# EIGRP on SVTI, DVTI, and IKEv2 FlexVPN with the "IP[v6] Unnumbered" Command Configuration Example

**Document ID: 116346**

Contributed by Michal Garcarz and Olivier Pelerin, Cisco TAC Engineers.

Sep 18, 2013

## Contents

## Introduction

This document describes how to configure Enhanced Interior Gateway Routing Protocol (EIGRP) in a number of commonly−encountered scenarios on Cisco IOS®. In order to accept an EIGRP neighbor adjacency, Cisco IOS must obtain the EIGRP HELLO packet from an IP address within the same subnet. It is possible to disable that verification with the ***ip unnumbered*** command.

The first part of the article presents an EIGRP failure when it receives a packet that is not in the same subnet.

Another example demonstrates the use of the ***ip unnumbered*** command that disables that verification, and allows EIGRP to form an adjacency between peers that belong to different subnets.

This article also presents a FlexVPN Hub and Spoke deployment with an IP address sent from the server. For this scenario, verification of subnets is disabled for the ***ip address negotiated*** command and also for the ***ip unnumbered*** command. The ***ip unnumbered*** command is primarily used for Point−to−Point (P2P) type interfaces, and this makes FlexVPN a perfect fit since it is based on a P2P architecture.

Lastly, an IPv6 scenario is presented along with differences for both Static Virtual Tunnel Interfaces (SVTI) and Dynamic Virtual Tunnel Interfaces (DVTI). There are slight changes in behavior when you compare IPv6 to IPv4 scenarios.

Additionally, the changes between Cisco IOS Versions 15.1 and 15.3 are presented (Cisco bug ID CSCtx45062).

The ***ip unnumbered*** command is always needed for DVTI. This is because statically−configured IP addresses on a virtual−template interface are never cloned to a virtual−access interface. Moreover, an interface without an IP address configured is not able to establish any dynamic routing protocol adjacency. The ***ip unnumbered*** command is not necessary for SVTI, but without that subnet, verification is performed when dynamic routing protocol adjacency is established. Also the ***ipv6 unnumbered*** command is not needed for IPV6 scenarios because of the link−local addresses that are used in order to build EIGRP adjacencies.

# Prerequisites

## Requirements

Cisco recommends that you have basic knowledge of these topics:

- VPN Configuration on Cisco IOS
- FlexVPN Configuration on Cisco IOS

## Components Used

The information in this document is based on Cisco IOS Version 15.3T.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

# EIGRP on One Ethernet Segment with Different Subnets

***Topology:*** Router 1 (R1) (e0/0: 10.0.0.1/24)−−−−−−−(e0/1: 10.0.1.2/24) Router 2 (R2)

***R1:***

```
interface Ethernet0/0
 ip address 10.0.0.1 255.255.255.0

router eigrp 100
 network 10.0.0.1 0.0.0.0
```

***R2:***

```
interface Ethernet0/0
 ip address 10.0.1.2 255.255.255.0

router eigrp 100
 network 10.0.1.2 0.0.0.0
```

R1 shows:

```
*Mar 3 16:39:34.873: EIGRP: Received HELLO on Ethernet0/0 nbr 10.0.1.2
*Mar 3 16:39:34.873:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0
*Mar 3 16:39:34.873: EIGRP−IPv4(100): Neighbor 10.0.1.2 not on common subnet
for Ethernet0/0
```

Cisco IOS does not form an adjacency, which is expected. For more information about this, refer to the What Do EIGRP "Not On Common Subnet" Messages Mean? article.

# EIGRP on SVTI Segment with Different Subnets

The same situation occurs when you use Virtual Tunnel Interfaces (VTI) (Generic Routing Encapsulation (GRE) Tunnel).

