management.

### 4. Interfaces:

- Acknowledging individual interface rules, I stressed their lower priority but specificity.



## Blocking IP's:

In this phase, I created a LAN rule to prevent PC2 communication and ensured that PC2 could ping the firewall and access the internet.

Before proceeding with rule configurations, I initiated a series of ping tests from PC2 to both the firewall (192.168.2.1) and an external server (8.8.8.8, representing Google's DNS server). This step aimed to confirm PC2's current connectivity status.

```

C:\Users\groot>ping 192.168.2.1

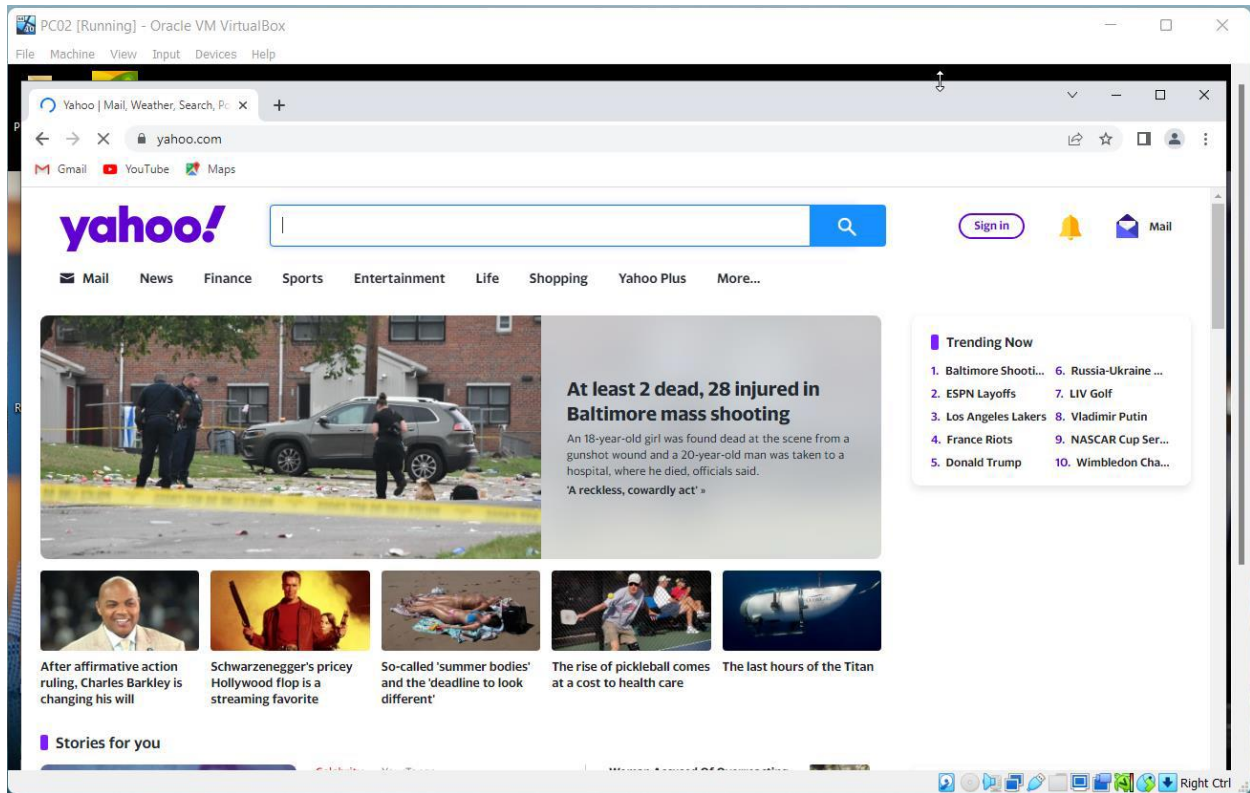
Pinging 192.168.2.1 with 32 bytes of data:
Reply from 192.168.2.1: bytes=32 time<1ms TTL=64
Reply from 192.168.2.1: bytes=32 time<1ms TTL=64
Reply from 192.168.2.1: bytes=32 time<1ms TTL=64
Reply from 192.168.2.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
  
