NAME

pfctl **-** control the packet filter (PF) device

SYNOPSIS

pfctl [**-AdeghMmNnOPqRrvz**] [**-a** *anchor*] [**-D** *macro*= *value*] [**-F** *modifier*] [**-f** *file*] [**-i** *interface*]
[**-K** *host* | *network*] [**-k** *host* | *network* | *label* | *id* | *gateway*] [**-o** *level*] [**-p** *device*] [**-s** *modifier*]
[**-t** *table* **-T** *command* [*address* ...]] [**-x** *level*]

DESCRIPTION

The **pfctl** utility communicates with the packet filter device using the ioctl interface described in pf(4). It allows ruleset and parameter configuration and retrieval of status information from the packet filter.

Packet filtering restricts the types of packets that pass through network interfaces entering or leaving the host based on filter rules as described in pf.conf(5). The packet filter can also replace addresses and ports of packets. Replacing source addresses and ports of outgoing packets is called NAT (Network Address Translation) and is used to connect an internal network (usually reserved address space) to an external one (the Internet) by making all connections to external hosts appear to come from the gateway. Replacing destination addresses and ports of incoming packets is used to redirect connections to different hosts and/or ports. A combination of both translations, bidirectional NAT, is also supported. Translation rules are described in pf.conf(5).

When the variable *pf* is set to YES in rc.conf(5), the rule file specified with the variable *pf_rules* is loaded automatically by the rc(8) scripts and the packet filter is enabled.

The packet filter does not itself forward packets between interfaces. Forwarding can be enabled by setting the sysctl(8) variables *net.inet.ip.forwarding* and/or *net.inet6.ip6.forwarding* to 1. Set them permanently in sysctl.conf(5).

The **pfctl** utility provides several commands. The options are as follows:

-A Load only the queue rules present in the rule file. Other rules and options are ignored.

-a *anchor*

Apply flags **-f**, **-F**, and **-s** only to the rules in the specified *anchor*. In addition to the main ruleset, **pfctl** can load and manipulate additional rulesets by name, called anchors. The main ruleset is the default anchor.

Anchors are referenced by name and may be nested, with the various components of the anchor path separated by '/' characters, similar to how file system hierarchies are laid out. The last component of the anchor path is where ruleset operations are performed.

Evaluation of *anchor* rules from the main ruleset is described in pf.conf(5).

For example, the following will show all filter rules (see the **-s** flag below) inside the anchor "authpf/smith(1234)", which would have been created for user "smith" by authpf(8), PID 1234:

```
# pfctl -a "authpf/smith(1234)" -s rules
```

Private tables can also be put inside anchors, either by having table statements in the pf.conf(5) file that is loaded in the anchor, or by using regular table commands, as in:

```
# pfctl -a foo/bar -t mytable -T add 1.2.3.4 5.6.7.8
```

When a rule referring to a table is loaded in an anchor, the rule will use the private table if one is defined, and then fall back to the table defined in the main ruleset, if there is one. This is similar to C rules for variable scope. It is possible to create distinct tables with the same name in the global ruleset and in an anchor, but this is often bad design and a warning will be issued in that case.

By default, recursive inline printing of anchors applies only to unnamed anchors specified inline in the ruleset. If the anchor name is terminated with a ‘*’ character, the **-s** flag will recursively print all anchors in a brace delimited block. For example the following will print the "authpf" ruleset recursively:

```
# pfctl -a ’authpf/*’ -sr
```

To print the main ruleset recursively, specify only ‘*’ as the anchor name:

```
# pfctl -a ’*’ -sr
```

-D *macro*=*value*

Define *macro* to be set to *value* on the command line. Overrides the definition of *macro* in the ruleset.

-d Disable the packet filter.

-e Enable the packet filter.

-F *modifier*

Flush the filter parameters specified by *modifier* (may be abbreviated):

-F nat Flush the NAT rules.
-F queue Flush the queue rules.
-F ethernet Flush the Ethernet filter rules.
-F rules Flush the filter rules.
-F states Flush the state table (NAT and filter).
-F Sources Flush the source tracking table.
-F info Flush the filter information (statistics that are not bound to rules).
-F Tables Flush the tables.
-F osfp Flush the passive operating system fingerprints.
-F all Flush all of the above.

-f *file* Load the rules contained in *file*. This *file* may contain macros, tables, options, and normalization, queueing, translation, and filtering rules. With the exception of macros and tables, the statements must appear in that order.

-g Include output helpful for debugging.

-h Help.

-i *interface*
Restrict the operation to the given *interface*.

-K *host* | *network*
Kill all of the source tracking entries originating from the specified *host* or *network*. A second **-K** *host* or **-K** *network* option may be specified, which will kill all the source tracking entries from the first host/network to the second.

-k *host* | *network* | *label* | *id* | *gateway*
Kill all of the state entries matching the specified *host*, *network*, *label*, *id*, or *gateway*.

