# netfilter/iptables FAQ

Harald Welte <coreteam@netfilter.org> Version $Revision: 6856 $, $Date: 2007-06-04 13:38:34 +0200 (lun, 04 jun 2007) $

This document contains the Frequently Asked Questions as encountered on the netfilter mailing list. Comments / additions / clarifications are appreciated and should be directed to the FAQ maintainer.

## Contents

# 1 General Questions

This section covers general netfilter (and non-netfilter) related questions we've encountered frequently on the mailing list.

## 1.1 Where can I get netfilter/iptables?

Netfilter and IPtables are integrated in the Linux kernel since 2.4.x series. Please obtain a recent kernel from `http://www.kernel.org/` or one of its mirrors.

The userspace tools 'iptables' and 'ip6tables' are available at the netfilter homepage on one of the mirrors at

`http://www.netfilter.org/` or

`http://www.iptables.org/` .

## 1.2 Is there a backport of netfilter to Linux 2.2?

No, there currently is none. But if anybody wants to start, it shouldn't be too difficult because of the clean interface to the network stack.

Please inform us about any work in this area.

## 1.3 Is there an ICQ conntrack/NAT helper module?

If you are used to masquerading on a Linux 2.2 box, you always used the ip_masq_icq module in order to get direct client-to-client ICQ working.

Nobody re-implemented this module for netfilter, because the ICQ protocol is too ugly :) But I guess it's just a matter of time until one is available.

Rusty once pointed out that only modules for protocols with at least one free client and one free server are going to get integrated into the main netfilter distribution. As for ICQ, there are only free clients, so it doesn't match this criteria. (free as in freedom, not in free beer, i.e. RMS' definition)

## 1.4 Where did the ip_masq_vdolive / ip_masq_quake / ... modules go?

Some of them are not required, and some haven't been ported to netfilter yet. Netfilter does full connection tracking even for UDP, and has a policy of trying to disturb the packets at little as possible, so sometimes things 'just work'.

## 1.5 What is this patch-o-matic all about and how do I use it?

The 2.4.x kernels is a stable release, so we can't just submit our current development into the mainstream kernel. All our code is developed and tested in netfilter patch-o-matic first. If you want to use any of the bleeding-edge netfilter functions, you may have to apply one or more of the patches from patch-o-matic. You can find patch-o-matic in the latest iptables package (or of course SVN), to be downloaded from the netfilter homepage.

patch-o-matic now has three different options:

- make pending-patches
- make most-of-pom
- make patch-o-matic

The first one is just to make sure all important bugfixes (which have been submitted to the kernel maintainers anyway) are applied to your kernel. The second 'most-of-pom' additionally prompts you for all new features which can be applied without conflict. The third option 'patch-o-matic' is for real experts who want to see all the patches - but be aware, they might conflict which each other.

patch-o-matic has a neat user interface. Just enter

```
make most-of-pom (or pending-patches or patch-o-matic, see above)
```

or, if your kernel tree is not in `/usr/src/linux` then use

```
make KERNEL_DIR={your-kernel-dir} most-of-pom
```

in the top directory of the iptables-package. patch-o-matic checks for each of the patches if it would apply against the kernel source you have installed. If a patch would apply, you will see a little prompt, where you can ask for more information about this patch, apply the patch, skip to the next one, ...

For more information about patch-o-matic, please see the netfilter extensions HOWTO, to be found at

http://www.netfilter.org/documentation/index.html#HOWTO

### 1.6 Where can I find ipnatctl and more information about it?

ipnatctl was used to set up your NAT rules from userspace in a very early development revision of netfilter during the 2.3.x kernels. It is no longer needed, thus no longer available. All of its functionality is provided by iptables itself. Have a look at the NAT HOWTO on the Netfilter homepage.

### 1.7 Can iptables/ip6tables do IPv6 NAT?

No, the NAT core does not support any kind of IPv6 or IPv6/IPv4 NAT.

