WonderShaper – A Tool to Limit Network Bandwidth in Linux

by Aaron Kili | Published: October 25, 2018 | Last Updated: October 29, 2018

Wondershaper is a small bash script that enables you to limit the network bandwidth in Linux. It employs the **tc command** line program as the backend for configuring traffic control. It is a handy tool for controlling bandwidth on a Linux server.

It allows you to set the maximum download rate and/or maximum upload rate. In addition, it also allows you to clear the limits that you have set and can display the current status of an interface from the command line. Instead of using the CLI options, you can run it persistently as a service under **systemd**.

In this article, we will show how to install and use **wondershaper** for limiting network bandwidth on Linux systems.

How to Install Wondershaper in Linux Systems

First, start by installing **wondershaper** using your Linux distribution package manager from the default repertoires as shown.

```
$ sudo apt install wondershaper   [On Debian/Ubuntu]
$ sudo yum install wondershaper   [On CentOS/RHEL]
$ sudo dnf install wondershaper   [On Fedora 22+]
```

Alternatively, to pull and install the latest updates, you need to clone the GitHub repository of **wondershaper** to your system, move into the local repository and install it using the following commands. Note that you should have the git command line tool installed:

```
$ cd bin
$ git clone https://github.com/magnific0/wondershaper.git
$ cd wondershaper
$ sudo make install
```

Before you start using **wondershaper**, you should first of all check all network interfaces attached to your machine using ifconfig or ip command.

This will help you know the interface on which you want to shape bandwidth usage, for example the wireless interface **wlp1s0** which is active.

```
$ ifconfig
OR
$ ip addr
```

```
aaronkilik@tecmint ~/bin/wondershaper $ ifconfig
enp2s0    Link encap:Ethernet  HWaddr 18:67:b0:29:8c:fd
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2803 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2803 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:244282 (244.2 KB)  TX bytes:244282 (244.2 KB)

wlp1s0    Link encap:Ethernet  HWaddr 18:67:b0:5b:af:f4
          inet addr:192.168.43.31  Bcast:192.168.43.255  Mask:255.255.255.0
          inet6 addr: fe80::dd8c:3d40:8171:8472/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:27217 errors:0 dropped:0 overruns:0 frame:0
          TX packets:26598 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23088858 (23.0 MB)  TX bytes:4817783 (4.8 MB)

aaronkilik@tecmint ~/bin/wondershaper $
```

Check Network Interfaces in Linux

How to Use Wondershaper to Limit Network Bandwidth in Linux

To define the maximum download rate in **Kbps** for an interface, run the following command using the option `-a` (defines interface) and `-d` (defines Kbps) i.e the download rate will be set to **4Mbps**.

```
$ wondershaper -a wlp1s0 -d 4048
```

To set the maximum upload rate in **Kbps** for an interface, use the `-u` option as follows.

```
$ wondershaper -a wlp1s0 -u 1048
```

You can also set download and upload at once with a single command, for instance.

```
$ wondershaper -a wlp1s0 -d 4048 -u 1048
```

The `-s` option allows you to view the current status of an interface.

```
$ wondershaper -sa wlp1s0
```

```
aaronkilik@tecmint ~ $ sudo wondershaper  -sa wlp1s0
qdisc htb 1: root refcnt 5 r2q 10 default 20 direct_packets_stat 135 direct_qlen 1000
 Sent 16105 bytes 135 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
qdisc ingress ffff: parent ffff:fff1 ----------------
 Sent 20002 bytes 101 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
aaronkilik@tecmint ~ $
```

Check Network Interface Status

You can also use iPerf – network throughput tool to test the bandwidth reduction by **wondershaper**, for example.

You can clear the download or upload limits you have set for an interface using the `-c` flag.

```
$ wondershaper -ca wlp1s0
```

It is also possible to run **wondershaper** as a service, where you define the parameters for shaping bandwidth in a config file. This enables wondershaper to start at boot time and limit bandwidth usage at all times, when the system is on, as explained in the next section.

How to Run Wondershaper Persistently Under Systemd

Under this mode, you need to set the interface, upload and download rates in the wondershaper configuration file located at **/etc/conf.d/wondershaper**. You can open this file for editing using your favorite CLI editor as shown.

```
$ sudo vim /etc/conf.d/wondershaper
```

Define the necessary parameters as follows.

```
[wondershaper]
# Adapter
IFACE="wlp1s0"

# Download rate in Kbps
DSPEED="4048"

# Upload rate in Kbps
USPEED="512"
```

Save the file and close it.

Next, start the **wondershaper** service for the mean time, enable it to auto-start at system boot and view its status, using the **systemctl command**.

```
$ sudo systemctl start wondershaper
$ sudo systemctl enable wondershaper
$ sudo systemctl status wondershaper
```

In case you alter the values of the parameters in the config file, you need to restart the **wonderservice** for the changes to be effected.

```
$ sudo systemctl restart wondershaper
```

To stop the wondershaper service, use the following command.

```
$ sudo systemctl stop wondershaper
```

For more help, see the Wondershaper Github repository: https://github.com/magnific0/wondershaper

Wondershaper is a traffic shaper for limiting network bandwidth on Linux systems. Try it out and share your thoughts with us via the feedback form below. If you know of any similar tools out there, you can as well mention to us in the comments – we will be grateful.