***Topology:*** R1(Tun1: 172.16.0.1/24)−−−−−−−(Tun1: 172.17.0.2/24) R2

***R1:***

```
interface Ethernet0/0
 ip address 10.0.0.1 255.255.255.0

interface Tunnel1
 ip address 172.16.0.1 255.255.255.0
 tunnel source Ethernet0/0
 tunnel destination 10.0.0.2

router eigrp 100
 network 172.16.0.1 0.0.0.0
 passive-interface default
 no passive-interface Tunnel1
```

***R2:***

```
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0

interface Tunnel1
 ip address 172.17.0.2 255.255.255.0
 tunnel source Ethernet0/0
 tunnel destination 10.0.0.1

router eigrp 100
 network 172.17.0.2 0.0.0.0
 passive-interface default
 no passive-interface Tunnel1
```

R1 shows:

```
*Mar  3 16:41:52.167: EIGRP: Received HELLO on Tunnel1 nbr 172.17.0.2
*Mar  3 16:41:52.167:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0
*Mar  3 16:41:52.167: EIGRP-IPv4(100): Neighbor 172.17.0.2 not on common subnet
for Tunnel1
```

This is expected behavior.

## Use the IP Unnumbered Command

This example shows how to use the ***ip unnumbered*** command that disables verification and allows for establishment of an EIGRP session between peers in different subnets.

The topology is similar to the previous example, but the addresses of the tunnels are now defined via the ***ip unnumbered*** command that points to loopbacks:

***Topology***: R1(Tun1: 172.16.0.1/24)−−−−−−−(Tun1: 172.17.0.2/24) R2

***R1:***

```
interface Ethernet0/0
 ip address 10.0.0.1 255.255.255.0

interface Loopback0
```

```
 ip address 172.16.0.1 255.255.255.0

interface Tunnel1
 ip unnumbered Loopback0
 tunnel source Ethernet0/0
 tunnel destination 10.0.0.2

router eigrp 100
 network 172.16.0.1 0.0.0.0
 passive-interface default
 no passive-interface Tunnel1

R2:
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0

interface Loopback0
 ip address 172.17.0.2 255.255.255.0

interface Tunnel1
 ip unnumbered Loopback0
 tunnel source Ethernet0/0
 tunnel destination 10.0.0.1

router eigrp 100
 network 172.17.0.2 0.0.0.0
 passive-interface default
 no passive-interface Tunnel1
```

R1 shows:

```
*Mar  3 16:50:39.046: EIGRP: Received HELLO on Tunnel1 nbr 172.17.0.2
*Mar  3 16:50:39.046:   AS 100, Flags 0x0:(NULL), Seq 0/0 interfaceQ 0/0
*Mar  3 16:50:39.046: EIGRP: New peer 172.17.0.2
*Mar  3 16:50:39.046: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 172.17.0.2
(Tunnel1) is up: new adjacency

R1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
0   172.17.0.2              Tu1               12 00:00:07    7  1434  0  13

R1#show ip route eigrp
      172.17.0.0/24 is subnetted, 1 subnets
D        172.17.0.0 [90/27008000] via 172.17.0.2, 00:00:05, Tunnel1

R1#show ip int brief
Interface                  IP-Address      OK? Method Status                Protocol
Ethernet0/0                10.0.0.1         YES manual up                    up
Loopback0                  172.16.0.1       YES manual up                    up
Tunnel1                    172.16.0.1       YES TFTP   up                    up
```

R2 is similar to this.

After you change the ***ip unnumbered*** command into a specific IP address configuration, an EIGRP adjacency does not form.

## EIGRP on SVTI to DVTI  Segment with Different Subnets

This example also uses the ip unnumbered command. The rules mentioned previously apply to DVTI as well.

***Topology:*** R1(Tun1: 172.16.0.1/24)−−−−−−−(Virtual−template: 172.17.0.2/24) R2

The previous example is modified here in order to use DVTI instead of SVTI. Additionally, tunnel protection is added in this example.

