

Firewall Quickstart for Vultr Cloud Servers

Learn how to quickly set up and configure firewalls for your Vultr Cloud Servers to enhance security and protect your infrastructure from unauthorized access.

Contents

01	Introduction	3
02	Which Firewall Does My Server Use?	3
03	Firewalld Quickstart	3
04	IPFW Quickstart	6
05	Packet Filter (pf) Quickstart	8
06	IP Filter (ipf) Quickstart	10
07	Uncomplicated Firewall (UFW) Quickstart	11
08	nftables Quickstart	13
09	Windows Firewall Quickstart	15
010	iptables Quickstart	17

Introduction

For security, the firewall is enabled when you first deploy a Vultr cloud server. Your server's firewall software varies depending on the operating system you deploy. This guide explains how to determine which firewall you have, allow and deny traffic, and learn more about your firewall.

Which Firewall Does My Server Use?

Depending on your operating system, your cloud server may use one of these firewalls:

- **Firewalld**: Fedora, CentOS 7, and other distributions based on Red Hat or SUSE Linux use this configuration tool for iptables.
- **IPFW**: FreeBSD
- **Packet Filter (pf)**: OpenBSD
- **IP Filter (ipf)**: FreeBSD, NetBSD, OpenBSD, and Solaris
- **Uncomplicated Firewall (UFW)**: Ubuntu's configuration tool for iptables.
- **nftables**: Debian servers use this upgrade to iptables.
- **Windows Firewall**: Microsoft Windows
- **iptables**: Many OS firewall utilities are front-ends for iptables, which is discussed at the end of this article.

Firewalld Quickstart

Firewalld is the default software firewall for Fedora, CentOS 7, and other modern distributions based on Red Hat or SUSE Linux. This quickstart guide outlines several useful commands and techniques to assist in debugging Firewalld.

Verify firewalld is active

```
$ firewall-cmd --state
running
```

Check the zones assigned to active interfaces

```
$ firewall-cmd --get-active-zones
public
  interfaces: ens3
```

Check which ports and services are allowed

Assuming your active zone is **public**, this quick check reveals what traffic is allowed.

```
$ firewall-cmd --zone=public --list-ports
7000-8000/tcp

$ firewall-cmd --zone=public --list-services
cockpit dhcpv6-client ssh
```

Example: Allow SSH

Assuming your active zone is **public**, use either of these two methods to allow SSH.

```
$ firewall-cmd --zone=public --add-service=ssh
```

or

```
# firewall-cmd --add-port=22/tcp
```

Panic Mode

Drop All Packets

As root, use the `--panic-on` switch.

```
# firewall-cmd --panic-on
```

All packets will be dropped. Active connections will be terminated after a period of inactivity.

Panic Mode Off

As root, use the `--panic-off` switch.

```
# firewall-cmd --panic-off
```

Check Panic Mode Status

```
firewall-cmd --query-panic && echo "enabled" || echo "Not enabled"
```

Permanent vs. Temporary Configuration

Temporary changes cause a common issue; the server works as expected until the next reboot. Make sure you permanently save your configuration.

To make a command permanent, add the `--permanent` option to all commands except `--direct` commands (which are temporary by nature). Setting made with the `--permanent` option do not take effect until the next firewall reload, service restart, or system reboot. Settings made without the `--permanent` option take effect immediately but are only valid until the next firewall reload, system boot, or service restart.

Disable firewalld

As root, mask and disable the service.

```
# systemctl mask --now firewalld.service
# systemctl disable --now firewalld.service
```

More Information

- [firewalld Documentation](#)

IPFW Quickstart

IPFW is a FreeBSD stateful firewall and packet filter. This quickstart guide outlines several useful commands and techniques to assist in debugging IPFW.

Enable and start IPFW

To enable IPFW at boot, add `firewall_enable="YES"` to `/etc/rc.conf`:

```
# sysrc firewall_enable="YES"
```

Start the firewall.

```
# service ipfw start
```

List all running rules.

```
# ipfw list
```

Delete all rules.

```
# ipfw -q -f flush
```

Disable and stop IPFW

Stop the firewall.

```
# /etc/rc.d/ipfw stop
```

To disable the firewall, set the following option in `/etc/rc.conf` file:

```
firewall_enable="NO"
```

Example: Allow SSH and deny all others

This example uses 192.0.2.123 as the server's IP address.

