NAME

pfsync - packet filter state table sychronisation interface

SYNOPSIS

device pfsync

DESCRIPTION

The **pfsync** interface is a pseudo-device which exposes certain changes to the state table used by pf(4). State changes can be viewed by invoking tcpdump(1) on the **pfsync** interface. If configured with a physical synchronisation interface, **pfsync** will also send state changes out on that interface, and insert state changes received on that interface from other systems into the state table.

By default, all local changes to the state table are exposed via **pfsync**. State changes from packets received by **pfsync** over the network are not rebroadcast. Updates to states created by a rule marked with the *no-sync* keyword are ignored by the **pfsync** interface (see pf.conf(5) for details).

The **pfsync** interface will attempt to collapse multiple state updates into a single packet where possible. The maximum number of times a single state can be updated before a **pfsync** packet will be sent out is controlled by the *maxupd* parameter to ifconfig (see ifconfig(8) and the example below for more details). The sending out of a **pfsync** packet will be delayed by a maximum of one second.

NETWORK SYNCHRONISATION

States can be synchronised between two or more firewalls using this interface, by specifying a synchronisation interface using ifconfig(8). For example, the following command sets fxp0 as the synchronisation interface:

```
# ifconfig pfsync0 syncdev fxp0
```

By default, state change messages are sent out on the synchronisation interface using IP multicast packets to the 224.0.0.240 group address. An alternative destination address for **pfsync** packets can be specified using the **syncpeer** keyword. This can be used in combination with ipsec(4) to protect the synchronisation traffic. In such a configuration, the syncdev should be set to the enc(4) interface, as this is where the traffic arrives when it is decapsulated, e.g.:

```
# ifconfig pfsync0 syncpeer 10.0.0.2 syncdev enc0
```

It is important that the pfsync traffic be well secured as there is no authentication on the protocol and it would be trivial to spoof packets which create states, bypassing the pf ruleset. Either run the pfsync protocol on a trusted network - ideally a network dedicated to pfsync messages such as a crossover cable between two firewalls, or specify a peer address and protect the traffic with ipsec(4).

pfsync has the following sysctl(8) tunables:

net.pfsync.carp_demotion_factor
Value added to *net.inet.carp.demotion* while **pfsync** tries to perform its bulk update. See carp(4) for more information. Default value is 240.

net.pfsync.pfsync_buckets
The number of **pfsync** buckets. This affects the performance and memory tradeoff. Defaults to twice the number of CPUs. Change only if benchmarks show this helps on your workload.

EXAMPLES

pfsync and carp(4) can be used together to provide automatic failover of a pair of firewalls configured in parallel. One firewall will handle all traffic until it dies, is shut down, or is manually demoted, at which point the second firewall will take over automatically.

Both firewalls in this example have three sis(4) interfaces. sis0 is the external interface, on the 10.0.0.0/24 subnet; sis1 is the internal interface, on the 192.168.0.0/24 subnet; and sis2 is the **pfsync** interface, using the 192.168.254.0/24 subnet. A crossover cable connects the two firewalls via their sis2 interfaces. On all three interfaces, firewall A uses the .254 address, while firewall B uses .253. The interfaces are configured as follows (firewall A unless otherwise indicated):

Interfaces configuration in */etc/rc.conf*:

```
network_interfaces="lo0 sis0 sis1 sis2"
ifconfig_sis0="10.0.0.254/24"
ifconfig_sis0_alias0="inet 10.0.0.1/24 vhid 1 pass foo"
ifconfig_sis1="192.168.0.254/24"
ifconfig_sis1_alias0="inet 192.168.0.1/24 vhid 2 pass bar"
ifconfig_sis2="192.168.254.254/24"
pfsync_enable="YES"
pfsync_syncdev="sis2"
```

pf(4) must also be configured to allow **pfsync** and carp(4) traffic through. The following should be added to the top of */etc/pf.conf*:

```
pass quick on { sis2 } proto pfsync keep state (no-sync)
pass on { sis0 sis1 } proto carp keep state (no-sync)
```

It is preferable that one firewall handle the forwarding of all the traffic, therefore the *advskew* on the

backup firewall's carp(4) vhids should be set to something higher than the primary's. For example, if firewall B is the backup, its carp1 configuration would look like this:

```
ifconfig_sis1_alias0="inet 192.168.0.1/24 vhid 2 pass bar advskew 100"
```

The following must also be added to */etc/sysctl.conf*:

```
net.inet.carp.preempt=1
```

SEE ALSO

tcpdump(1), bpf(4), carp(4), enc(4), inet(4), inet6(4), ipsec(4), netintro(4), pf(4), pf.conf(5), protocols(5), rc.conf(5), ifconfig(8)

HISTORY

The **pfsync** device first appeared in OpenBSD 3.3. It was first imported to FreeBSD 5.3.

The **pfsync** protocol and kernel implementation were significantly modified in FreeBSD 9.0. The newer protocol is not compatible with older one and will not interoperate with it.