### 1.8 Are there any plans to support SIP?

An implementation of conntrack and NAT for the SIP (Session Initiation Protocol) has been in the patch-o-matic for some time now implemented by Christian Hentschel. The implementation is available since Linux Kernel 2.6.18.

### 1.9 Does netfilter/iptables support failover/HA?

If you are thinking about a full failover, while all the state information is preserved, you have to use **conntrack-tools** from `http://people.netfilter.org/pablo/conntrack-tools/`.

## 2 Problems during the build process

### 2.1 I cannot compile iptables-1.1.1 with kernel >= 2.4.0-test4

This is a known issue. The mechanism for the detection which patches to apply is broken. Try using "make build" instead of "make".

Better solution: Upgrade to iptables-1.1.2 or later

### 2.2 I cannot compile iptables 1.1.0 with recent kernels (>= 2.3.99-pre8)

Internal structures in iptables have changed. Upgrade to iptables >= 1.1.1

### 2.3 Some patch-o-matic patches from iptables >= 1.2.1a don't work with kernel >= 2.4.4

Please use a recent iptables release.

### 2.4 ipt_BALANCE, ip_nat_ftp, ip_nat_irc, ipt_SAME, ipt_NETMAP don't compile

Most likely you are experiencing compile problems with a function called `ip_nat_setup_info`.

If you are using iptables <= 1.2.2, you **NEED** to apply the 'dropped-table' and 'ftp-fixes' patches.

If you are using iptables > 1.2.2 or recent SVN, please **don't** apply the 'dropped-table', as it is incompatible with BALANCE, NETMAP, irc-nat, SAME and talk-nat.

### 2.5 I'm using Alan Cox' 2.4.x-acXX series kernel and I experience problems

The netfilter core team bases development on Linus' kernel tree, so using the -ac series is on your own risk.

### 2.6 ERROR: Invalid option KERNEL_DIR=/usr/src/linux-2.4.19

I bet you are trying to run something like :

```
# ./runme pending KERNEL_DIR=/usr/src/linux-2.4.19
```

But bash/sh is not like make, and a variable cannot be passed as a parameter. You have to set the variable before you run the runme script :

```
# KERNEL_DIR=/usr/src/linux-2.4.19 ./runme pending
```

## 3 Problems at runtime

### 3.1 NAT: X dropping untracked packet Y Z aaa.aaa.aaa.aaa -> 224.bbb.bbb.bbb

This message is printed by the NAT code, because multicast packets are hitting the NAT table, and connection tracking doesn't handle multicast packets right now. In case you have no idea what multicast is, or don't need it at all, use:

```
iptables -t mangle -I PREROUTING -j DROP -d 224.0.0.0/8
```

### 3.2 NAT: X dropping untracked packet Y Z aaa.aaa.aaa.aaa -> bbb.bbb.bbb.bbb

My syslog or my console shows the message:

```
NAT: X dropping untracked packet Y Z aaa.aaa.aaa.aaa -> bbb.bbb.bbb.bbb
```

This message is printed by the NAT code. It drops packets, because in order to do NAT it has to have valid connection tracking information. This message is printed for all packets for which connection tracking was unable to determine conntrack information.

Possible reasons are:

- maximum limit of entries in the conntrack database reached
- couldn't determine inverted tuple (multicast, broadcast)
- kmem_cache_alloc fails (out of memory)
- reply on unconfirmed connection
- multicast packet (please see previous question)
- icmp packet too short
- icmp is fragmented

- icmp checksum wrong

If you want to have a more detailed logging of these packets (i.e. if you suspect it are remote probe / scanning packets), use the following rule:

```
iptables -t mangle -A PREROUTING -j LOG -m state --state INVALID
```

And yes, you have to put the rule in the mangle table, because the packets get dropped by the NAT code before they reach the filter table.