```

```

C:\Users\groot>ping 8.8.8.8

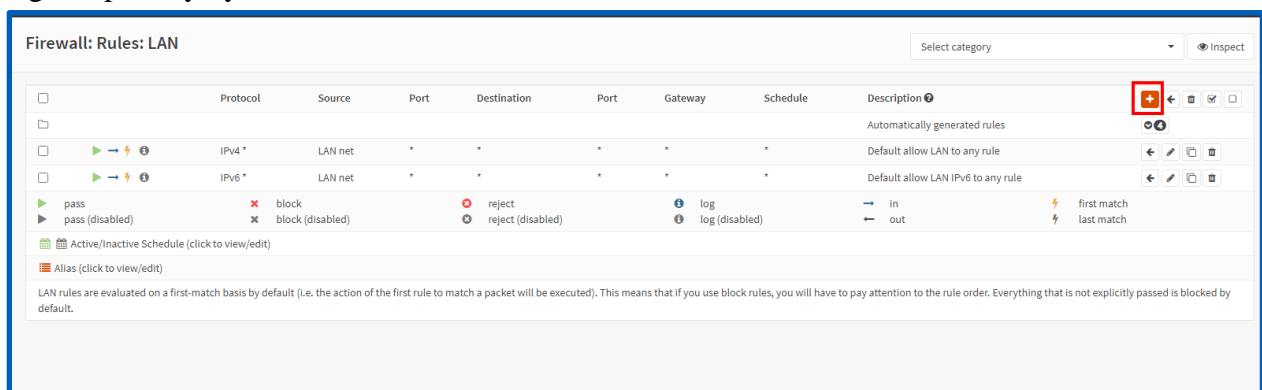
Pinging 8.8.8.8 with 32 bytes of data:
Reply from 8.8.8.8: bytes=32 time=8ms TTL=127
Reply from 8.8.8.8: bytes=32 time=5ms TTL=127
Reply from 8.8.8.8: bytes=32 time=12ms TTL=127
Reply from 8.8.8.8: bytes=32 time=11ms TTL=127
  
```



**Figure: A verification window showing PC2 can access the internet on the browser**

With the successful ping tests confirming PC2's connectivity, I navigated to PC1's OPNsense firewall settings and accessed the Rules section specifically for the LAN interface.

I observed the presence of automatically generated rules on the upper right, representing system rules. By clicking on the arrow next to "4," I could view these rules. These system rules have the highest priority by default.



I acknowledged the existence of default rules allowing all LAN traffic. While these rules can be deleted, they are initially in place to ensure basic connectivity. It's important to note that OPNsense follows a default deny-all philosophy, meaning that without specific rules, all traffic is denied at

the system level. I emphasized a security-oriented approach, suggesting that focusing on essential allowed traffic is more effective than trying to block specific "bad" things. This philosophy aligns with a well-configured network that employs aliases to allow only necessary traffic while blocking everything else by default.

This groundwork of connectivity verification and understanding default rules sets the stage for configuring specific rules to control traffic effectively.

To implement a rule for blocking PC2, I opted for the "**Reject**" action instead of "Block." This choice was deliberate, as the "Reject" action allows us to observe detailed logs of the blocked packets, providing transparency and insights into the traffic handling.

I set the direction of the rule to "in" since the pings from PC2 were considered incoming traffic to the firewall interface.



Ensuring compatibility with the IPv4 protocol, I specified this parameter in the rule configuration.

The screenshot shows the 'Edit Firewall rule' configuration page. It includes several settings, each highlighted with a red box and a numbered annotation (1-5):

- 1** Action: A dropdown menu set to 'Reject'.
- 2** Quick: A checkbox labeled 'Apply the action immediately on match.' which is checked.
- 3** Interface: A dropdown menu set to 'LAN'.
- 4** Direction: A dropdown menu set to 'in'.
- 5** TCP/IP Version: A dropdown menu set to 'IPv4'.

Other visible settings include 'Disabled' (checkbox) and 'Disable this rule' (checkbox), both of which are unchecked.

When defining the source for the rule, I utilized the alias functionality. Specifically, I searched for and selected the "guests" alias, streamlining the rule configuration process by referencing the pre-established alias for PC2.