Allow anything outbound from this address.

```
# ipfw -q add allow all from 192.0.2.123 to any out
```

Deny anything outbound from other addresses.

```
# ipfw -q add deny log all from any to any out
```

Allow TCP through if setup succeeds.

```
# ipfw -q add allow tcp from any to any established
```

Allow IP fragments

```
# ipfw -q add allow all from any to any frag
```

Allow inbound ssh

```
# ipfw -q add allow tcp from any to 192.0.2.123 22 setup
```

Everything else is denied and logged.

```
# ipfw -q add deny log all from any to any
```

Permanent vs. Temporary Rules

It's possible to make changes on-the-fly to the `ipfw` configuration without saving permanently. This causes a common issue; the server works as expected until the next reboot. Make sure you permanently save your configuration.

To make your rules permanent, put your rules into a file such as `/etc/ipfw.conf`, then add this to `/etc/rc.conf`:

```
firewall_enable="YES"
firewall_type="/etc/ipfw.conf"
```

An example `/etc/ipfw.conf` to allow SSH and deny all others looks like this:

```
# =====
# IPFW Example - Allow SSH, deny all other
# 192.0.2.123 is the example IP address
# =====

# Allow anything outbound from this address.
add allow all from 192.0.2.123 to any out

# Deny anything outbound from other addresses.
add deny log all from any to any out

# Allow TCP through if setup succeeded.
add allow tcp from any to any established

# Allow IP fragments
add allow all from any to any frag

# Allow inbound ssh
add allow tcp from any to 192.0.2.123 22 setup

# Everything else is denied and logged.
add deny log all from any to any
```

More information

See the [IPFW documentation](#) for more details.

Packet Filter (pf) Quickstart

OpenBSD Packet Filter (pf) is a stateful packet filter firewall. pf was developed for OpenBSD, but has been ported to many other operating systems. This quickstart guide outlines several useful commands and techniques to assist in debugging pf.

Enable and start pf

To enable pf at boot, add `pf_enable=yes` to `/etc/rc.conf`:

```
# sysrc pf_enable=yes
```

Start pf manually.

```
# pfctl -e
```

View the pf ruleset

Show the current ruleset.

```
# pfctl -sr
```

Show everything possible.

```
# pfctl -sa
```

Stop and disable pf

Stop pf.

```
# pfctl -d
```

Disable pf at boot.

```
# rcctl disable pf
```

Example: Allow SSH, block all other

This trivial example will allow SSH into the server while blocking everything else. Add the following to `/etc/pf.conf`.

```
block all
pass out proto tcp to any port 22 keep state
```

More Information

See the [pf documentation](#) for more details.

IP Filter (ipf) Quickstart

IP Filter (commonly referred to as **ipf**) is an open-source firewall available on several operating systems, including FreeBSD, NetBSD, OpenBSD, and Solaris. IPFILTER is included in the basic FreeBSD install as a kernel loadable module. This quickstart guide provides a few helpful commands and techniques to assist in debugging IPFilter.

Start ipf

```
# service ipfilter start
```

View the active packet filtering ruleset

```
ipfstat -io
```

Remove all filtering rules from the ruleset

```
ipf -F a
```

Stop ipf

```
# service ipfilter stop
```

Example: Allow SSH, deny all other

Add the following to `/etc/ipf.conf` for a trivial firewall that blocks everything except SSH (port 22) for example IP 192.0.2.123.

```
block in on any all
pass in quick on any proto tcp from any to 192.0.2.123/32 port = 22 keep state
```

More Information

See the [ipf documentation](#) for more details.

Uncomplicated Firewall (UFW) Quickstart

UFW is the default firewall configuration tool for Ubuntu. This quickstart guide outlines several useful commands and techniques to assist in debugging UFW.

Enable UFW

Enable UFW with the default set of rules:

```
$ sudo ufw enable
```

View status

Check the status of the server firewall with `sudo ufw status`. You may see one of these results:

UFW is not installed

```
$ sudo ufw status
ufw: command not found
```

UFW is installed but not configured

```
$ sudo ufw status  
Status: inactive
```

UFW is running

The firewall rules in force are displayed.

```
$ sudo ufw status verbose  
Status: active  
  
To Action From  
--  
22 ALLOW Anywhere  
22 (v6) ALLOW Anywhere (v6)
```

Disable UFW

```
$ sudo ufw disable
```

Reset UFW to default

```
$ sudo ufw reset
```

Examples

Allow SSH, deny all others

An example that blocks all inbound traffic except SSH (port 22).

```
$ sudo ufw default deny incoming  
$ sudo ufw default allow outgoing  
$ sudo ufw allow ssh  
$ sudo ufw enable  
$ sudo ufw reload
```

Allow port 80 (HTTP) and 443 (HTTPS), deny all others

An example that blocks all inbound traffic except HTTP and HTTPS.

```
$ sudo ufw default deny incoming
$ sudo ufw default allow outgoing
$ sudo ufw allow 80/tcp
$ sudo ufw allow 443/tcp
$ sudo ufw enable
$ sudo ufw reload
```

nftables Quickstart

nftables provides firewall support and NAT. This quickstart guide outlines several useful commands and techniques to assist in debugging nftables.

