

## Manual Installation Instructions

### Manual Installation Instructions

This document and its actions has been superseded by the [Install \*nftfw\* from Debian package](#). Users are recommended to use that route to install and update *nftfw*.

For those of you who just want to follow a list of instructions without any verbiage, this document lists all the steps in the [Installing \*nftfw\*](#) document. There are links, shown as 'Explanation', to the Installation document.

### Basic package installations

#### *nftables*

(Explanation)

```
$ sudo apt install nftables
```

Optional: If this installs a version less than 0.9.3, then edit `/etc/apt/sources.d` and add

```
# backports
```

```
deb <YOUR SOURCE> buster-backports main contrib non-free
```

and then

```
$ sudo apt update
```

```
$ sudo apt upgrade
```

```
$ sudo apt -t buster-backports install nftables
```

Raspberry Pi OS doesn't support buster-backports at the time of writing.

#### Python

(Explanation)

```
$ sudo apt install python3-pip python3-setuptools python3-wheel
```

```
$ sudo apt install python3-prettytable
```

### Check on iptables

Check on the state of *iptables*, and set things up to use the *nftables* compatibility mode

(Explanation)

```
$ sudo iptables -V
```

```
iptables v1.8.2 (nf_tables)
```

If the output looks like this, then skip to 'Installing *nftfw*'. This is the most likely scenario.

If the word in brackets is 'legacy', do the following

```
$ sudo iptables-save > ipsaved
```

```
$ sudo ip6tables-save > ip6saved
```

```
$ sudo update-alternatives --config iptables
```

```
# select selection 0, /usr/sbin/iptables-nft, auto mode
```

```
$ sudo update-alternatives --config ip6tables
```

```
# select selection 0, /usr/sbin/iptables-nft, auto mode
```

Run the `sudo iptables -v` again, to check things have switched, and

```
$ sudo iptables-restore < ipsaved
$ sudo ip6tables-restore < ip6saved
$ sudo iptables-legacy -F
$ sudo ip6tables-legacy -F
```

### Installing *nftfw*

Get *nftfw* installation and install ([Explanation](#))

```
# Change to a suitable directory, perhaps
$ cd /usr/local/src
$ sudo apt install git
$ sudo git clone https://github.com/pcollinson/nftfw
```

Change into the *nftfw* directory you've just installed and:

```
$ sudo pip3 install .
Successfully installed nftfw-<version>
```

*pip3* may complain about being run as the superuser, it's safe to ignore that warning.

Install *nftfw* infrastructure, without any user interaction:

```
$ cp Autoinstall.default Autoinstall.conf
```

edit the *AUTO\_USER* line to the user you want to use own the files in *etc/nftfw*. The *Autoinstall.conf* file will be ignored by *git* so this script can be used to update any future releases.

Install *nftfw* infrastructure:

```
$ sudo sh Install.sh
```

If you've not created *Autoinstall.conf*, the *Install.sh* will ask some questions. Answers for default installation: - *Install under /usr/local?* yes - *See the files installed?* your choice - *Install?* yes - *User to replace root?* 'admin' for Symbiosis, 'symp1' for Symbl, 'return' for root on other systems - *Install Manual pages?* yes

### Disable cron and incron actions for Sympl or Symbiosis

On Symbiosis move */etc/cron.d/symbiosis-firewall* to a safe place. On Symbl move */etc/incron.d/symbiosis-firewall* to a safe place. On Sympl move */etc/cron.d/syml-firewall* to a safe place.

### Test that *nftfw* doesn't complain

```
$ sudo nftfw -x -v load
nftfw[15264]: Loading data from /usr/local/etc/nftfw
nftfw[15264]: Creating reference files in /usr/local/var/lib/nftfw/test.d
nftfw[15264]: Test files using nft command
nftfw[15264]: Testing nft rulesets from nftfw_init.nft
nftfw[15264]: Determine required installation
nftfw[15264]: No install needed
```

The number in the log is the process id, so will be different for you.

### Taking precautions if you have a live firewall

If you don't have a running *nftables* or *iptables* firewall, then [skip to 'Run a test...'](#).

If you DO carry on here ([Explanation](#))

If you have a running firewall, save its rules first:

```
$ sudo nftfwadm save
$ sudo nftfw -f -v load
```

Output should end with 'Install rules in ...' - wherever the *config.ini* file tells *nftfw* to store the *nftables.conf* file.

```
$ sudo nft list ruleset
```

will list the ruleset.