***R1:***

```
crypto isakmp policy 1
 encr 3des
 authentication pre−share
 group 2
crypto isakmp key cisco address 0.0.0.0 0.0.0.0
!
crypto ipsec transform−set TS esp−des esp−md5−hmac
!
crypto ipsec profile prof
 set transform−set TS
!
interface Loopback0
 ip address 172.16.0.1 255.255.255.0
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel destination 10.0.0.2
 tunnel protection ipsec profile prof
!
router eigrp 100
 network 172.16.0.1 0.0.0.0
 passive−interface default
 no passive−interface Tunnel1
```

***R2:***

```
crypto isakmp policy 1
 encr 3des
 authentication pre−share
 group 2
crypto isakmp key cisco address 0.0.0.0 0.0.0.0
crypto isakmp profile profLAN
   keyring default
   match identity address 10.0.0.1 255.255.255.255
   virtual−template 1
!
crypto ipsec transform−set TS esp−des esp−md5−hmac
!
crypto ipsec profile profLAN
 set transform−set TS
 set isakmp−profile profLAN

interface Loopback0
 ip address 172.17.0.2 255.255.255.0
!
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0
!
interface Virtual−Template1 type tunnel
 ip unnumbered Loopback0
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile profLAN
!
!
router eigrp 100
 network 172.17.0.2 0.0.0.0
 passive−interface default
 no passive-interface Virtual-Template1
```

Everything works as expected:

```
R1#show crypto session
Crypto session current status
Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 10.0.0.2 port 500
  IKEv1 SA: local 10.0.0.1/500 remote 10.0.0.2/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map


R1#show crypto ipsec sa
interface: Tunnel1
    Crypto map tag: Tunnel1-head-0, local addr 10.0.0.1
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   current_peer 10.0.0.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 89, #pkts encrypt: 89, #pkts digest: 89
    #pkts decaps: 91, #pkts decrypt: 91, #pkts verify: 91


R1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
0   172.17.0.2              Tu1               13 00:06:31    7  1434  0  19

R1#show ip route eigrp
      172.17.0.0/24 is subnetted, 1 subnets
D        172.17.0.0 [90/27008000] via 172.17.0.2, 00:06:35, Tunnel1




R2#show crypto session
Crypto session current status
Interface: Virtual-Access1
Profile: profLAN
Session status: UP-ACTIVE
Peer: 10.0.0.1 port 500
  IKEv1 SA: local 10.0.0.2/500 remote 10.0.0.1/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map


R2#show crypto ipsec sa
interface: Virtual-Access1
    Crypto map tag: Virtual-Access1-head-0, local addr 10.0.0.2
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   current_peer 10.0.0.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 107, #pkts encrypt: 107, #pkts digest: 107
    #pkts decaps: 105, #pkts decrypt: 105, #pkts verify: 105


R2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(100)
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
0   172.16.0.1              Vi1               13 00:07:41   11   200  0  16
```

```
R2#show ip route eigrp
      172.16.0.0/24 is subnetted, 1 subnets
D        172.16.0.0 [90/1433600] via 172.16.0.1, 00:07:44, Virtual−Access1
```

As for the previous examples, when you try to configure 172.16.0.1 and 172.17.0.2 directly under the tunnel interfaces, EIGRP fails with exactly the same error as before.

## EIGRP on IKEv2 Flex VPN with Different Subnets

Here is the example for the FlexVPN Hub and Spoke configuration. The server sends the IP address via the configuration mode for the client.

***Topology:*** R1(e0/0: 172.16.0.1/24)−−−−−−−(e0/1: 172.16.0.2/24) R2

Hub (R1) configuration:

```
aaa new−model
aaa authorization network LOCALIKEv2 local

crypto ikev2 authorization policy AUTHOR−POLICY
 pool POOL
!
crypto ikev2 keyring KEYRING
 peer R2
  address 172.16.0.2
  pre−shared−key CISCO
 !

crypto ikev2 profile default
 match identity remote key−id FLEX
 authentication remote pre−share
 authentication local pre−share
 keyring local KEYRING
 aaa authorization group psk list LOCALIKEv2 AUTHOR−POLICY
 virtual−template 1

interface Loopback0
 ip address 1.1.1.1 255.255.255.0
!
interface Ethernet0/0
 ip address 172.16.0.1 255.255.255.0

interface Virtual−Template1 type tunnel
 ip unnumbered Loopback0
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile default
!
!
router eigrp 1
 network 1.1.1.1 0.0.0.0
 passive−interface default
 no passive−interface Virtual−Template1
!
ip local pool POOL 192.168.0.1 192.168.0.10
```