## 3.3 ip_conntrack: max number of expected connections N of M reached for aaa.aaa.aaa.aaa -> bbb.bbb.bbb.bbb

My sylog or console regularly shows messages like:

```
ip_conntrack: max number of expected connections N of M reached for aaa.aaa.aaa.aaa -> bbb.bbb.bbb.bbb
```

This is normally nothing to worry about, especially if `N` and `M` are `1`, and the message is followed by `, reusing`. In particular versions of the linux kernel (2.4.19<=x<=2.4.21-pre3), this message was printed for FTP - And in fact this can happen during normal FTP operation.

With upcoming kernel versions (>=2.4.21-pre4) this message is no longer printed to not confuse the user. If you still receive messages like this, contact the http://lists.netfilter.org/mailman/listinfo/netfilter-devel mailinglist.

## 3.4 I'm unable to use netfilter in combination with the Linux bridging code

So you want to build a completely transparent firewall? Great idea! In current kernel there is no need to patch anything. You can find more at

http://linux-net.osdl.org/index.php/Bridge .

## 3.5 The IRC module is unable to handle DCC RESUME

Well, that's half the truth. Only the NAT module is unable to handle them. If you just use firewalling without NAT it should work fine. Patches welcome.

## 3.6 How does SNAT to multiple addresses work?

Netfilter tries to mangle as little as possible. So if we have a freshly- rebooted machine, and somebody behind the SNAT box opens a connection with local port 1234, the netfilter box only mangles the IP address and the port stays the same.

As soon as somebody else opens another connection with the same source port, netfilter would have to mangle IP and port if it only has a single IP for SNAT.

But if there are more than one available, it **again** only has to mangle the IP part.

## 3.7 ip_conntrack: maximum limit of XXX entries exceeded

If you notice the following message in syslog, it looks like the conntrack database doesn't have enough entries for your environment. Connection tracking by default handles up to a certain number of simultaneous connections. This number is dependent on you system's maximum memory size (at 64MB: 4096, 128MB: 8192, ...).

You can easily increase the number of maximal tracked connections, but be aware that each tracked connection eats about 350 bytes of non-swappable kernel memory!

To increase this limit to e.g. 8192, type:

```
echo "8192" > /proc/sys/net/ipv4/ip_conntrack_max
```

To optimize performance, please also raise the number of hash buckets by using the `hashsize` module loadtime parameter of the `ip_conntrack.o` module. Please note that due to the nature of the current hashing algorithm, an even hash bucket count (and esp. values of the power of two) are a bad choice.

Example (with 1023 buckets):

```
modprobe ip_conntrack hashsize=1023
```

## 3.8 How do I list all tracked / masqueraded connections, similar to 'ipchains -L -M' in 2.2.x ?

There is a file in the proc-filesystem, which is called `/proc/net/ip_conntrack`. You can print the output of this file using

```
cat /proc/net/ip_conntrack
```

## 3.9 How do I list all available IP tables?

All available IP tables are listed with

```
cat /proc/net/ip_tables_names
```

## 3.10 iptables-save / iptables-restore from iptables-1.2 segfaults

Known Bug. Please update to latest SVN or use iptables >= 1.2.1 as soon as it is available.

## 3.11 iptables -L takes a very long time to display the rules

This is because iptables does a DNS lookup for each IP address. As each rule consists out of two addresses, the worst case is two DNS lookups per rule.

The problem is, if you use private IP addresses (like 10.x.x.x or 192.168.x.x), DNS is unable to resolve a hostname and times out. The sum of all these timeouts may be _very_ long, depending on your ruleset.

Please use the -n (numeric) option for iptables in order to prevent it from making reverse DNS lookups.

## 3.12 How do I stop the LOG target from logging to my console?

You have to configure your syslogd and/or klogd appropriately: The LOG target logs to facility kern at priority warning (4). See the syslogd.conf manpage to learn more about facilities and priorities.

By default, all kernel messages at priority more severe than debug (7) are sent to the console. If you raise that to 4, instead of 7, you will make the LOG messages no longer appear on the console.

Be aware that this might also suppress other important messages from appearing on the console (does not affect syslog).

