NAME

stf - 6to4 tunnel interface

SYNOPSIS

device stf

DESCRIPTION

The **stf** interface supports "6to4" and "6rd" IPv6 in IPv4 encapsulation. It can tunnel IPv6 traffic over IPv4, as specified in RFC3056 or RFC5969.

For ordinary nodes in a 6to4 or 6RD site, you do not need **stf** interface. The **stf** interface is necessary for site border routers (called "6to4 routers" or "6rd Customer Edge (CE)" in the specification).

Each **stf** interface is created at runtime using interface cloning. This is most easily done with the ifconfig(8) **create** command or using the *cloned_interfaces* variable in rc.conf(5).

6to4

Due to the way 6to4 protocol is specified, **stf** interface requires certain configuration to work properly. Single (no more than 1) valid 6to4 address needs to be configured to the interface. "A valid 6to4 address" is an address which has the following properties. If any of the following properties are not satisfied, **stf** raises runtime error on packet transmission. Read the specification for more details.

- matches 2002:xxyy:zzuu::/48 where xxyy:zzuu is a hexadecimal notation of an IPv4 address for the node. IPv4 address can be taken from any of interfaces your node has. Since the specification forbids the use of IPv4 private address, the address needs to be a global IPv4 address.

- Subnet identifier portion (48th to 63rd bit) and interface identifier portion (lower 64 bits) are properly filled to avoid address collisions.

If you would like the node to behave as a relay router, the prefix length for the IPv6 interface address needs to be 16 so that the node would consider any 6to4 destination as "on-link". If you would like to restrict 6to4 peers to be inside certain IPv4 prefix, you may want to configure IPv6 prefix length as "16 + IPv4 prefix length". **stf** interface will check the IPv4 source address on packets, if the IPv6 prefix length is larger than 16.

stf can be configured to be ECN friendly. This can be configured by IFF_LINK1. See gif(4) for details.

Please note that 6to4 specification is written as "accept tunnelled packet from everyone" tunnelling device. By enabling **stf** device, you are making it much easier for malicious parties to inject fabricated IPv6 packet to your node. Also, malicious party can inject an IPv6 packet with fabricated source address to make your node generate improper tunnelled packet. Administrators must take caution when enabling the interface. To prevent possible attacks, **stf** interface filters out the following packets. Note that the checks are no way complete:

- Packets with IPv4 unspecified address as outer IPv4 source/destination (0.0.0.0/8)

- Packets with loopback address as outer IPv4 source/destination (127.0.0.0/8)

- Packets with IPv4 multicast address as outer IPv4 source/destination (224.0.0.0/4)

- Packets with limited broadcast address as outer IPv4 source/destination (255.0.0.0/8)

- Packets with private address as outer IPv4 source/destination (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16)

- Packets with subnet broadcast address as outer IPv4 source/destination. The check is made against subnet broadcast addresses for all of the directly connected subnets.

- Packets that does not pass ingress filtering. Outer IPv4 source address must meet the IPv4 topology on the routing table. Ingress filter can be turned off by IFF_LINK2 bit.

- The same set of rules are applied against the IPv4 address embedded into inner IPv6 address, if the IPv6 address matches 6to4 prefix.

It is recommended to filter/audit incoming IPv4 packet with IP protocol number 41, as necessary. It is also recommended to filter/audit encapsulated IPv6 packets as well. You may also want to run normal ingress filter against inner IPv6 address to avoid spoofing.

By setting the IFF_LINK0 flag on the **stf** interface, it is possible to disable the input path, making the direct attacks from the outside impossible. Note, however, there are other security risks exist. If you wish to use the configuration, you must not advertise your 6to4 address to others.

6rd

Like "6to4" "6rd" also requires configuration before it can be used. The required configuration parameters are:

- The IPv6 address and prefix length.

- The border router IPv4 address.

- The IPv4 WAN address.