Spoke configuration:

```
aaa new−model
aaa authorization network FLEX local

crypto ikev2 authorization policy FLEX
 route set interface
!
!
!
crypto ikev2 keyring KEYRING
 peer R1
  address 172.16.0.1
  pre−shared−key CISCO
 !
!
!
crypto ikev2 profile default
 match identity remote address 172.16.0.1 255.255.255.255
 identity local key−id FLEX
 authentication remote pre−share
 authentication local pre−share
 keyring local KEYRING
 aaa authorization group psk list FLEX FLEX

interface Loopback0
 ip address 2.2.2.2 255.255.255.0
!
interface Ethernet0/0
 ip address 172.16.0.2 255.255.255.0

interface Tunnel0
 ip address negotiated
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel destination 172.16.0.1
 tunnel protection ipsec profile default

router eigrp 1
 network 0.0.0.0
 passive−interface default
 no passive−interface Tunnel0
```

The Spoke uses SVTI in order to connect to the Hub that uses DVTI for all the spokes. Because EIGRP is not as flexible as Open Shortest Path First (OSPF) and it is not possible to configure it under the interface (SVTI or DVTI), ***network 0.0.0.0*** is used on the Spoke in order to ensure that EIGRP is enabled on the ***Tun0*** interface. A passive interface is used in order to ensure that the adjacency is formed only on the ***Tun0*** interface.

For this deployment, it is also necessary to configure ***ip unnumbered*** on the Hub. When you manually configure an IP address under the virtual−template interface, it is not cloned to the virtual−access interface. Then, the virtual−access interface does not have an IP address assigned, and the EIGRP adjacency does not form. This is why the ***ip unnumbered*** command is always required for DVTI interfaces in order to form an EIGRP adjacency.

In this example, an EIGRP adjacency is built between 1.1.1.1 and 192.168.0.9.

Testing on the Hub:

```
R1#show ip int brief
Interface              IP−Address      OK? Method Status                Protocol
Ethernet0/0            172.16.0.1      YES NVRAM  up                    up
Ethernet0/1            unassigned      YES NVRAM  administratively down down
Ethernet0/2            unassigned      YES NVRAM  administratively down down
Ethernet0/3            unassigned      YES NVRAM  administratively down down
Loopback0              1.1.1.1         YES manual up                    up
```

```
Virtual-Access1        1.1.1.1         YES unset  up                    up
Virtual-Template1      1.1.1.1         YES manual up                    down

R1#show crypto session
Crypto session current status

Interface: Virtual-Access1
Session status: UP-ACTIVE
Peer: 172.16.0.2 port 500
  IKEv2 SA: local 172.16.0.1/500 remote 172.16.0.2/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map

R1#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address             Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                               (sec)         (ms)       Cnt Num
0   192.168.0.9          Vi1                      10 01:28:49   12  1494  0  13

R1#show ip route eigrp
....
Gateway of last resort is not set

      2.0.0.0/24 is subnetted, 1 subnets
D        2.2.2.0 [90/27008000] via 192.168.0.9, 01:28:52, Virtual-Access1
```

From the Spoke perspective, the ***ip address negotiated*** command works the same as the ***ip address unnumbered*** command, and verification of the subnet is disabled.

Testing on the Spoke:

```
R2#show ip int brief
Interface              IP-Address      OK? Method Status                Protocol
Ethernet0/0            172.16.0.2      YES NVRAM  up                    up
Ethernet0/1            unassigned      YES NVRAM  administratively down down
Ethernet0/2            unassigned      YES NVRAM  administratively down down
Ethernet0/3            unassigned      YES NVRAM  administratively down down
Loopback0              2.2.2.2         YES NVRAM  up                    up
Tunnel0                192.168.0.9     YES NVRAM  up                    up

R2#show crypto session
Crypto session current status

Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 172.16.0.1 port 500
  IKEv2 SA: local 172.16.0.2/500 remote 172.16.0.1/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map

R2#show ip eigrp neighbors
EIGRP-IPv4 Neighbors for AS(1)
H   Address             Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                               (sec)         (ms)       Cnt Num
0   1.1.1.1             Tu0                     14 01:30:18   15   1434 0   14

R2#show ip route eigrp
....
      1.0.0.0/24 is subnetted, 1 subnets
D        1.1.1.0 [90/27008000] via 1.1.1.1, 01:30:21
```

## Configuration Mode for Routing

Internet Key Exchange version 2 (IKEv2) is another option. It is possible to use the configuration mode in order to push routes. In this scenario, EIGRP and the ***ip unnumbered*** command are not needed.

