Project name: IPV6 GRE TUNNEL
Date of the project: 8 september 2007
Persons involved in the project: Iwan Hoogendoorn & Willem Eradus
Project status: 100%
Network diagram included: No

Detailed project description

The goal of this project is to set up a IPV6 GRE TUNNEL between 2 sites.
The sites used in this project are from Iwan (Rotterdam-The Netherlands) & Willem (De Kwakel-The Netherlands)
The end result will be that Willem can reach Iwan's IPV6 internal network and Iwan Willem's internal IPV6 network.

IPv6 network segments

The first 3 octets are masked because of the security policy that we have.

```
Willem's internal network = xxxx:xxx:xx9C:FEED::/64 &
xxxx:xCF:xxxx:DEAD::/64
Iwan's internal network = xxx:xxx:xcc:50::/64
```

Steps

What you basically do is that you:
- create a tunnel interface
- assign an IPV6 address to the tunnel
- enable ipv6
- specify the dialer interface of your internet connection
- specify the ipv4 address where you are peering with
- create ipv6 ACL's for security purposes
- make sure the traffic is routed trough the new ipv6 GRE tunnel

Configuration snips

Willem Cisco 877:

```
interface Tunnel1
 description IPv6 tunnel Willem-to-Iwan
 no ip address
 ipv6 address xxxx:xxx:xxCF:1::1/64
 ipv6 enable
 ipv6 traffic-filter IPV6_IN in
 ipv6 traffic-filter IPV6_OUT out
tunnel source dialer0
 tunnel destination <ipv4 address peer>
 tunnel mode ipv6ip
end
!
ipv6 access-list IPV6_IN
 evaluate IPV6_REFLECT
 permit icmp any any
 sequence 1000 deny ipv6 any any
!
ipv6 access-list IPV6_OUT
 permit tcp any any reflect IPV6_REFLECT
 permit udp any any reflect IPV6_REFLECT
 permit icmp any any
 deny ipv6 any any log
!
ipv6 route xxx:xxx:xcc:50::/64 tunnel 1
```

Iwan Cisco 877:

```
interface Tunnel1
 description IPv6 tunnel Willem-to-Iwan
 no ip address
 ipv6 address xxxx:xxx:xxCF:2::2/64
 ipv6 enable
 ipv6 traffic-filter IPV6_IN in
 ipv6 traffic-filter IPV6_OUT out
tunnel source dialer0
 tunnel destination <ipv4 address peer>
 tunnel mode ipv6ip
end
!
ipv6 access-list IPV6_IN
 evaluate IPV6_REFLECT
 permit icmp any any
 sequence 1000 deny ipv6 any any
!
ipv6 access-list IPV6_OUT
 permit tcp any any reflect IPV6_REFLECT
 permit udp any any reflect IPV6_REFLECT
 permit icmp any any
 deny ipv6 any any log
!
ipv6 route xxxx:xxx:x9C:FEED::/64 tunnel 1
ipv6 route xxxx:xxx:xxCF:DEAD::/64 tunnel 1
```

Testing

Traceroute from Rani (Linux server on Iwan's ipv6 network)
The traceroute is done with the –I option, this option is to do a traceroute with ICMP.

```
[root@rani ~]# traceroute6 -I xxxx:xxx:x9c:feed:211:24ff:fe3c:dc32
traceroute to xxxx:xxx:x9c:feed:211:24ff:fe3c:dc32
(xxxx:xxx:x9c:feed:211:24ff:fe3c:dc32), 30 hops max, 40 byte packets
 1  xxxx:xxx:xcc:50::1 (xxxx:xxx:xcc:50::1)  2.968 ms  3.556 ms  4.137 ms
 2  xxxx:xxx:xcc:dead::1 (xxxx:xxx:xcc:dead::1)  7.686 ms  7.850 ms  8.017 ms
 3  xxxx:xxx:xxcf:2::1 (xxxx:xxx:xxcf:2::1)  35.834 ms  39.603 ms  42.559 ms
 4  imac-e.ipv6.xxx.net (xxxx:xxx:x9c:feed:211:24ff:fe3c:dc32)  45.797 ms  47.917
ms  51.907 ms
[root@rani ~]#
```

The traffic is going trough the **`(xxxx:xxx:xcc:dead::1)`** hop so this is the proof that the tunnel is used.
Check if the tunnel is up:

```
NL-RTD-RT-01#sh ipv6 interface brief Tunnel1
Tunnel1                    [up/up]
    xxxx::xxxx:xxxx
    xxxx:xxx:xxCF:2::2
NL-RTD-RT-01#
```

Devices used in the project with specified software:
- 2 x Cisco 877 Router
- 1 x linux server (Fedora Core 5)
- 1 x I-MAC

Problems experienced:
None