- The prefix length of the IPv4 WAN address.

These can parameters are all configured through ifconfig(8).

The IPv6 address and prefix length can be configured like any other IPv6 address. Note that the prefix length is the IPv6 prefix length excluding the embedded IPv4 address bits. The prefix length of the delegated network is the sum of the IPv6 prefix length and the IPv4 prefix length.

The border router IPv4 address is configured with the ifconfig(8) **stfv4br** command.

The IPv4 WAN address and IPv4 prefix length are configured using the ifconfig(8) **stfv4net** command.

SYSCTL VARIABLES

The following sysctl(8) variables can be used to control the behavior of the **stf**. The default value is shown next to each variable.

net.link.stf.permit_rfc1918: 0

The RFC3056 requires the use of globally unique 32-bit IPv4 addresses. This sysctl variable controls the behaviour of this requirement. When it set to not 0, **stf** allows the use of private IPv4 addresses described in the RFC1918. This may be useful for an Intranet environment or when some mechanisms of network address translation (NAT) are used.

EXAMPLES

Note that 8504:0506 is equal to 133.4.5.6, written in hexadecimals.

```
# ifconfig ne0 inet 133.4.5.6 netmask 0xffffff00
# ifconfig stf0 inet6 2002:8504:0506:0000:a00:5aff:fe38:6f86 \
          prefixlen 16 alias
```

The following configuration accepts packets from IPv4 source 9.1.0.0/16 only. It emits 6to4 packet only for IPv6 destination 2002:0901::/32 (IPv4 destination will match 9.1.0.0/16).

```
# ifconfig ne0 inet 9.1.2.3 netmask 0xffff0000
# ifconfig stf0 inet6 2002:0901:0203:0000:a00:5aff:fe38:6f86 \
          prefixlen 32 alias
```

The following configuration uses the **stf** interface as an output-only device. You need to have alternative IPv6 connectivity (other than 6to4) to use this configuration. For outbound traffic, you can reach other 6to4 networks efficiently via **stf**. For inbound traffic, you will not receive any 6to4-tunneled packets (less security drawbacks). Be careful not to advertise your 6to4 prefix to others (2002:8504:0506::/48), and not to use your 6to4 prefix as a source.

```
# ifconfig ne0 inet 133.4.5.6 netmask 0xffffff00
# ifconfig stf0 inet6 2002:8504:0506:0000:a00:5aff:fe38:6f86 \
        prefixlen 16 alias deprecated link0
# route add -inet6 2002:: -prefixlen 16 ::1
# route change -inet6 2002:: -prefixlen 16 ::1 -ifp stf0
```

The following example configures a "6rd" tunnel on a "6rd CE" where the ISP's "6rd" IPv6 prefix is 2001:db8::/32. The border router is 192.0.2.1. The "6rd CE" has a WAN address of 192.0.2.2 and the full IPv4 address is embedded in the "6rd IPv6 address:"

```
# ifconfig stf0 inet6 2001:db8:c000:0202:: prefixlen 32 up
# ifconfig stf0 stfv4br 192.0.2.1
# ifconfig stf0 stfv4net 192.0.2.2/32
```

SEE ALSO

gif(4), inet(4), inet6(4)

Brian Carpenter and Keith Moore, *Connection of IPv6 Domains via IPv4 Clouds*, RFC, 3056, February 2001.

Jun-ichiro itojun Hagino, *Possible abuse against IPv6 transition technologies*, draft-itojun-ipv6-transition-abuse-01.txt, July 2000, work in progress.

HISTORY

The **stf** device first appeared in WIDE/KAME IPv6 stack.

BUGS

No more than one **stf** interface is allowed for a node, and no more than one IPv6 interface address is allowed for an **stf** interface. It is to avoid source address selection conflicts between IPv6 layer and IPv4 layer, and to cope with ingress filtering rule on the other side. This is a feature to make **stf** work right for all occasions.