You can modify the previous example in order to configure the Hub to send that route via configuration mode:

```
crypto ikev2 authorization policy AUTHOR−POLICY
 pool POOL
 route set access−list SPLIT

ip access−list standard SPLIT
 permit 1.1.1.0 0.0.0.255
```

The Spoke sees 1.1.1.1 as static, not EIGRP:

```
R2#show ip route
....
      1.0.0.0/24 is subnetted, 1 subnets
S        1.1.1.0 is directly connected, Tunnel0
```

The same process works in the opposite direction. The Spoke sends a route to the Hub:

```
crypto ikev2 authorization policy FLEX
 route set access−list SPLIT

ip access−list standard SPLIT
 permit 2.2.2.0 0.0.0.255
```

The Hub sees it as static (not EIGRP):

```
R1#show ip route
....
      2.0.0.0/24 is subnetted, 1 subnets
S        2.2.2.0 is directly connected, Virtual−Access1
```

For this scenario, the dynamic routing protocol and the ***ip unnumbered*** command are not needed.

# IPV6 EIGRP on SVTI Segment with Different Subnets

For IPv6, the situation is different. This is because IPv6 link−local addresses (FE80::/10) are used in order to build EIGRP or OSPF adjacency. Valid link−local addresses always belong to the same subnet, so there is no need to use the ***ipv6 unnumbered*** command for that.

The topology here is the same as for the previous example, except that all IPv4 addresses are replaced with IPv6 addresses.

R1 configuration:

```
interface Tunnel1
 no ip address
 ipv6 address FE80:1::1 link−local
 ipv6 address 2001:1::1/64
 ipv6 enable
 ipv6 eigrp 100
 tunnel source Ethernet0/0
 tunnel mode gre ipv6
 tunnel destination 2001::2

interface Loopback0
 description Simulate LAN
 no ip address
 ipv6 address 2001:100::1/64
 ipv6 enable
 ipv6 eigrp 100

interface Ethernet0/0
 no ip address
 ipv6 address 2001::1/64
 ipv6 enable

ipv6 router eigrp 100
```

R2 configuration:

```
interface Tunnel1
 no ip address
 ipv6 address FE80:2::2 link−local
 ipv6 address 2001:2::2/64
 ipv6 enable
 ipv6 eigrp 100
 tunnel source Ethernet0/0
 tunnel mode gre ipv6
 tunnel destination 2001::1

interface Loopback0
 description Simulate LAN
 no ip address
 ipv6 address 2001:200::1/64
 ipv6 enable
 ipv6 eigrp 100

interface Ethernet0/0
 no ip address
 ipv6 address 2001::2/64
 ipv6 enable

ipv6 router eigrp 100
```

The tunnel addresses are in different subnets (2001:1::1/64 and 2001:2::2/64), but that is not important. Link−local addresses are used in order to build adjacency. With these addresses, it always succeeds.

On R1:

```
R1#show ipv6 int brief
Ethernet0/0            [up/up]
    FE80::A8BB:CCFF:FE00:6400
    2001::1
Loopback0              [up/up]
    FE80::A8BB:CCFF:FE00:6400
    2001:100::1
Tunnel1                [up/up]
    FE80:1::1
    2001:1::1

R1#show ipv6 eigrp neighbors
EIGRP−IPv6 Neighbors for AS(100)
H   Address             Interface              Hold Uptime   SRTT   RTO  Q  Seq
```

```
                                               (sec)         (ms)       Cnt Num
0   Link-local address: Tu1                      12 00:13:58  821  4926  0  17
    FE80:2::2

R1#show ipv6 route eigrp
...
D   2001:2::/64 [90/28160000]
     via FE80:2::2, Tunnel1
D   2001:200::/64 [90/27008000]
     via FE80:2::2, Tunnel1
```

