

NAME

ip-xfrm – transform configuration

SYNOPSIS

ip [*OPTIONS*] **xfrm** { *COMMAND* | **help** }

ip xfrm *XFRM-OBJECT* { *COMMAND* | **help** }

XFRM-OBJECT := **state** | **policy** | **monitor**

ip xfrm state { **add** | **update** } *ID* [*ALGO-LIST*] [**mode** *MODE*] [**mark** *MARK* [**mask** *MASK*]] [**reqid** *REQID*] [**seq** *SEQ*] [**replay-window** *SIZE*] [**replay-seq** *SEQ*] [**replay-oseq** *SEQ*] [**replay-seq-hi** *SEQ*] [**replay-oseq-hi** *SEQ*] [**flag** *FLAG-LIST*] [**sel** *SELECTOR*] [*LIMIT-LIST*] [**encap** *ENCAP*] [**coa** *ADDR[/PLEN]*] [**ctx** *CTX*] [**extra-flag** *EXTRA-FLAG-LIST*] [**output-mark** *OUTPUT-MARK*]

ip xfrm state allocspi *ID* [**mode** *MODE*] [**mark** *MARK* [**mask** *MASK*]] [**reqid** *REQID*] [**seq** *SEQ*] [**min** *SPI* **max** *SPI*]

ip xfrm state { **delete** | **get** } *ID* [**mark** *MARK* [**mask** *MASK*]]

ip [-4 | -6] **xfrm state deleteall** [*ID*] [**mode** *MODE*] [**reqid** *REQID*] [**flag** *FLAG-LIST*]

ip [-4 | -6] **xfrm state list** [*ID*] [**nokeys**] [**mode** *MODE*] [**reqid** *REQID*] [**flag** *FLAG-LIST*]

ip xfrm state flush [**proto** *XFRM-PROTO*]

ip xfrm state count

ID := [**src** *ADDR*] [**dst** *ADDR*] [**proto** *XFRM-PROTO*] [**spi** *SPI*]

XFRM-PROTO := **esp** | **ah** | **comp** | **route2** | **hao**

ALGO-LIST := [*ALGO-LIST*] *ALGO*

ALGO := { **enc** | **auth** } *ALGO-NAME* *ALGO-KEYMAT* |
auth-trunc *ALGO-NAME* *ALGO-KEYMAT* *ALGO-TRUNC-LEN* |
aead *ALGO-NAME* *ALGO-KEYMAT* *ALGO-ICV-LEN* |
comp *ALGO-NAME*

MODE := **transport** | **tunnel** | **beet** | **ro** | **in_trigger**

FLAG-LIST := [*FLAG-LIST*] *FLAG*

FLAG := **noecn** | **decap-dscp** | **nopmtudisc** | **wildrecv** | **icmp** | **af-unspec** | **align4** | **esn**

SELECTOR := [**src** *ADDR[/PLEN]*] [**dst** *ADDR[/PLEN]*] [**dev** *DEV*] [*UPSPEC*]

UPSPEC := **proto** { *PROTO* |
{ **tcp** | **udp** | **sctp** | **dccp** } [**sport** *PORT*] [**dport** *PORT*] |
{ **icmp** | **ipv6-icmp** | **mobility-header** } [**type** *NUMBER*] [**code** *NUMBER*] |

```

    gre [ key { DOTTED-QUAD | NUMBER } ] }

LIMIT-LIST := [ LIMIT-LIST ] limit LIMIT

LIMIT := { time-soft | time-hard | time-use-soft | time-use-hard } SECONDS |
          { byte-soft | byte-hard } SIZE |
          { packet-soft | packet-hard } COUNT

ENCAP := { espinudp | espinudp-nonike } SPORT DPORT OADDR

EXTRA-FLAG-LIST := [ EXTRA-FLAG-LIST ] EXTRA-FLAG

EXTRA-FLAG := dont-encap-dscp

ip xfrm policy { add | update } SELECTOR dir DIR [ ctx CTX ] [ mark MARK [ mask MASK ] ] [ in-
               dex INDEX ] [ ptype PTYPE ] [ action ACTION ] [ priority PRIORITY ] [ flag
               FLAG-LIST ] [ LIMIT-LIST ] [ TMPL-LIST ]

ip xfrm policy { delete | get } { SELECTOR | index INDEX } dir DIR [ ctx CTX ] [ mark MARK [ mask
               MASK ] ] [ ptype PTYPE ]

ip [ -4 | -6 ] xfrm policy { deleteall | list } [ nosock ] [ SELECTOR ] [ dir DIR ] [ index INDEX ] [ ptype
               PTYPE ] [ action ACTION ] [ priority PRIORITY ] [ flag FLAG-LIST ]

ip xfrm policy flush [ ptype PTYPE ]

ip xfrm policy count

ip xfrm policy set [ hthresh4 LBITS RBITS ] [ hthresh6 LBITS RBITS ]

SELECTOR := [ src ADDR[/PLEN] ] [ dst ADDR[/PLEN] ] [ dev DEV ] [ UPSPEC ]

UPSPEC := proto { PROTO |
                { tcp | udp | sctp | dccp } [ sport PORT ] [ dport PORT ] |
                { icmp | ipv6-icmp | mobility-header } [ type NUMBER ] [ code NUMBER ] |
                gre [ key { DOTTED-QUAD | NUMBER } ] }

DIR := in | out | fwd

PTYPE := main | sub

ACTION := allow | block

FLAG-LIST := [ FLAG-LIST ] FLAG

FLAG := localok | icmp

LIMIT-LIST := [ LIMIT-LIST ] limit LIMIT

LIMIT := { time-soft | time-hard | time-use-soft | time-use-hard } SECONDS |
          { byte-soft | byte-hard } SIZE |
          { packet-soft | packet-hard } COUNT

TMPL-LIST := [ TMPL-LIST ] tmpl TMPL