Tags: commandline tools | linux bandwidth limit

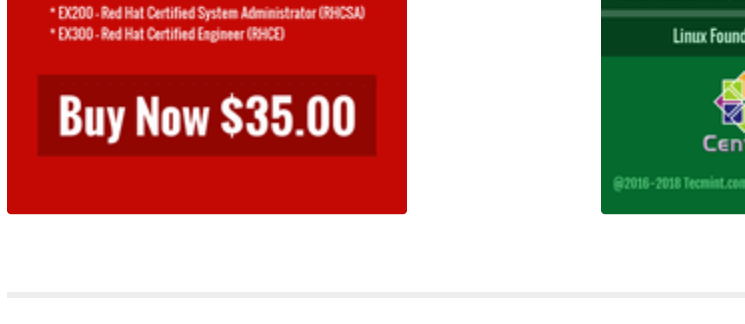
Aaron Kili · View all Posts

Aaron Kili is a Linux and F.O.S.S enthusiast, an upcoming Linux SysAdmin, web developer, and currently a content creator for TecMint who loves working with computers and strongly believes in sharing knowledge.

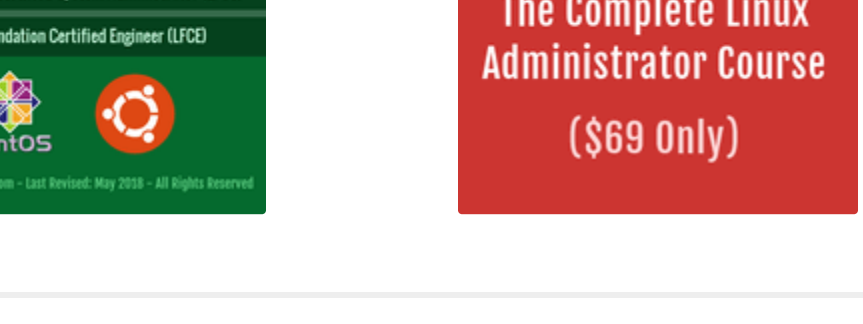
PREVIOUS STORY
How to Install Asterisk on CentOS/RHEL 8/7

NEXT STORY
How to Setup Central Logging Server with Rsyslog in Linux

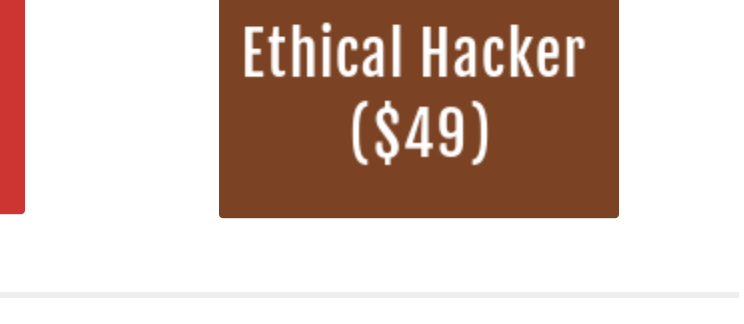
YOU MAY ALSO LIKE...



How to Test Network Throughput Using iperf3 Tool in Linux
28 SEP, 2018



How to Configure and Manage Network Connections Using 'nmcli' Tool
3 MAR, 2016



3 Ways to Set a Static IP Address in RHEL 8
4 DEC, 2018

5 RESPONSES

Comments 5 | Pingbacks 0

Morgan Collett · April 9, 2019 at 4:43 pm
On Ubuntu 18.04 the syntax is completely different:

wondershaper [interface] [downlink] [uplink]

Reply

Kevin Penrose · October 26, 2018 at 5:10 pm
The article is about wondershaper, but the install commands say wondersharper. Is that a typo?

Reply

> **Ravi Saive** · October 29, 2018 at 10:53 am
> @Kevin,
>
> Yes it is **wondershaper**, sorry for the typo, corrected in the article.
>
> Reply

Stef Bon · October 26, 2018 at 9:29 am
Real traffic shaping is something different than this.

Reply

> **Aaron Kili** · October 28, 2018 at 2:53 pm
> @Stef
>
> Yes, this is true. However, the main aim of wondershaper is to simply the whole process. You can use tc, wondershaper's backend or any other tool, for comprehensive bandwidth shaping.
>
> Reply

GOT SOMETHING TO SAY? JOIN THE DISCUSSION.

Comment

Name * Email *

Website

Save my name, email, and website in this browser for the next time I comment.

Notify me of followup comments via e-mail. You can also subscribe without commenting.

Post Comment

This site uses Akismet to reduce spam. Learn how your comment data is processed.