On R2:

```
R2#show ipv6 int brief
Ethernet0/0            [up/up]
    FE80::A8BB:CCFF:FE00:6500
    2001::2
Loopback0              [up/up]
    FE80::A8BB:CCFF:FE00:6500
    2001:200::1
Tunnel1                [up/up]
    FE80:2::2
    2001:2::2

R2#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(100)
H   Address             Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                               (sec)         (ms)       Cnt Num
0   Link-local address: Tu1                      14 00:15:31   21  1470  0  18
    FE80:1::1

R2#show ipv6 route eigrp
...
D   2001:1::/64 [90/28160000]
     via FE80:1::1, Tunnel1
D   2001:100::/64 [90/27008000]
     via FE80:1::1, Tunnel1
```

The peer IPv6 network is installed by the EIGRP process. On R1, the 2001:2::/64 network is installed, and that network is a different subnet than 2001:1::/64. The same is true on R2. For example, 2001::1/64 is installed, which is a subnet for its peer IP address. There is no need for the ***ipv6 unnumbered*** command here. Additionally, the ***ipv6 address*** command is not needed on the tunnel interface in order to establish EIGRP adjacency, because link−local addresses are used (and those are generated automatically when you enable IPv6 with the ***ipv6 enable*** command).

## IPV6 EIGRP on IKEv2 Flex VPN with Different Subnets

DVTI configuration for IPv6 is different than for IPv4: it is not possible to configure a static IP address anymore.

```
R1(config)#interface Virtual-Template2 type tunnel
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 address ?
  autoconfig  Obtain address using autoconfiguration
  dhcp        Obtain a ipv6 address using dhcp
  negotiated  IPv6 Address negotiated via IKEv2 Modeconfig

R1(config-if)#ipv6 address
```

This is expected, since a static address is never cloned to a virtual−access interface. This is why the ***ipv6***

***unnumbered*** command is recommended for Hub configuration, and the ***ipv6 address negotiated*** command is recommended for Spoke configuration.

The topology is the same as the previous example, except that all IPv4 addresses are replaced with IPv6 addresses.

Hub (R1) configuration:

```
aaa authorization network LOCALIKEv2 local

crypto ikev2 authorization policy AUTHOR-POLICY
 ipv6 pool POOL

crypto ikev2 keyring KEYRING
 peer R2
  address 2001::2/64
  pre-shared-key CISCO

crypto ikev2 profile default
 match identity remote key-id FLEX
 authentication remote pre-share
 authentication local pre-share
 keyring local KEYRING
 aaa authorization group psk list LOCALIKEv2 AUTHOR-POLICY
 virtual-template 1

interface Loopback0
 no ip address
 ipv6 address 2001:100::1/64
 ipv6 enable
 ipv6 eigrp 100

interface Ethernet0/0
 no ip address
 ipv6 address 2001::1/64
 ipv6 enable

interface Virtual-Template1 type tunnel
 no ip address
 ipv6 unnumbered Loopback0
 ipv6 enable
 ipv6 eigrp 100
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv6
 tunnel protection ipsec profile default

ipv6 local pool POOL 2001:10::/64 64
ipv6 router eigrp 100
 eigrp router-id 1.1.1.1
```

Spoke (R2) configuration:

```
aaa authorization network FLEX local

crypto ikev2 authorization policy FLEX
 route set interface

crypto ikev2 keyring KEYRING
 peer R1
  address 2001::1/64
  pre-shared-key CISCO

crypto ikev2 profile default
```

```
 match identity remote address 2001::1/64
 identity local key-id FLEX
 authentication remote pre-share
 authentication local pre-share
 keyring local KEYRING
 aaa authorization group psk list FLEX FLEX

interface Tunnel0
 no ip address
 ipv6 address negotiated
 ipv6 enable
 ipv6 eigrp 100
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv6
 tunnel destination 2001::1
 tunnel protection ipsec profile default
!
interface Ethernet0/0
 no ip address
 ipv6 address 2001::2/64
 ipv6 enable

ipv6 router eigrp 100
 eigrp router-id 2.2.2.2
```

