# Filtering traffic based on thousands of IPs efficiently

Posted by uljanow on Wed 4 Jul 2007 at 11:11

Trying to insert 70.000 rules in iptables on a recent machine takes about an hour and going through these rules for each packet is even more of a burden. But iptables can send packets to userspace to be handled there. This article describes how to filter network traffic based on thousands of IPs with a new tool called nfqueue efficiently.

## Prerequisites

`nfqueue` requires a 2.6.14 kernel or later with the option `CONFIG_NETFILTER_XT_TARGET_NFQUEUE` enabled (module or build-in). On a standard Debian installation (Etch) the additional packages `libnetfilter-queue1` and

## Installation

Install prerequisites

```
aptitude install libnetfilter-queue1 libnfnetlink1
```

Get the Debian nfqueue package and install it

```
wget http://nfqueue.sf.net/debian/nfqueue_0.11-1_i386.deb
dpkg -i nfqueue_0.11-1_i386.deb
```

## Overview

IP ranges are specified in p2p, dat, csv text files or in nfq binary format.

A p2p format looks like this:

```
foo : 127.0.0.1 - 127.0.0.2
```

A dat file looks like this:

```
127.0.0.1, 127.0.0.2, <0-255>, foo
```

(Values less than 127 are dropped.)

For available lists take a look at `/usr/share/doc/nfqueue/README.lists`.

Sending packets to userspace is done by using the NFQUEUE target. E.g:

```
iptables -I INPUT -p all -j NFQUEUE
```

From userspace there are basically 3 things one can do with packets.

- Accept
- Drop
- Repeat

Repeating Packets sends them back to the chain (IN-, OUTPUT or FORWARD) they came from. Since this could lead to endless loops marking packets is possible. The other options Accept and Drop are terminating targets. See "`man 1 nfqueue`" for more details.

# Example - Blocking whole Countries

Get the csv file from webhosting.info

```
wgethttp://ip-to-country.webhosting.info/downloads/ip-to-country.csv.zip
```

Let's assume we want to block the whole US. First we put the ip ranges of the USA into a nfq binary to make loading faster.

```
unzip -c ip-to-country.csv.zip | grep -i usa | \
  nfqueue -t repeat -o usa.nfq -
```

The easy way now would be to use the `/usr/share/doc/nfqueue/nfqueue.sh` script which I will explain later. Updating these values is all that needs to be done:

```
INPUT_FILES=/path/to/usa.nfq
OUTPUT_FILES=/path/to/usa.nfq
```

Run:

```
nfqueue.sh start
nfqueue.sh stop
nfqueue.sh status
```

# nfqueue.sh Script

## What does the script do exactly

Packets are filtered in the INPUT and OUTPUT chain. For each new connection (both directions) nfqueue looks if the IP is specified in `usa.nfq`. If the IP is found then it gets marked and repeated so that it can be rejected by iptables. If the IP is not found nfqueue marks the packet to avoid looping forever and sends it back (repeat again) to be handled by the rest of the iptables configuration.

The script only rejects packets from clients specified in files and the rest is handled by your iptables configuration.

Note that the script rejects packet properly instead of just dropping.

# Notes

There is also an `ipset` tool from netfiler.org which requires kernel-patching and some scripting to parse the IPs from files and insert them.

---