Enable and start nftables

Recent versions of Debian have nftables installed by default.

If you need to install nftables:

```
# aptitude install nftables
```

To enable nftables at boot:

```
# systemctl enable nftables.service
```

List current ruleset

```
# nft list ruleset
```

Delete all rules

To stop nftables from filtering traffic, delete all the rules.

```
nft flush ruleset
```

Disable and stop nftables

To disable nftables from starting:

```
# systemctl mask nftables.service
```

To uninstall nftables:

```
# aptitude purge nftables
```

Simple example for SSH and web

This trivial example allows SSH, HTTP, HTTPS, and ICMP. It denies all other inbound traffic.

Edit `/etc/nftables.conf`.

```
sudo nano /etc/nftables.conf
```

Replace `/etc/nftables.conf` with the following rules.

```
#!/usr/sbin/nft -f

flush ruleset

table inet filter {
    chain input {
        type filter hook input priority 0; policy drop;

        # accept any localhost traffic
        iif lo accept

        # accept traffic originated from us
        ct state established,related accept

        # drop invalid packets
        ct state invalid counter drop
    }
}
```

```
# accept ssh, http, and https
tcp dport { 22, 80, 443 } accept

# accept icmp
ip protocol icmp accept

# count and reject everything else
counter reject with icmpx type admin-prohibited
}

chain forward {
    type filter hook forward priority 0; policy drop;
}

chain output {
    type filter hook output priority 0; policy accept;
}

}
```

More information

See <https://wiki.debian.org/nftables> for more details.

Windows Firewall Quickstart

Windows Firewall with Advanced Security can be accessed from the GUI, a command prompt, or PowerShell. This quickstart guide outlines several useful commands and techniques to assist in debugging Windows Firewall. For more details, see our guide [How to Configure the Firewall on Windows Server 2019](#).

Run Windows Firewall from the GUI

If using the GUI, use one of the following methods to launch the Windows Firewall.

From Windows UI

1. Click **Search**
2. Type **Windows Firewall**
3. Select **Windows Firewall with Advanced Security**

MMC

- Add snap-in for **Windows Firewall with Advanced Security**.

From the command line

- Type `wf.msc`.

To turn off the Windows Firewall with Advanced Security console

1. Open **Server Manager**.
2. Select **Windows Firewall with Advanced Security** from the **Tools** menu.
3. In the center pane, click **Windows Firewall Properties**.
4. There are three profile tabs in the properties window, corresponding to the three Windows Firewall profiles (domain, private, and public). In each profile tab, select **Off** from the **Firewall state** dropdown list.
5. Click **OK** to close the firewall properties window.

Enable and disable Windows Firewall from the command line

These commands must be run from an administrative command prompt or PowerShell.

Turn on the firewall using netsh

```
netsh advfirewall set allprofiles state on
```

Turn off the firewall using netsh

```
netsh advfirewall set allprofiles state off
```

Turn on the firewall using Windows PowerShell

```
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True
```

Turn off the firewall using Windows PowerShell

```
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
```

Additional resources

For more help with Windows Firewall using PowerShell, see the Microsoft article [Windows Defender Firewall with Advanced Security Administration with Windows PowerShell](#)

For more details about using Group Policy or MMC snap-ins, [refer to the Microsoft documentation](#).

iptables Quickstart

iptables is a user-space utility program that allows you to configure the IP packet filter rules of the Linux kernel firewall.

List All Running Rules

To view the current firewall rules:

```
# iptables -L -v
```

Disable and Flush all Rules

To disable the firewall temporarily, flush all rules.

```
$ sudo iptables -P INPUT ACCEPT
$ sudo iptables -P OUTPUT ACCEPT
$ sudo iptables -P FORWARD ACCEPT
$ sudo iptables -F
```

Deny all traffic

To block everything, drop all packets on all chains.

```
$ sudo iptables -P INPUT DROP
$ sudo iptables -P OUTPUT DROP
$ sudo iptables -P FORWARD DROP
```

A Common Example

Here is a typical example of allowing SSH, HTTP, and HTTPS but dropping everything else.

Step 1

Append a rule to the INPUT chain:

- Protocol TCP
- Destination port 22, 80 & 443

For those packets, jump to ACCEPT.

```
$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Step 2

Append a rule to the INPUT chain: Drop everything else.

```
$ sudo iptables -A INPUT -j DROP
```

More information

To learn more about iptables, see the [Ubuntu](#) and [CentOS](#) guides.



VULTR