For example, to kill all of the state entries originating from "host":

```
# pfctl -k host
```

A second **-k** *host* or **-k** *network* option may be specified, which will kill all the state entries from the first host/network to the second. To kill all of the state entries from "host1" to "host2":

```
# pfctl -k host1 -k host2
```

To kill all states originating from 192.168.1.0/24 to 172.16.0.0/16:

```
      # pfctl -k 192.168.1.0/24 -k 172.16.0.0/16
```

A network prefix length of 0 can be used as a wildcard. To kill all states with the target "host2":

```
      # pfctl -k 0.0.0.0/0 -k host2
```

It is also possible to kill states by rule label or state ID. In this mode the first **-k** argument is used to specify the type of the second argument. The following command would kill all states that have been created from rules carrying the label "foobar":

```
      # pfctl -k label -k foobar
```

To kill one specific state by its unique state ID (as shown by pfctl -s state -vv), use the *id* modifier and as a second argument the state ID and optional creator ID. To kill a state with ID 4823e84500000003 use:

```
      # pfctl -k id -k 4823e84500000003
```

To kill a state with ID 4823e84500000018 created from a backup firewall with hostid 00000002 use:

```
      # pfctl -k id -k 4823e84500000018/2
```

It is also possible to kill states created from a rule with the route-to/reply-to parameter set to route the connection through a particular gateway. Note that rules routing via the default routing table (not via a route-to rule) will have their rt_addr set as 0.0.0.0 or ::. To kill all states using a gateway of 192.168.0.1 use:

```
      # pfctl -k gateway -k 192.168.0.1
```

A network prefix length can also be specified. To kill all states using a gateway in 192.168.0.0/24:

```
      # pfctl -k gateway -k 192.168.0.0/24
```

-M Kill matching states in the opposite direction (on other interfaces) when killing states. This applies to states killed using the -k option and also will apply to the flush command when flushing states. This is useful when an interface is specified when flushing states. Example:

```
# pfctl -M -i interface -Fs
```

-m Merge in explicitly given options without resetting those which are omitted. Allows single options to be modified without disturbing the others:

```
# echo "set loginterface fxp0" | pfctl -mf -
```

-N Load only the NAT rules present in the rule file. Other rules and options are ignored.

-n Do not actually load rules, just parse them.

-O Load only the options present in the rule file. Other rules and options are ignored.

-o *level*

Control the ruleset optimizer, overriding any rule file settings.

-o none Disable the ruleset optimizer.
-o basic Enable basic ruleset optimizations. This is the default behaviour.
-o profile Enable basic ruleset optimizations with profiling.
For further information on the ruleset optimizer, see pf.conf(5).

-P Do not perform service name lookup for port specific rules, instead display the ports numerically.

-p *device*

Use the device file *device* instead of the default */dev/pf*.

-q Only print errors and warnings.

-R Load only the filter rules present in the rule file. Other rules and options are ignored.

-r Perform reverse DNS lookups on states when displaying them.

-s *modifier*

Show the filter parameters specified by *modifier* (may be abbreviated):

-s nat Show the currently loaded NAT rules.

-s queue Show the currently loaded queue rules. When used together with **-v**, per-queue statistics are also shown. When used together with **-v -v**, **pfctl** will loop and show updated queue statistics every five seconds, including measured bandwidth and packets per second.

-s ether Show the currently loaded Ethernet rules. When used together with **-v**, the per-rule statistics (number of evaluations, packets, and bytes) are also shown.

-s rules Show the currently loaded filter rules. When used together with **-v**, the per-rule statistics (number of evaluations, packets, and bytes) are also shown. Note that the "skip step" optimization done automatically by the kernel will skip evaluation of rules where possible. Packets passed statefully are counted in the rule that created the state (even though the rule is not evaluated more than once for the entire connection).

-s Anchors Show the currently loaded anchors directly attached to the main ruleset. If **-a** *anchor* is specified as well, the anchors loaded directly below the given *anchor* are shown instead. If **-v** is specified, all anchors attached under the target anchor will be displayed recursively.

-s states Show the contents of the state table.

-s Sources Show the contents of the source tracking table.

-s info Show filter information (statistics and counters). When used together with **-v**, source tracking statistics are also shown.

-s Running Show the running status and provide a non-zero exit status when disabled.

-s labels Show per-rule statistics (label, evaluations, packets total, bytes total, packets in, bytes in, packets out, bytes out, state creations) of filter rules with labels, useful for accounting.

-s timeouts Show the current global timeouts.

-s memory Show the current pool memory hard limits.

-s Tables Show the list of tables.

-s osfp Show the list of operating system fingerprints.

-s Interfaces Show the list of interfaces and interface drivers available to PF. When used together with **-v**, it additionally lists which interfaces have skip rules activated. When used together with **-vv**, interface statistics are also shown. **-i** can be used to select an interface or a group of interfaces.

-s all Show all of the above, except for the lists of interfaces and operating system fingerprints.

-T *command* [*address* ...]

Specify the *command* (may be abbreviated) to apply to the table. Commands include:

-T kill Kill a table.

-T flush Flush all addresses of a table.

-T add Add one or more addresses in a table. Automatically create a nonexisting table.

-T delete Delete one or more addresses from a table.

