

HOWTO Manage Traffic Shaping with fwbuilder

Dr. Michael Schwartzkopff*

April 24, 2009

MultiNET Services GmbH, 85630 Grasbrunn, Munich, Germany

Apr, 24th 2009: rev. 0.0.1

Abstract

In this document I want to explain HOWTO use the fwbuilder GUI to manage the QoS traffic management toward the upstream provider. In my setup I use the debian lenny distributon. But the transfer to other distributions should be easy. So have fun using SSH while ssomebody else downloads the latest DVD image.

License: GNU FDL, see <http://www.gnu.org/copyleft/fdl.html>. Invariant section: Author.

1 Basics

The connection to the internet service provider (ISP) ist the most crucial point in internet access for a comapany. Besides beeing a Single-Point-of-Failure is is also the bottleneck that every communication packet has to pass. Quite often it happens that heavy downloads jam the traffic on that line while operators have the do some debugging of network equipment or remote servers. The answer times of the command line is fruststrating in such situations.

The solutions for such a problem in the bandwidth management resources on the edge towards the bottleneck. There are very good reasons to do the traffic management on the firewall. One reason or instance would be the bandwith management inside VPN tunnels that terminate also on the firewall.

The soltion I present in the HOWTO uses the features of Linux Advanced Routing and Traffic Control (lartc) that provides really advanced control over all parameters that bandwidth management would need. The traffic control setup is done via a separate script. This script is called during the setup of the interface that connects the firewall with the internet and set us all nescessary queues and bandwidth limits.

Some additions to the script are make to add filters the use firewall marks to distribute incomming packets to the right prio queue. fwbuilder offers a very nice GUI that I will use to classify packets and attach firewall marks according to usual IP criteria like source or destion IP address or port number.

2 QoS Script

If you search the internet for QoS solution for Linux will find a lot of various scripts. Some provide for hard bandwith limits with overbooking while other just offer prioritisation of specific traffic. Basically all scripts use the `tc` command to create qdiscs, classes and filters to classify and prioritise traffic.

In this HOWTO I will not go into the details of Linux traffic control. Please read the documentation of the according project¹ or the README of the script you are interested in. In the following I use

*misch@multinet.de

¹<http://lartc.org>

the `wondershaper` script². This script is quite old but still works like a charm to speed up interactive connections during a heavy download. The `wondershaper` is included in most of the distributions and quite simple to understand. `wondershaper` does provide only prioritisation for classified traffic but NO bandwidth limits. But the principles I show here can be applied to any other script also that also limits bandwidth for classified traffic.

`wondershaper` uses the classes 1:10 for high prio traffic, 1:20 for normal traffic and 1:30 for traffic that has to wait for the other two queues. `wondershaper` also defines some filters that classify traffic. By default `wondershaper` includes filters that sort interactive ssh traffic as well as ICMP's to 1:10 and eDonky traffic with a source port 4666 to 1:30. You can change these filter if they do not fit your needs. Additionally I add the following lines to the script:

```
tc filter add dev eth0 protocol ip prio 12 handle 10 fw flowid 1:10
tc filter add dev eth0 protocol ip prio 12 handle 30 fw flowid 1:30
```

Please take care the the number after the `prio` option is not used in some other filters. The first line installs a filter for the firewall mark 10 and throws all that traffic to queue 1:10. Please note that `eth0` is my external interface. Please change the command according to your needs. The second line throws traffic with firewall mark 30 into the slow queue. All other traffic is automatically routed to queue 1:20.

3 fwbuilder GUI

Now you have to add rules to your firewall policy that classify traffic and mark it according. For this task `fwbuilder` provides the `Mark` target. But before you can create your first rule you have to add custom `TagService` with the correct firewall mark. Right click on the `TagService` in the `User Objects` tree under `Services` and add a `New Service`. Call it `highPrio` and assign the code 10 to it. Create a second `TagService` called `lowPrio` with the code 30.

Now you can add a new rule to your rule base. I add the rule after the stealth rules. Just enter the criteria (source, destination or service), select only the input direction and select "Mark" as the Action. Double click the `Action` tab and `Drag&Drop` the New created `highPrio` `TagService` into the field provided. You can go on creating new traffic prioritisation rules as you like. Please see picture below to get some ideas.



Figure 1: Excerpt from my policy that shows the traffic classification rules. My PC is allowed to suf fast while all GamersPC are slowed down. The same fate happes to traffic the a bad service.

I add the traffic control rules after the stealth rules of my policy. But you also could add it on the top. Please take case that your firewall settings do NOT have the `Tag` and `Classify` actions terminating the firewall script. It also makes sense to make these rules stateless in the options of the rules itself and turn logging off. When you compile the policy you get basically the following rules:

```
$IPTABLES -t mangle -A PREROUTING -i + -s 192.168.1.35 -j MARK --set-mark 10
$IPTABLES -t mangle -A PREROUTING -i + -p tcp -m tcp --dport 666 -j MARK --set-mark 30
$IPTABLES -t mangle -A PREROUTING -i + -s 192.168.1.96/29 -j MARK --set-mark 30
```

When you install the policy now, the `iptables` part marks the packages and the traffic control part of the kernel send out the prioritized packages first.

Have fun shaping your traffic!

Please mail my success stories, bugs or other remarks to misch@multinet.de

²<http://lartc.org/wondershaper>