

NAME

ipnat - Network Address Translation kernel interface

SYNOPSIS

```
#include <netinet/ip_compat.h>
#include <netinet/ip_fil.h>
#include <netinet/ip_proxy.h>
#include <netinet/ip_nat.h>
```

IOCTLS

To add and delete rules to the NAT list, two 'basic' ioctls are provided for use. The ioctl's are called as:

```
ioctl(fd, SIOCADNAT, struct ipnat **)
ioctl(fd, SIOCRMNAT, struct ipnat **)
ioctl(fd, SIOCGNATS, struct natstat **)
ioctl(fd, SIOCGNATL, struct natlookup **)
```

Unlike **ipf(4)**, there is only a single list supported by the kernel NAT interface. An inactive list which can be swapped to is not currently supported.

These ioctl's are implemented as being routing ioctls and thus the same rules for the various routing ioctls and the file descriptor are employed, mainly being that the fd must be that of the device associated with the module (i.e., /dev/ipl).

The structure used with the NAT interface is described below:

```
typedef struct ipnat {
    struct ipnat *in_next;
    void *in_ifp;
    u_short in_flags;
    u_short in_pnext;
    u_short in_port[2];
    struct in_addr in_in[2];
    struct in_addr in_out[2];
    struct in_addr in_nextip;
    int in_space;
    int in_redir; /* 0 if it's a mapping, 1 if it's a hard redir */
    char in_ifname[IFNAMSIZ];
} ipnat_t;
```

```

#define in_pmin    in_port[0]    /* Also holds static redir port */
#define in_pmax    in_port[1]
#define in_nip     in_nextip.s_addr
#define in_inip    in_in[0].s_addr
#define in_inmsk   in_in[1].s_addr
#define in_outip   in_out[0].s_addr
#define in_outmsk  in_out[1].s_addr

```

Recognised values for `in_redir`:

```

#define NAT_MAP      0
#define NAT_REDIRECT 1

```

NAT statistics Statistics on the number of packets mapped, going in and out are kept, the number of times a new entry is added and deleted (through expiration) to the NAT table and the current usage level of the NAT table.

Pointers to the NAT table inside the kernel, as well as to the top of the internal NAT lists constructed with the **SIOCADNAT** ioctls. The table itself is a hash table of size `NAT_SIZE` (default size is 367).

To retrieve the statistics, the **SIOCGNATS** ioctl must be used, with the appropriate structure passed by reference, as follows:

```
ioctl(fd, SIOCGNATS, struct natstat *)
```

```

typedef struct natstat {
    u_long ns_mapped[2];
    u_long ns_added;
    u_long ns_expire;
    u_long ns_inuse;
    nat_t  ***ns_table;
    ipnat_t *ns_list;
} natstat_t;

```

BUGS

It would be nice if there were more flexibility when adding and deleting filter rules.

FILES

/dev/ipnat

SEE ALSO

ipf(4), ipnat(5), ipf(8), ipnat(8), ipfstat(8)