If you have a problem, revert to old rules:

```
$ sudo nftfwadm restore
```

if not

```
$ sudo nftfwadm clean
```

### Run a test if you don't have a live firewall

If you DON'T have a running *nftables* or *iptables* firewall ([Explanation](#)) If you DO then you've done this bit above.

```
$ sudo nftfw -f -v load
```

to test installation. Output should end with 'Install rules in ...' - wherever the *config.ini* file tells *nftfw* to store the *nftables.conf* file.

```
$ sudo nft list ruleset
```

will list the ruleset.

### Final steps

([Explanation](#))

Edit */usr/local/etc/nftfw/config.ini* to put the *nftables.conf* file in the right place

```
# Location of system nftables.conf
# Usually /etc/nftables.conf
nftables_conf = /etc/nftables.conf
```

run to write it there

```
$ sudo nftfw -f load
```

Tell *systemctl* to enable and start its *nftables* service.

```
$ sudo systemctl enable nftables
$ sudo systemctl start nftables
$ sudo systemctl status nftables
```

On a Symbiosis system -

```
$ cd /etc/network
# put into a safe place - in case you want to revert
$ sudo mv if-up.d/symbiosis-firewall ~/up-symbiosis-firewall
$ sudo mv if-down.d/symbiosis-firewall ~/down-symbiosis-firewall
```

On a Syml system -

```
$ cd /etc/network
# put into a safe place - in case you want to revert
$ sudo mv if-up.d/syml-firewall ~/up-syml-firewall
$ sudo mv if-down.d/syml-firewall ~/down-syml-firewall
```

This turns out to be an important step, rebooting without having this done results in a bad combination of two firewalls, because the *nftables* settings are loaded before the Symbiosis/Syml ones.

## Installing cron

### (Explanation)

Change into the *cronfiles* directory in the distribution.

```
$ cd cronfiles
# check that the paths used in cron-nftfw are correct for you
$ sudo cp cron-nftfw /etc/cron.d/nftfw
# cron wants the file to be writeable only by owner
$ sudo chmod g-w /etc/cron.d/nftfw
$ cd ..
```

## Active control directories

### (Explanation)

Make *nftfw* update the firewall when files in the control directories change. If you don't do this, then you will need to run

```
$ sudo nftfw -f load
```

when you make a change by hand.

Install *systemd* control files from *systemd* in the *nftfw* distribution:

```
$ cd systemd
# check nftfw.path and nftfw.service have correct paths
$ sudo cp nftfw.* /etc/systemd/system
$ cd ..
# start the path unit only
$ sudo systemctl enable nftfw.path
$ sudo systemctl start nftfw.path
$ sudo systemctl status nftfw.path
# DON'T start or enable nftfw.service
# it will be started when needed by nftfw.path
```

Stop *incron* if it's running and you no longer need it

```
$ sudo systemctl stop incron
$ sudo systemctl disable incron
```

Finally a tip that's hard to find: reload *systemd* if you change the *nftfw* files after installation and starting:

```
$ sudo systemctl daemon-reload
```

## Installing Geolocation

This will add country detection to *nftfw*s, which is optional but desirable. See the [document](#). If you plan on blocking addresses by country, then the Geolocation system from Maxmind can provide tools to generate lists of IP addresses in the correct format.

## Configuring the firewall

*nftfw* is distributed with no rules specified for outbound packets (the *nftfw\_init.nft* file has some builtin default rules. The set of inbound rules are aimed at permitting access to *ssh*, *http* and *https*, *ftp* and various email ports. The incoming *ftp* rules are designed to support *Pure FTP*. Firewall configuration is a matter of creating or deleting files in the various directories in */usr/local/etc/nftfw*. You probably need to change settings for your system. Scan through the [How do I.. or a User's Quick Guide](#) document for a quick start on setting up access for your needs.

## Sympl users: Update your mail system after installation

A repository that steps through the changes I make to the standard *exim4/dovecot* systems on Sympl to improve feedback and detection of bad IPs - see [Sympl mail system update](#).

## You Are There

Now look at:

- [Updating \*nftfw\*](#)
  - How to update *nftfw*.
- [Installing Geolocation](#)
  - How to install GeoIP2
- [Getting CIDR lists](#)
  - How to obtain blocking lists for countries
- [User's Guide to \*nftfw\*](#)
  - The full User guide, the first section explains how the system is controlled.
- [How do I.. or a User's Quick Guide](#)
  - Answers a bunch of questions about the system.
- [nftfw web site](#)
  - All documents are available on the *nftfw* web site.

## Acknowledgement

All of this is made possible by shamelessly borrowing ideas from Patrick Cherry who created the Symbiosis hosting package for Bytemark of which the firewall system is part.