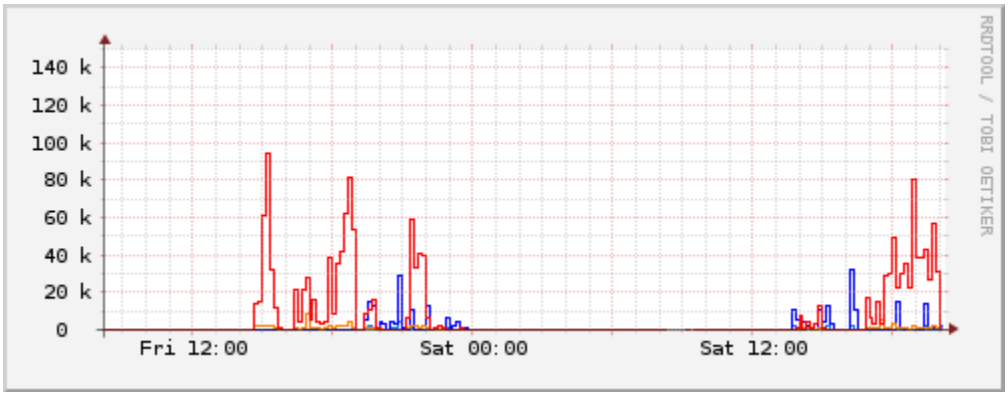
NETWORK STATISTICS WITH IPTABLES AND RRDTOOL

Netfilter is a powerful tool when it comes to select traffic on a Linux router.
When you set up a chain of rules with iptables, you are also get set of traffic counters with each rule you set, which can be used to see how many times the rule have matched.
If you place a chain of rules without any jump, the packet get counted and goes forward the chain, so that you can write a set of rules just to get some statistic of selected pattern of traffic in your network.
In this post I'll show how to write some simple rule, get the data on a rrdtool database and plot a traffic graphic out of it.



Couter Chains

On my setup, I wanted to keep track of traffic for two internal hosts, with a data for inbound and outbound traffic. To accomplish that, the firewall script have been modified to include a new "stat" chain, called from the FORWARD one, with the following rules:

```
iptables -N stats

iptables -A stats -s 192.168.0.3 -o $WAN -j RETURN

iptables -A stats -d 192.168.0.3 -i $WAN -j RETURN

iptables -A stats -s 192.168.0.4 -o $WAN -j RETURN

iptables -A stats -d 192.168.0.4 -i $WAN -j RETURN

iptables -A stats -o $WAN -j RETURN

iptables -A stats -i $WAN -j RETURN

iptables -A FORWARD -j stats
```

The last rules matches all the traffic not captured by others.

Database Creation

The round-robin database have to be created with a dataset for each host to track. Datasets are of COUNTER type, as the count keep growing each reading, check that out with the “iptables -L -v” command.

That’s the command use to create the database:

```
#!/bin/sh

DB=/root/traffic.rrd

# 5 minutes points, 48 hours data
# 30 minutes points, 25 days data
# 2 hours points, 2 months data
# 24 hours points, 2 years data

rrdtool create $DB \
  DS:out_3:COUNTER:600:U:U  \
  DS:in_3:COUNTER:600:U:U  \
  DS:out_4:COUNTER:600:U:U  \
  DS:in_4:COUNTER:600:U:U  \
  DS:out_other:COUNTER:600:U:U  \
  DS:in_other:COUNTER:600:U:U  \
  RRA:AVERAGE:0.5:1:576   \
  RRA:AVERAGE:0.5:6:720   \
  RRA:AVERAGE:0.5:24:720  \
  RRA:AVERAGE:0.5:288:730

```

Database Update

The database is updated with a small script called every 5 minutes with a cron job.

The script uses the iptables-save and rrdupdate utility with some glue logic to get the counters to a CSV-like line.

That’s the script:

```
#!/bin/sh

DB=/root/traffic.rrd
RRDUPDATE=rrdupdate
IPTABLES_SAVE=/sbin/iptables-save

data=$( $IPTABLES_SAVE  -c | grep -- '-A stats' | \
        sed -r 's/\[([0-9]*):([0-9]*).*/:\2/' | \
        xargs echo | sed 's/ //g' )

$RRDUPDATE $DB N$data

```

Just be sure that the fields in the database matches the iptables rules.

Plotting

The data is plotted into a png file for web visualization using another script.

This is an example of how to plot a 2-day graphic for the two tracked hosts:

```
#!/bin/sh

DB=/root/traffic.rrd
```

```
RRDTOOL=rrdtool
OUT=/opt/lighttpd/www/htdocs/rrd/traffic.png
OPTS="-w 700 -h 200"

$RRDTOOL graph $OUT $OPTS --start -2d \
  DEF:out_3=$DB:out_3:AVERAGE \
  DEF:in_3=$DB:in_3:AVERAGE \
  DEF:out_4=$DB:out_4:AVERAGE \
  DEF:in_4=$DB:in_4:AVERAGE \
  DEF:in_other=$DB:in_other:AVERAGE \
  DEF:out_other=$DB:out_other:AVERAGE \
  LINE1:out_3#0080ff:192.168.0.3_OUT \
  LINE1:in_3#0000ff:192.168.0.3_IN \
  LINE1:out_4#ff8000:192.168.0.4_OUT \
  LINE1:in_4#ff0000:192.168.0.4_IN
```

That’s it! Once you learned the basics, you will be able to log every kind of network-related data very quickly.

Source : http://fabiobaltieri.com/2012/01/14/network-statistics-with-iptables-and-rrdtool/