```

```

TMPL := ID [ mode MODE ] [ reqid REQID ] [ level LEVEL ]

ID := [ src ADDR ] [ dst ADDR ] [ proto XFRM-PROTO ] [ spi SPI ]

XFRM-PROTO := esp | ah | comp | route2 | hao

MODE := transport | tunnel | beet | ro | in_trigger

LEVEL := required | use

ip xfrm monitor [ all-nsid ] [ nokeys ] [ all
                  | LISTofXFRM-OBJECTS ]

LISTofXFRM-OBJECTS := [ LISTofXFRM-OBJECTS ] XFRM-OBJECT

XFRM-OBJECT := acquire | expire | SA | policy | aevent | report

```

DESCRIPTION

xfrm is an IP framework for transforming packets (such as encrypting their payloads). This framework is used to implement the IPsec protocol suite (with the **state** object operating on the Security Association Database, and the **policy** object operating on the Security Policy Database). It is also used for the IP Payload Compression Protocol and features of Mobile IPv6.

<code>ip xfrm state add</code>	add new state into xfrm
<code>ip xfrm state update</code>	update existing state in xfrm
<code>ip xfrm state allocspi</code>	allocate an SPI value
<code>ip xfrm state delete</code>	delete existing state in xfrm
<code>ip xfrm state get</code>	get existing state in xfrm
<code>ip xfrm state deleteall</code>	delete all existing state in xfrm
<code>ip xfrm state list</code>	print out the list of existing state in xfrm
<code>ip xfrm state flush</code>	flush all state in xfrm
<code>ip xfrm state count</code>	count all existing state in xfrm

ID is specified by a source address, destination address, transform protocol *XFRM-PROTO*, and/or Security Parameter Index *SPI*. (For IP Payload Compression, the Compression Parameter Index or CPI is used for *SPI*.)

XFRM-PROTO

specifies a transform protocol: IPsec Encapsulating Security Payload (**esp**), IPsec Authentication Header (**ah**), IP Payload Compression (**comp**), Mobile IPv6 Type 2 Routing Header (**route2**), or Mobile IPv6 Home Address Option (**hao**).

ALGO-LIST

contains one or more algorithms to use. Each algorithm *ALGO* is specified by:

- the algorithm type: encryption (**enc**), authentication (**auth** or **auth-trunc**), authenticated encryption with associated data (**aead**), or compression (**comp**)
- the algorithm name *ALGO-NAME* (see below)
- (for all except **comp**) the keying material *ALGO-KEYMAT*, which may include both a key and a salt or nonce value; refer to the corresponding RFC

- (for **auth-trunc** only) the truncation length *ALGO-TRUNC-LEN* in bits
- (for **aead** only) the Integrity Check Value length *ALGO-ICV-LEN* in bits

Encryption algorithms include **ecb(cipher_null)**, **cbc(des)**, **cbc(des3_ede)**, **cbc(cast5)**, **cbc(blowfish)**, **cbc(aes)**, **cbc(serpent)**, **cbc(camellia)**, **cbc(twofish)**, and **rfc3686(ctr(aes))**.

Authentication algorithms include **digest_null**, **hmac(md5)**, **hmac(sha1)**, **hmac(sha256)**, **hmac(sha384)**, **hmac(sha512)**, **hmac(rmd160)**, and **xcbc(aes)**.

Authenticated encryption with associated data (AEAD) algorithms include **rfc4106(gcm(aes))**, **rfc4309(ccm(aes))**, and **rfc4543(gcm(aes))**.

Compression algorithms include **deflate**, **lzs**, and **lzjh**.