The screenshot displays a firewall rule configuration interface. The 'Protocol' field is set to 'guest'. The 'Source / Invert' section is active, and the 'Source' field has a dropdown menu open. The dropdown menu shows 'Aliases' and 'GuestPC' as the selected option. The 'Source port range' section is set to 'from: any' and 'to: any'. The 'Destination / Invert' section is inactive, and the 'Destination' field is set to 'any'.

To encompass all other aspects not covered by the specified parameters, I set the remaining configurations to "any," effectively blocking all additional traffic.

To ensure comprehensive logging of packets affected by this rule, I enabled the option to log packets handled by the rule. This step enhances visibility into the firewall's activities and aids in troubleshooting.

<b>Protocol</b>	any	
<b>Source / Invert</b>	<input type="checkbox"/>	
<b>Source</b>	GuestPC	
<b>Source port range</b>	from: any	to: any
<b>Destination / Invert</b>	<input type="checkbox"/>	
<b>Destination</b>	any	
<b>Destination port range</b>	from: any	to: any
<b>Log</b>	<input checked="" type="checkbox"/> log packets that are handled by this rule	

Adhering to best practices, I provided values for both the category and description fields associated with the rule. This organizational approach contributes to a well-structured and easily manageable rule set.

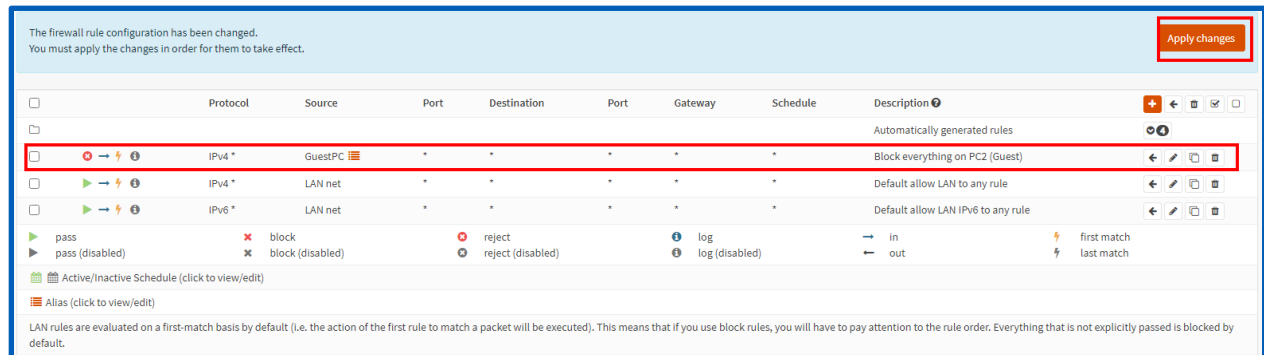
After configuring the rule parameters, I saved the settings to implement the rule within the OPNsense firewall.

Recognizing the importance of rule priority, I took the necessary step of reordering the rules to ensure that the newly created rule takes precedence over the default "allow all" rules. This proactive adjustment prevents potential conflicts and ensures the effective implementation of the intended block rule.

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description ?
<input type="checkbox"/>	IPv4 *	LAN net	*	*	*	*	*	Default allow LAN to any rule
<input type="checkbox"/>	IPv6 *	LAN net	*	*	*	*	*	Default allow LAN IPv6 to any rule
<input checked="" type="checkbox"/>	IPv4 *	GuestPC	*	*	*	*	*	Block everything on PC2 (Guest)

Utilizing the OPNsense interface, I selected the newly created rule and executed a priority adjustment, moving it ahead of the existing rules. This action ensures that the rule will be applied before other rules in the sequence.

With the rule configurations in place, I initiated the application of changes through the "Apply Changes" function within the OPNsense firewall interface. This step finalizes the rule adjustments and brings the changes into effect.



To validate the impact of the rule, I proceeded to PC2 and attempted to ping the firewall and access the internet. This real-world testing provides immediate feedback on the rule's effectiveness in blocking the specified traffic.

Following the testing on PC2, I assessed the outcomes to determine whether the rule successfully prevented ping requests and internet access. This evaluation serves as a crucial verification step for the configured rule.

