NAME

enc - Encapsulating Interface

SYNOPSIS

To compile this driver into the kernel, place the following line in your kernel configuration file:

device enc

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

if_enc_load="YES"

DESCRIPTION

The **enc** interface is a software loopback mechanism that allows hosts or firewalls to filter ipsec(4) traffic using any firewall package that hooks in via the pfil(9) framework.

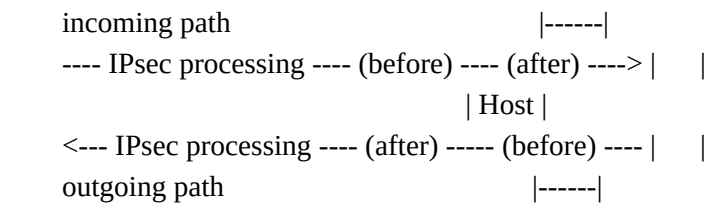
The **enc** interface allows an administrator to see incoming and outgoing packets before and after they will be or have been processed by ipsec(4) via tcpdump(1).

The "enc0" interface inherits all IPsec traffic. Thus all IPsec traffic can be filtered based on "enc0", and all IPsec traffic could be seen by invoking tcpdump(1) on the "enc0" interface.

What can be seen with tcpdump(1) and what will be passed on to the firewalls via the pfil(9) framework can be independently controlled using the following sysctl(8) variables:

Name	Defaults	Suggested
net.enc.out.ipsec_bpf_mask	0x00000003	0x00000001
net.enc.out.ipsec_filter_mask	0x00000001	0x00000001
net.enc.in.ipsec_bpf_mask	0x00000001	0x00000002
net.enc.in.ipsec_filter_mask	0x00000001	0x00000002

For the incoming path a value of 0x1 means "before stripping off the outer header" and 0x2 means "after stripping off the outer header". For the outgoing path 0x1 means "with only the inner header" and 0x2 means "with outer and inner headers".



Most people will want to run with the suggested defaults for **ipsec_filter_mask** and rely on the security policy database for the outer headers.

Note that packets are captured by BPF before firewall processing. The special value 0x4 can be configured in the *ipsec_bpf_mask* and packets will be also captured after firewall processing.

EXAMPLES

To see the packets the processed via ipsec(4), adjust the sysctl(8) variables according to your need and run:

```
tcpdump -i enc0
```

SEE ALSO

tcpdump(1), bpf(4), ipf(4), ipfw(4), ipsec(4), pf(4)