MODE specifies a mode of operation for the transform protocol. IPsec and IP Payload Compression modes are **transport**, **tunnel**, and (for IPsec ESP only) Bound End-to-End Tunnel (**beet**). Mobile IPv6 modes are route optimization (**ro**) and inbound trigger (**in_trigger**).

FLAG-LIST

contains one or more of the following optional flags: **noecn**, **decap-dscp**, **nopmtudisc**, **wildrecv**, **icmp**, **af-unspec**, **align4**, or **esn**.

SELECTOR

selects the traffic that will be controlled by the policy, based on the source address, the destination address, the network device, and/or *UPSPEC*.

UPSPEC

selects traffic by protocol. For the **tcp**, **udp**, **sctp**, or **dccp** protocols, the source and destination port can optionally be specified. For the **icmp**, **ipv6-icmp**, or **mobility-header** protocols, the type and code numbers can optionally be specified. For the **gre** protocol, the key can optionally be specified as a dotted-quad or number. Other protocols can be selected by name or number *PROTO*.

LIMIT-LIST

sets limits in seconds, bytes, or numbers of packets.

ENCAP

encapsulates packets with protocol **espinudp** or **espinudp-nonike**, using source port *SPORT*, destination port *DPORT*, and original address *OADDR*.

MARK used to match xfrm policies and states

OUTPUT-MARK

used to set the output mark to influence the routing of the packets emitted by the state

ip xfrm policy add	add a new policy
ip xfrm policy update	update an existing policy
ip xfrm policy delete	delete an existing policy

<code>ip xfrm policy get</code>	get an existing policy
<code>ip xfrm policy deleteall</code>	delete all existing xfrm policies
<code>ip xfrm policy list</code>	print out the list of xfrm policies
<code>ip xfrm policy flush</code>	flush policies

nosock filter (remove) all socket policies from the output.

SELECTOR

selects the traffic that will be controlled by the policy, based on the source address, the destination address, the network device, and/or *UPSPEC*.

UPSPEC

selects traffic by protocol. For the **tcp**, **udp**, **sctp**, or **dccp** protocols, the source and destination port can optionally be specified. For the **icmp**, **ipv6-icmp**, or **mobility-header** protocols, the type and code numbers can optionally be specified. For the **gre** protocol, the key can optionally be specified as a dotted-quad or number. Other protocols can be selected by name or number *PROTO*.

DIR selects the policy direction as **in**, **out**, or **fwd**.

CTX sets the security context.

PTYPE can be **main** (default) or **sub**.

ACTION

can be **allow** (default) or **block**.

PRIORITY

is a number that defaults to zero.

FLAG-LIST

contains one or both of the following optional flags: **local** or **icmp**.

LIMIT-LIST

sets limits in seconds, bytes, or numbers of packets.

TMPL-LIST

is a template list specified using *ID*, *MODE*, *REQID*, and/or *LEVEL*.

ID is specified by a source address, destination address, transform protocol *XFRM-PROTO*, and/or Security Parameter Index *SPI*. (For IP Payload Compression, the Compression Parameter Index or CPI is used for *SPI*.)

XFRM-PROTO

specifies a transform protocol: IPsec Encapsulating Security Payload (**esp**), IPsec Authentication Header (**ah**), IP Payload Compression (**comp**), Mobile IPv6 Type 2 Routing Header (**route2**), or Mobile IPv6 Home Address Option (**hao**).

MODE specifies a mode of operation for the transform protocol. IPsec and IP Payload Compression modes are **transport**, **tunnel**, and (for IPsec ESP only) Bound End-to-End Tunnel (**beet**). Mobile IPv6 modes are route optimization (**ro**) and inbound trigger (**in_trigger**).

LEVEL can be **required** (default) or **use**.

`ip xfrm policy count` count existing policies

Use one or more `-s` options to display more details, including policy hash table information.

`ip xfrm policy set` configure the policy hash table

Security policies whose address prefix lengths are greater than or equal policy hash table thresholds are hashed. Others are stored in the `policy_inexact` chained list.

LBITS specifies the minimum local address prefix length of policies that are stored in the Security Policy Database hash table.

RBITS specifies the minimum remote address prefix length of policies that are stored in the Security Policy Database hash table.

`ip xfrm monitor` state monitoring for xfrm objects

The xfrm objects to monitor can be optionally specified.

If the **all-nsid** option is set, the program listens to all network namespaces that have a nsid assigned into the network namespace where the program is running. A prefix is displayed to show the network namespace where the message originates. Example:

```
[nsid 1]Flushed state proto 0
```

AUTHOR

Manpage revised by David Ward <david.ward@ll.mit.edu>

Manpage revised by Christophe Gouault <christophe.gouault@6wind.com>

Manpage revised by Nicolas Dichtel <nicolas.dichtel@6wind.com>