To gain deeper insights into the firewall's behavior and understand the specifics of blocked traffic, I navigated back to the firewall settings and accessed the Log Files section. This step involves reviewing the logs to interpret the recorded events related to the applied rule.

Within the Log Files section, I specifically accessed the Live View feature. This real-time monitoring capability allows for the dynamic observation of firewall activities, providing immediate feedback on traffic events.

Upon examining the Live View, I visually confirmed the active operation of the rule. This verification step reinforces the rule's functionality and confirms that it is actively blocking the designated traffic.



Firewall: Log Files: Live View						
filter			25		<input checked="" type="checkbox"/> Auto refresh <input type="checkbox"/> Lookup hostnames	
Interface	Time	Source	Destination	Proto	Label	
lan	→ Jul 15 01:49:57	192.168.2.10:57733	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:57	192.168.2.10:57733	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:53	192.168.2.10:64946	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:52	192.168.2.10:53986	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:49	192.168.2.10:64946	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:48	192.168.2.10:53986	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:47	192.168.2.10:64946	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:46	192.168.2.10:53986	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:46	192.168.2.10:64946	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:45	192.168.2.10:64946	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:45	192.168.2.10:53986	192.168.2.1:53	udp	Block everything on PC2 (Guest)	
lan	→ Jul 15 01:49:45	192.168.2.10:53986	192.168.2.1:53	udp	Block everything on PC2 (Guest)	

In the interest of testing and experimentation, I opted to disable the rule temporarily. This action allows for the observation of changes in PC2's connectivity when the rule is not in effect.

	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description
Automatically generated rules								
<input checked="" type="checkbox"/>	IPv4 *	*	*	*	*	*	*	Block everything on PC2 (Guest)
<input type="checkbox"/>	IPv4 *	LAN net	*	*	*	*	*	Default allow LAN to any rule
<input type="checkbox"/>	IPv6 *	LAN net	*	*	*	*	*	Default allow LAN IPv6 to any rule
pass	block			reject			log	in
pass (disabled)	block (disabled)			reject (disabled)			log (disabled)	out

### Part 3 - Rules: Blocking specific protocols

To enforce a rule blocking ICMP traffic from PC2 to the firewall, I navigated to the LAN rules section within the OPNsense firewall interface.

