How To Configure Firewall with UFW on Ubuntu 20.04 LTS

https://www.cyberciti.biz/faq/how-to-configure-firewall-with-ufw-on-ubuntu-20-04-lts/

How do I set up and configure firewall with UFW on Ubuntu 20.04 LTS server?

UFW is an acronym for an uncomplicated firewall. Securing a network with an uncomplicated firewall is super easy and highly recommended. This page explains how to set up and secure your Ubuntu 20.04 LTS server with ufw.

Tutorial requirements	
Requirements	Ubuntu Linux 20.04 LTS
Root privileges	Yes
Difficulty level	Easy
Category	Firewall
Prerequisites	ufw command
Est. reading time	7 minutes

Table of contents ↓

Step 1 – Set Up default UFW policies

To view status of ufw, type:

```
$ sudo ufw status
```

Sample outputs:

```
Status: inactive
```

The default policy firewall works out great for both the servers and desktop. It is always a good policy to closes all ports on the server and open only required ports one by one. Let us block all incoming connection and only allow outgoing connections from the Ubuntu 20.04 LTS box:

```
$ sudo ufw default allow outgoing
$ sudo ufw default deny incoming
```

Enabling IPv6 support

Make sure the directive `IPV6=yes` do exists in **`/etc/default/ufw`** file. For instance:

```
$ cat /etc/default/ufw
```

Step 2 – Open SSH TCP port 22 connections

The next logical step is to allow incoming SSH ports. We can easily open SSH TCP port 22 using UFW as follows:

```
$ sudo ufw allow ssh
```

If you are running ssh on TCP port 2222 or TCP port 2323, enter:

```
$ sudo ufw allow 2222/tcp
```

```
$ sudo ufw allow 2323/tcp
```

Some sysadmins have a static IP address (such as 202.54.2.5) at home or office location. In that case, only allow ssh access from the static IP address such as 202.54.2.5 to Ubuntu server IP address 172.24.13.45:

```
$ sudo ufw allow proto tcp from 202.54.2.5 to 172.24.13.45 port 22
```

But how do I find out my static IP 202.54.2.5 on Ubuntu server itself? Try the w command or lastlog command:

```
$ w
```

```
$ lastlog -u {YOUR_ADMIN_LOGIN_NAME_HERE}
```

```
$ lastlog -u vivek
```

And here is what I see:

```
Username         Port     From             Latest
vivek            pts/0    202.54.2.5       Thu Sep 29 15:19:21 +0000 2022
```

Next, let us limit ssh port, run:

```
$ sudo ufw limit ssh
```

See “How to limit SSH (TCP port 22) connections with ufw on Ubuntu Linux” for more information.

Step 3 – Turn on firewall

Now we got basic configuration enabled. In other words, the firewall will drop all incoming traffic except for ssh TCP port 22. Let us true it on the UFW, enter:

```
$ sudo ufw enable
```

Remember, once UFW enabled, it runs across system reboots too. We can verify that easily as follows using the systemctl command:

```
$ sudo systemctl status ufw.service
```

Want to disable the UFW based firewall? Try

If you need to stop the firewall and disable on system startup, eenter:

```
$ sudo ufw disable
```

Sample outputs:

```
Firewall stopped and disabled on system startup
```

Step 4 – Open specific incoming connections/ports

Let us add more rules. Say you want to open ports and allow IP address with ufw. The syntax is as follows to open TCP port 80 and 443:

```
$ sudo ufw allow 80/tcp comment 'accept Apache'
$ sudo ufw allow 443/tcp comment 'accept HTTPS connections'
```

Open UDP/1194 (OpenVPN) server:

```
$ sudo ufw allow 1194/udp comment 'OpenVPN server'
```

Allow port ranges via ufw

We can allow port ranges too say, tcp and udp 3000 to 4000:

```
$ sudo ufw allow 3000:4000/tcp
$ sudo ufw allow 3000:4000/udp
```