## 3.13 How do I build a transparent proxy using squid and iptables?

First, of course, you need a suitable DNAT or REDIRECT rule. Use REDIRECT only if squid is running on the NAT box itself. Example:

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to 192.168.22.33:3128
```

After that, you have to configure squid appropriately. We can only give short notes here, please refer to the squid documentation for further details.

The squid.conf for Squid 2.3 needs to be something like the following:

```
http_port 3128
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy  on
httpd_accel_uses_host_header on
```

Squid 2.4 needs an **additional** line added:

```
httpd_accel_single_host off
```

## 3.14 How do I use the LOG target / How can i LOG and DROP?

The LOG target is what we call a "non-terminating target", i.e. it doesn't terminate the packets rule traversal. If you use the LOG target, the packet will be logged, and rule traversal continues at the next rule.

So how do I log and drop at the same time? Nothing easier than that, you create a custom chain which contains the two rules:

```
iptables -N logdrop
iptables -A logdrop -j LOG
iptables -A logdrop -j DROP
```

Now everytime you want to log and drop a packet, you can easily use a "-j logdrop".

## 3.15 How do I stop worm XYZ with netfilter ?

The short answer is you cannot do that properly with netfilter. Most of the worms are using a legitimate high level protocol (i.e. HTTP, SMTP(i.e VB script attached in email), or any exploit of a vulnerability found in the daemon handling the protocol). By high level protocol, we mean above TCP/IP. As iptables does not understand these high level protocols, it's almost impossible to filter part of them out properly. For that you need application filtering proxies.

Please do not use the string match from patch-o-matic instead of application proxy filtering. It would be defeated anytime by fragmented packets (i.e. an HTTP request split on two TCP packets), by IDS evasion techniques, etc... you have been warned! The string match is useful but for different purposes.

### 3.16 kernel logs: Out of window data xxx

You use the tcp-window-tracking patch from patch-o-matic, which code keeps track the acceptable packets for the allowed TCP streams according to the sequence/acknowledgement numbers, segment sizes, etc of the packets. When it detects that one of the packets is not acceptable (out of the window), it marks it as INVALID and prints the message above.

Newer versions logs the packet and exactly what condition failed for it:

- ACK is under the lower bound (possibly overly delayed ACK)
- ACK is over the upper bound (ACKed data has never seen yet)
- SEQ is under the lower bound (retransmitted already ACKed data)
- SEQ is over the upper bound (over the window of the receiver)

Also, in newer versions the logging can completely be suppressed via sysctl

```
echo 0 > /proc/sys/net/ipv4/netfilter/ip_ct_tcp_log_out_of_window
```

### 3.17 Why does the connection tracking system track UNREPLIED connections with a high timeout?

First, check if you are running a 2.4.20 kernel. If yes, please apply immediately the patch from https://bugzilla.netfilter.org/cgi-bin/bugzilla/showattachment.cgi?attach_id=8 ! The 2.4.20 kernel did contain a bug resulting in lots of bogus UNREPLIED conntrack entries (see https://bugzilla.netfilter.org/cgi-bin/bugzilla/show_bug.cgi?id=56 ).

So you have read /proc/net/ip_conntrack and found UNREPLIED entries with a very high timer (can be up to five days) and are wondering why we want to waste conntrack entries with UNREPLIED entries (which are obviously not connections)?

The answer is easy: UNREPLIED entries are temporary entries, i.e. as soon as we run out of connection tracking entries (we reach /proc/sys/net/ipv4/ip_conntrack_max), we delete old UNREPLIED entries. In other words: instead of having empty conntrack entries, we'd rather keep some maybe useful information in them until we really need them.

### 3.18 Why isn't the 'iptables -C' (–check) option implemented?

Well, first of all, we're lazy ;). To be honest, implementing a check option is almost impossible as soon as you start to do stateful firewalling. Traditional stateless firewalling bases it's decision just on information present in the packets header. But with connection tracking (and '-m state' based rules), the outcome of the filtering decision depends on header+payload, as well as header+payload of previous packets within this connection.

### 3.19 Why can't i use the REJECT target in PREROUTING chains?

The REJECT target is for filtering. The filter table has got the INPUT/OUPUT/FORWARD chains, therfore the REJECT target can only be used in those chains (and sub-chains, naturally).

Netfilter users do not filter packets in the nat or mangle tables.

### 3.20 'iptables: Invalid argument' after kernel update (nat table)

You have just upgraded your kernel and suddenly some of the commands (especially in the 'nat' table), and you experience something like:

```
# iptables -A POSTROUTING -t nat -o ppp0 -j MASQUERADE
iptables: Invalid argument
```

This happens when the structure size between kernel and userspace changes. You will need to recompile the iptables userspace program using the include files of your new kernel. This only happens if you (or the vendor of your kernel) has applied some patches either only to the old or only to the new kernel. It is not supposed to happen between vanilla kernel.org kernels. If it does, please inform the netfilter-devel mailinglist.

## 4 Questions about netfilter development

### 4.1 I don't understand how to use the QUEUE target from userspace

A library called libipq is provided for userspace packet handling. There is now documentation for this in the form of man pages. You need to build and install the iptables development components:

```
make install-devel
```

then see libipq(3).

You may also be interested in the Perl bindings for libipq, Perlipq at:

http://www.intercode.com.au/jmorris/perlipq/ . The binding itself is an example of using the library.

Other code examples include:

- testsuite/tools/intercept.c from netfilter SVN
- ipqmpd (see http://www.gnumonks.org/projects/ )
- nfqtest, part of netfilter-tools (see http://www.gnumonks.org/projects/ )
- Jerome Etienne's WAN simulator (see http://www.off.net/~jme/ )

### 4.2 My libipq application says "Failed to received netlink message: No buffer space available"

This means that the kernel-side Netlink socket buffer ran out of space; the userspace application is not able to handle the amount of data being delivered from the kernel.

*Is it possible to make those kernel buffers bigger so that I don't run into this problem?*

Yes, these are standard Netlink sockets, and you can tune their receive buffer sizes via /proc/sys/net/core, sysctl, or use the SO_RCVBUF socket option on the file descriptor.

You can also try ensuring that your application is reading any received data as quickly as possible. If you don't need the entire packet, try copying less data to userspace (see ipq_set_mode(3)).

## 4.3 I want to contribute some code, but have no idea what to do

The netfilter core-team keeps a TODO list where it lists all the desired changes / new features. You can retrieve this list via anonymous SVN, instructions are on the netfilter Homepage. Alternatively you can also go to http://svn.netfilter.org/cgi-bin/cvsweb/netfilter/TODO/ using CVSweb.

## 4.4 I've fixed a bug or written an extension. How do I contribute it?

If you want to publish it, please send it to the netfilter-devel mailinglist. Subscription instructions are at http://lists.netfilter.org/mailman/listinfo/netfilter-devel/ .

The correct way of sending a patch is the following :

- Subject starting with **[PATCH]**
- Included straight in the body of the message, not MIME'd.
- a svn-checkin/Changelog entry outside the diff.
- 'diff -u old new' form, from outside root directory (ie. can be applied with -p1 when sitting in the untarred dir.

If you wrote a new extension, or added some new options to an old extension, it's usually a good idea to also update the netfilter-extension-HOWTO to include that new extension/functionality description. Additionally, it will draw more users to your extension, and will allow you to get more feedback in general.

## 4.5 Is there an C/C++ API for adding/removing rules?

The answer unfortunately is: No.

Now you might think 'but what about libiptc?'. As has been pointed out numerous times on the mailinglist(s), libiptc was _NEVER_ meant to be used as a public interface. We don't guarantee a stable interface, and it is planned to remove it in the next incarnation of linux packet filtering. libiptc is way too low-layer to be used reasonably anyway.

We are well aware that there is a fundamental lack for such an API, and we are working on improving that situation. Until then, it is recommended to either use system() or open a pipe into stdin of iptables-restore. The latter will give you a way better performance.