As a prerequisite, I ensured that PC2 could ping the firewall before implementing the rule. This preliminary verification step helps establish a baseline for testing the rule's effectiveness.

```
C:\Users\groot>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:
Reply from 192.168.2.1: bytes=32 time<1ms TTL=64
Reply from 192.168.2.1: bytes=32 time<1ms TTL=64
Reply from 192.168.2.1: bytes=32 time<1ms TTL=64
Reply from 192.168.2.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Returning to the Firewall Rules page under LAN settings, I prepared to create a new rule to regulate ICMP traffic.

I initiated the rule creation process by clicking the "Add" button, signaling my intent to define a new rule within the LAN rules configuration.

**Edit Firewall rule**

**Action**  
Reject  
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 is sent back to the sender, whereas with block the packet is dropped silently. In either case, the packet is not delivered to its destination.

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

**Quick**  
☒ Apply the action immediately on match.  
If a packet matches a rule specifying quick, then that rule is considered a match. When a rule does not have quick enabled, the last matching rule wins.

**Interface**  
LAN  
Choose on which interface packets must come in to match this rule.

In configuring the rule, I opted for the "Reject" action instead of "Block" to allow for logging of blocked packets. This decision enhances visibility into the rule's impact on ICMP traffic.

Maintaining consistency with the direction of ICMP traffic, I specified the rule to operate on incoming traffic, aligning with the nature of pings directed towards the firewall interface.

In defining the protocol for this rule, I set it specifically for ICMP traffic, targeting the ping functionality.

To streamline the rule configuration, I utilized the GuestPC alias by typing "guest" in the source field, eliminating the need to remember or manually input the IP address for PC2.

The screenshot displays a firewall rule configuration interface. The 'Protocol' field is set to 'ICMP' and is highlighted with a red rectangle. The 'Source' field is set to 'any', and a dropdown menu is open, showing the 'Aliases' section with 'GuestPC' selected. The 'Destination / Invert' field is partially visible at the bottom.

Direction	in
TCP/IP Version	IPv4
Protocol	ICMP
ICMP type	any
Source / Invert	<input type="checkbox"/>
Source	any
Destination / Invert	<input type="checkbox"/>

In the destination field, I entered "firewall" to locate and select the alias associated with the firewall. It's noteworthy that an existing alias for the firewall is available by default in the drop-down menu.

To ensure comprehensive logging of packets affected by this rule, I activated the option to log packets handled by the rule. Enabling this feature contributes to a detailed understanding of the firewall's actions.

Adhering to best practices, I assigned values to both the category and definition fields, promoting an organized and easily understandable rule set.

After configuring the rule parameters, I saved the settings and applied the changes. This step ensures that the rule becomes active within the firewall.

The screenshot shows a 'Firewall Rule Configuration' dialog box with the following fields and options:

- Destination:** A dropdown menu set to 'Firewall'.
- Destination port range:** Two input fields labeled 'from:' and 'to:', both containing the text 'any'.
- Log:** A checkbox labeled 'Log packets that are handled by this rule' which is checked.
- Category:** A text input field containing the word 'Demo'.
- Description:** A text input field containing the text 'Block GuestPC pings to Firewall'.
- Buttons:** At the bottom, there are two buttons: 'Save' (highlighted with a red border) and 'Cancel'.

As part of the testing process, I proceeded to PC2 and attempted to ping the firewall. This step allows for immediate feedback on whether the rule successfully blocks ICMP traffic from PC2.

Returning to the firewall rules configuration, I accessed the relevant section to review and manage the rules currently in place.

The screenshot shows the 'Firewall Rules' configuration page. At the top, a light blue banner states: 'The firewall rule configuration has been changed. You must apply the changes in order for them to take effect.' with an 'Apply changes' button. Below is a table of rules:

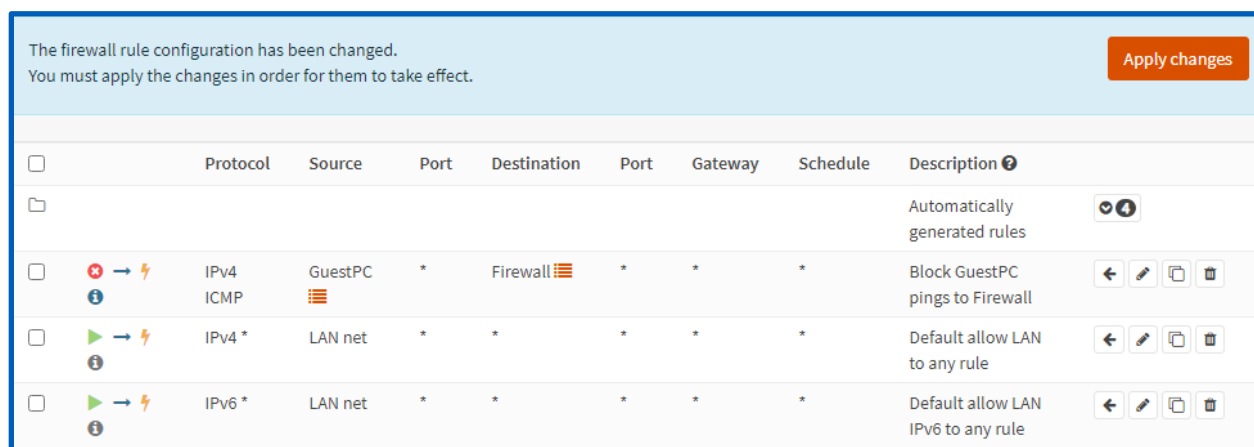
	Protocol	Source	Port	Destination	Port	Gateway	Schedule	Description ?
<input type="checkbox"/>								Automatically generated
<input type="checkbox"/>	IPv4 *	LAN net	*	*	*	*	*	Default allow LAN to any rule
<input type="checkbox"/>	IPv6 *	LAN net	*	*	*	*	*	Default allow LAN IPv6 to any rule