In this example, you want to allow ALL connections from an IP address called 104.22.10.214, enter:

```
$ sudo ufw allow from 104.22.10.214
```

Let us allow connections from an IP address called 104.22.11.213 to our port 25, enter:

```
$ sudo ufw allow from 104.22.11.213 to any port 25 proto tcp
```

We can set dest IP 222.222.222.222 for port 25 too:

```
$ sudo ufw allow from 104.22.11.213 to 222.222.222.222 port 25 proto tcp
```

Allow connection on specific interface

Open port 22 for wg0 interface only:

```
$ sudo ufw allow in on wg0 to any port 22
```

Say you want to allow connection for TCP port 3306 on lxdbr0 interface from 10.105.28.22, then add:

`$ sudo ufw allow in on lxdbr0 from 10.105.28.22 to any port 3306 proto tcp`

Let us add sub/net instead of single IP address:

`$ sudo ufw allow in on lxdbr0 from 10.105.28.0/24 to any port 3306 proto tcp`

Step 5 – Block and deny incoming connections/ports

Do you want to close ports and block certain IP addresses? The syntax is as follows to deny access. In other words, simply ignoring access to port 25:

`$ sudo ufw deny 25/tcp`

Make sure we deny all connections from an IP address called 203.5.1.43, enter:

`$ sudo ufw deny from 203.5.1.43`

Deny all connections from an IP/subnet called 103.13.42.13/29, enter:

`$ sudo ufw deny from 103.13.42.13/29`

Want to deny access to 1.1.1.2 (say bad guys IP) on port 22? Try:

`$ sudo ufw deny from 1.1.1.2 to any port 22 proto tcp`

Step 6 – Verify status of UFW

Use the status command as follows:

`$ sudo ufw status`

```
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
80/tcp                     ALLOW       Anywhere                # accept Apache
443/tcp                    ALLOW       Anywhere                # accept HTTPS connections
1194/udp                   ALLOW       Anywhere                # OpenVPN server
3000:4000/tcp              ALLOW       Anywhere
3000:4000/udp              ALLOW       Anywhere
Anywhere                   ALLOW       104.22.10.214
25/tcp                     ALLOW       104.22.11.213
222.222.222.222 25/tcp     ALLOW       104.22.11.213
Anywhere                   DENY        203.5.1.43
Anywhere                   DENY        103.13.42.8/29
22/tcp                     DENY        1.1.1.2
22/tcp (v6)                ALLOW       Anywhere (v6)
80/tcp (v6)                ALLOW       Anywhere (v6)           # accept Apache
443/tcp (v6)               ALLOW       Anywhere (v6)           # accept HTTPS connections
1194/udp (v6)              ALLOW       Anywhere (v6)           # OpenVPN server
3000:4000/tcp (v6)         ALLOW       Anywhere (v6)
```

```
3000:4000/udp (v6)        ALLOW       Anywhere (v6)
```

Want verbose outputs? Try:

```
$ sudo ufw status verbose
```

Ubuntu 20.04 LTS UFW delete rules

So far we learned how to add, deny, and list the firewall rules. It is time to delete unwanted rules. The syntax is as follows to list all of the current rules in a numbered list format:

```
$ sudo ufw status numbered
```

```
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 80/tcp                     ALLOW IN    Anywhere                   # accept Apache
[ 3] 443/tcp                    ALLOW IN    Anywhere                   # accept HTTPS connections
[ 4] 1194/udp                   ALLOW IN    Anywhere                   # OpenVPN server
[ 5] 3000:4000/tcp              ALLOW IN    Anywhere
[ 6] 3000:4000/udp              ALLOW IN    Anywhere
```

To delete 6th rule type the command:

```
$ sudo ufw delete 6
```

```
$ sudo ufw status numbered
```

Revision #2
Created 23 December 2023 14:58:20 by ColtM
Updated 7 August 2024 23:24:39 by ColtM