-T expire *number*

Delete addresses which had their statistics cleared more than *number* seconds ago. For entries which have never had their statistics cleared, *number* refers to the time they were added to the table.

-T replace Replace the addresses of the table. Automatically create a nonexisting table.

-T show Show the content (addresses) of a table.

-T test Test if the given addresses match a table.

-T zero Clear all the statistics of a table.

-T load Load only the table definitions from pf.conf(5). This is used in conjunction with the **-f** flag, as in:

```
# pfctl -Tl -f pf.conf
```

For the **add**, **delete**, **replace**, and **test** commands, the list of addresses can be specified either directly on the command line and/or in an unformatted text file, using the **-f** flag. Comments starting with a ‘#’ or ‘;’ are allowed in the text file. With these commands, the **-v** flag can also be used once or twice, in which case **pfctl** will print the detailed result of the operation for each individual address, prefixed by one of the following letters:

A The address/network has been added.
C The address/network has been changed (negated).
D The address/network has been deleted.
M The address matches (**test** operation only).
X The address/network is duplicated and therefore ignored.
Y The address/network cannot be added/deleted due to conflicting ‘!’ attributes.
Z The address/network has been cleared (statistics).

Each table can maintain a set of counters that can be retrieved using the **-v** flag of **pfctl**. For example, the following commands define a wide open firewall which will keep track of packets going to or coming from the OpenBSD FTP server. The following commands configure the firewall and send 10 pings to the FTP server:

```
# printf "table <test> counters { ftp.openbsd.org }\n \
    pass out to <test>\n" | pfctl -f-
# ping -qc10 ftp.openbsd.org
```

We can now use the table **show** command to output, for each address and packet direction, the number of packets and bytes that are being passed or blocked by rules referencing the table. The time at which the current accounting started is also shown with the "Cleared" line.

```
# pfctl -t test -vTshow
```

```
    129.128.5.191
     Cleared:     Thu Feb 13 18:55:18 2003
     In/Block:    [ Packets: 0        Bytes: 0        ]
     In/Pass:     [ Packets: 10       Bytes: 840      ]
     Out/Block:   [ Packets: 0        Bytes: 0        ]
     Out/Pass:    [ Packets: 10       Bytes: 840      ]
```

Similarly, it is possible to view global information about the tables by using the **-v** modifier twice and the **-s Tables** command. This will display the number of addresses on each table, the number of rules which reference the table, and the global packet statistics for the whole table:

```
  # pfctl -vvsTables
  --a-r-C test
     Addresses:   1
     Cleared:     Thu Feb 13 18:55:18 2003
     References:  [ Anchors: 0        Rules: 1        ]
     Evaluations: [ NoMatch: 3496     Match: 1        ]
     In/Block:    [ Packets: 0        Bytes: 0        ]
     In/Pass:     [ Packets: 10       Bytes: 840      ]
     In/XPass:    [ Packets: 0        Bytes: 0        ]
     Out/Block:   [ Packets: 0        Bytes: 0        ]
     Out/Pass:    [ Packets: 10       Bytes: 840      ]
     Out/XPass:   [ Packets: 0        Bytes: 0        ]
```

As we can see here, only one packet - the initial ping request - matched the table, but all packets passing as the result of the state are correctly accounted for. Reloading the table(s) or ruleset will not affect packet accounting in any way. The two "XPass" counters are incremented instead of the "Pass" counters when a "stateful" packet is passed but does not match the table anymore. This will happen in our example if someone flushes the table while the ping(8) command is running.

When used with a single **-v**, **pfctl** will only display the first line containing the table flags and name. The flags are defined as follows:

c For constant tables, which cannot be altered outside pf.conf(5).

p For persistent tables, which do not get automatically killed when no rules refer to them.

a For tables which are part of the *active* tableset. Tables without this flag do not really exist, cannot contain addresses, and are only listed if the **-g** flag is given.

i For tables which are part of the *inactive* tableset. This flag can only be witnessed briefly during the loading of pf.conf(5).

r For tables which are referenced (used) by rules.

h This flag is set when a table in the main ruleset is hidden by one or more tables of the same name from anchors attached below it.

C This flag is set when per-address counters are enabled on the table.

-t *table* Specify the name of the table.

-v Produce more verbose output. A second use of **-v** will produce even more verbose output including ruleset warnings. See the previous section for its effect on table commands.

-x *level*

Set the debug *level* (may be abbreviated) to one of the following:

-x none Do not generate debug messages.

-x urgent Generate debug messages only for serious errors.

-x misc Generate debug messages for various errors.

-x loud Generate debug messages for common conditions.

-z Clear per-rule statistics.

FILES

/etc/pf.conf Packet filter rules file.

/etc/pf.os Passive operating system fingerprint database.

SEE ALSO

pf(4), pf.conf(5), pf.os(5), rc.conf(5), services(5), sysctl.conf(5), authpf(8), ftp-proxy(8), rc(8), sysctl(8)

HISTORY

The **pfctl** program and the pf(4) filter mechanism appeared in OpenBSD 3.0. They first appeared in FreeBSD 5.3 ported from the version in OpenBSD 3.5