<input checked="" type="checkbox"/>	IPv4 ICMP	GuestPC	*	Firewall	*	*	*	Block GuestPC pings to Firewall

Annotations in the screenshot include a green circle around the checkbox for the 'Block GuestPC pings to Firewall' rule, and a tooltip that says 'Move selected rules before this rule' pointing to the left arrow icon in the rule's action column.

Recognizing the importance of rule priority, I adjusted the rule sequence to ensure that the ICMP block rule takes precedence over the default "allow all" IPv4 rules. This adjustment is crucial to guarantee the rule's effectiveness.

Using the OPNsense interface, I executed the necessary steps to move the ICMP block rule to a position of higher priority, ensuring it is processed before the less restrictive IPv4 allow all rule.

With the rule sequence adjusted, I applied the changes within the OPNsense interface and proceeded to retest PC2's ability to ping the firewall. This iterative testing approach helps verify the rule's functionality.



Post-adjustment testing allowed me to assess whether the rule modifications were effective in blocking ICMP traffic from PC2. This step serves as a crucial validation of the rule adjustment.

To gain deeper insights into the firewall's behavior and verify the logging functionality, I navigated to the Log Files section within OPNsense and accessed the Live View feature. This step allows for real-time monitoring and verification of the rule's impact on traffic.

▶ wan	←	2022-07-23T20:44:05-04:00	192.168.10.6:38403	142.251.32.106:443	udp	let out anything from firewall host itself (force gw)	🔍
▶ wan	←	2022-07-23T20:44:05-04:00	192.168.10.6:4931	216.239.34.10:53	udp	let out anything from firewall host itself (force gw)	🔍
▶ wan	←	2022-07-23T20:44:02-04:00	192.168.10.6:123	104.131.139.195:123	udp	let out anything from firewall host itself (force gw)	🔍
🔍 lan	→	2022-07-23T20:43:49-04:00	192.168.2.10	192.168.2.1	icmp		🔍
🔍 lan	→	2022-07-23T20:43:44-04:00	192.168.2.10	192.168.2.1	icmp		🔍
🔍 lan	→	2022-07-23T20:43:39-04:00	192.168.2.10	192.168.2.1	icmp		🔍
▶ wan	←	2022-07-23T20:43:36-04:00	192.168.10.6:64044	142.250.80.36:443	udp	let out anything from firewall host itself (force gw)	🔍
🔍 lan	→	2022-07-23T20:43:34-04:00	192.168.2.10	192.168.2.1	icmp		🔍
▶ wan	←	2022-07-23T20:43:34-04:00	192.168.10.6:34011	51.104.15.253:443	tcp	let out anything from firewall host itself (force gw)	🔍
▶ wan	←	2022-07-23T20:43:34-04:00	192.168.10.6:21197	64.48.1.53	udp	let out anything from firewall host itself (force gw)	🔍

Within the Log Files section, I specifically accessed the Live View feature. This real-time monitoring capability allows for the dynamic observation of firewall activities, providing immediate feedback on traffic events.

Through the Live View, I visually confirmed the active operation of the rule, validating that ICMP traffic from PC2 to the firewall was successfully blocked. This confirmation reinforces the effectiveness of the rule.

<input type="text" value="filter"/> <span>25</span> <input checked="" type="checkbox"/> Auto refresh <input type="checkbox"/> Lookup hostnames						
Interface	Time	Source	Destination	Proto	Label	
lan	→ Jul 15 15:25:13	192.168.2.10:53224	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53222	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53223	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53221	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53225	13.68.225.90:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53226	13.68.225.90:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53216	13.68.225.90:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53215	13.68.225.90:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53224	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53220	204.79.197.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53223	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53222	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:13	192.168.2.10:53221	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:12	192.168.2.10:53222	13.107.21.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:12	192.168.2.10:53218	204.79.197.200:443	tcp	Block all web traffic ports on PC2	ⓘ
lan	→ Jul 15 15:25:12	192.168.2.10:53219	204.79.197.200:443	tcp	Block all web traffic ports on PC2	ⓘ

## Concluding Remark

By navigating firewall interfaces and implementing rules, I successfully configured and tested various security scenarios. The project covered crucial aspects, including creating aliases and applying rules for enhanced security. The emphasis on real-time verification and log analysis deepened my understanding of the firewall's functionality. As I conclude this project, I've not only applied OPNsense firewall features practically but also gained a foundation for effective network security management in diverse settings.