

535 pfSense - NAT and Firewall Rules

Step 1: Creating NAT Port-Forwarding Rules

- With a sense of purpose, I delved into the realm of network address translation (NAT) within pfSense, accessing the web interface with eager anticipation.
- Navigating to Firewall > NAT, I embarked on the journey of creating port-forwarding rules by switching to the "Port Forward" tab.
- With a few clicks of the mouse, I initiated the creation of a new port forwarding rule, selecting the WAN interface to direct incoming traffic from the outside world to specific services within my LAN.
- Methodically configuring the rule, I specified the protocol, destination port range, and the internal IP address of the server or device to which the traffic should be forwarded.
- With a sense of satisfaction, I saved the rule and applied changes, confident in its ability to efficiently manage incoming traffic.

Firewall: Aliases: Edit ?

Alias Edit

Name	<input type="text" value="Computer1"/> <small>The name of the alias may only consist of the characters "a-z, A-Z and 0-9".</small>				
Description	<input type="text" value="IP address of Computer1."/> <small>You may enter a description here for your reference (not parsed).</small>				
Type	Host(s) ▼				
Host(s)	<div><div>Enter as many hosts as you would like. Hosts must be specified by their IP address.</div><table><thead><tr><th>IP</th><th>Description</th></tr></thead><tbody><tr><td>192.168.1.200</td><td><input type="text" value="The IP address of Computer1."/></td></tr></tbody></table></div>	IP	Description	192.168.1.200	<input type="text" value="The IP address of Computer1."/>
IP	Description				
192.168.1.200	<input type="text" value="The IP address of Computer1."/>				

Step 2: Developing Firewall Rules to Control Traffic

- Transitioning seamlessly into the realm of firewall rules, I navigated to Firewall > Rules, ready to define rules to allow or deny traffic based on my network policies.

- With a clear understanding of my objectives, I meticulously crafted each firewall rule, specifying the action (pass, block, or reject), protocol, source, destination, and port as needed.
- Taking care to arrange the rules in the desired order, I ensured that they would be processed from top to bottom, with the first matching rule taking precedence.
- With a sense of purpose, I saved the rules and applied changes, eager to test their efficacy in controlling and directing traffic within my network.

Type

Network(s)

Network(s)

Networks are specified in CIDR format. Select the CIDR mask that pertains to each entry. /32 specifies a single host, /24 specifies 255.255.255.0, etc. Hostnames (FQDNs) may also be specified, using a /32 mask. You may also enter an IP range such as 192.168.1.1-192.168.1.254 and a list of CIDR networks will be derived to fill the range.

Network	CIDR	Description
192.168.0.0	16	CIDR format of a typical private network.
192.168.0.0-192.168.255.255	32	Range format of a typical private network.
www.bunkerhollow.com	32	An example of a FQDN hostname.

Type

Port(s)

Port(s)

Enter as many ports as you wish. Port ranges can be expressed by seperating with a colon.

Port	Description
12345	An individual port.
55100:55199	A range of ports.

Firewall: Aliases: Edit



Alias Edit

Name	<input type="text" value="voip_all_phones"/> <small>The name of the alias may only consist of the characters "a-z, A-Z and 0-9".</small>												
Description	<input type="text" value="All VoIP Phones"/> <small>You may enter a description here for your reference (not parsed).</small>												
Type	Host(s) ▾												
Host(s)	<div><div>Enter as many hosts as you would like. Hosts must be specified by their IP address.</div><table><thead><tr><th>IP</th><th></th><th>Description</th></tr></thead><tbody><tr><td>voip_phone1</td><td>32 ▾</td><td><input type="text"/></td></tr><tr><td>voip_phone2</td><td>32 ▾</td><td><input type="text"/></td></tr><tr><td>voip_phone3</td><td>32 ▾</td><td><input type="text"/></td></tr></tbody></table><div></div></div>	IP		Description	voip_phone1	32 ▾	<input type="text"/>	voip_phone2	32 ▾	<input type="text"/>	voip_phone3	32 ▾	<input type="text"/>
IP		Description											
voip_phone1	32 ▾	<input type="text"/>											
voip_phone2	32 ▾	<input type="text"/>											
voip_phone3	32 ▾	<input type="text"/>											

Save Cancel

Step 3: Testing and Validating Rules with Various Network Devices

- With a spirit of curiosity, I embarked on the crucial phase of testing and validating the configured rules, ensuring they functioned as intended.
- Testing NAT port forwarding, I attempted to access internal services from external devices using the WAN IP and the forwarded ports, verifying their accessibility.
- Transitioning to testing firewall rules, I generated traffic that matched the defined rules, carefully observing the firewall logs to confirm whether the traffic was allowed or blocked as expected.
- With each successful test and validation, I iterated on the rules as necessary, making modifications and adjustments until achieving the desired outcomes.

Verification and Troubleshooting:

- In the event of any discrepancies or unexpected behavior, I diligently reviewed the configured NAT port-forwarding rules, ensuring the correctness of WAN IP addresses, internal IPs, and port settings.

- Similarly, I meticulously scrutinized the firewall rules, verifying their order and structure to identify any overlapping or conflicting rules that may affect traffic flow.
- Leveraging the diagnostic tools within pfSense, I meticulously reviewed logs and system states, using the insights gained to troubleshoot and resolve any issues related to blocked or allowed traffic.

With a sense of accomplishment, I successfully configured NAT and firewall rules within pfSense, empowering myself to control and direct traffic through my network with precision and confidence.