Verification:

```
R2#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(100)
H   Address             Interface              Hold Uptime   SRTT   RTO  Q  Seq
                                               (sec)         (ms)       Cnt Num
0   Link-local address: Tu0                      11 00:12:32   17  1440  0  12
    FE80::A8BB:CCFF:FE00:6500

R2#show ipv6 route eigrp
....
D   2001:100::/64 [90/27008000]
     via FE80::A8BB:CCFF:FE00:6500, Tunnel0

R2#show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Tunnel0
Uptime: 00:13:17
Session status: UP-ACTIVE
Peer: 2001::1 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 2001::1
      Desc: (none)
  IKEv2 SA: local 2001::2/500
          remote 2001::1/500 Active
          Capabilities:(none) connid:1 lifetime:23:46:43
  IPSEC FLOW: permit ipv6 ::/0 ::/0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 190 drop 0 life (KB/Sec) 4271090/2803
        Outbound: #pkts enc'ed 194 drop 0 life (KB/Sec) 4271096/2803

R2#ping 2001:100::1 repeat 100
Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 2001:100::1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 1/4/5 ms
```

```
R2#show crypto session detail
Crypto session current status

Code: C − IKE Configuration mode, D − Dead Peer Detection
K − Keepalives, N − NAT−traversal, T − cTCP encapsulation
X − IKE Extended Authentication, F − IKE Fragmentation

Interface: Tunnel0
Uptime: 00:13:27
Session status: UP−ACTIVE
Peer: 2001::1 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 2001::1
      Desc: (none)
  IKEv2 SA: local 2001::2/500
          remote 2001::1/500 Active
          Capabilities:(none) connid:1 lifetime:23:46:33
  IPSEC FLOW: permit ipv6 ::/0 ::/0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 292 drop 0 life (KB/Sec) 4271071/2792
        Outbound: #pkts enc'ed 296 drop 0 life (KB/Sec) 4271082/2792
```

For DVTI, IPv6 cannot be configured manually. The ***ipv6 unnumbered*** command is recommended for the Hub, and the ***ipv6 address negotiated*** command is recommended on the Spoke.

This scenario presents the ***ipv6 unnumbered*** command for DVTI. It is important to notice that, for IPv6 as opposed to IPv4, the ***ipv6 unnumbered*** command on the virtual−template interface is not needed. The reason for this is the same as for the IPv6 SVTI scenario: the link−local ipv6 address is used for building adjacency. The virtual−Access interface, which is cloned from the virtual−template, inherits the IPv6 link−local address, and that is sufficient in order to build EIGRP adjacency.

# Verify

There is currently no verification procedure available for this configuration.

# Troubleshoot

There is currently no specific troubleshooting information available for this configuration.

# Known Caveats

Cisco bug ID CSCtx45062 FlexVPN: Eigrp should not check common subnets if tunnel ip's are /32.

This bug and fix is not FlexVPN−specific. Enter this command before you implement the fix (Software Release 15.1):

```
R2(config−if)#do show run int tun1
Building configuration...

Current configuration : 165 bytes

interface Tunnel1
 tunnel source Ethernet0/0
 tunnel destination 192.168.0.1
 tunnel protection ipsec profile prof1

R2(config−if)#ip address 192.168.200.1 255.255.255.255
Bad mask /32 for address 192.168.200.1
```

Enter this command after the fix (software 15.3):

```
R2(config-if)#do show run int tun1
Building configuration...

Current configuration : 165 bytes

interface Tunnel1
 tunnel source Ethernet0/0
 tunnel destination 192.168.0.1
 tunnel protection ipsec profile prof1

R2(config-if)#ip address 192.168.200.1 255.255.255.255
R2(config-if)#
*Jun 14 18:01:12.395: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor
192.168.100.1 (Tunnel1) is up: new adjacency
```

There are actually two changes in Software Release 15.3:

- ***Netmask /32*** is accepted for all IP addresses.
- There is no subnet verification for an EIGRP neighbor when you use the ***/32*** address.

# Summary

EIGRP behavior is changed by the ***ip unnumbered*** command. It disables checks for the same subnet while it establishes an EIGRP adjacency.

It is also important to remember that when you use DVTIs statically configured IP address on the virtual−template, it is not cloned to the virtual−access. This is why the ***ip unnumbered*** command is needed.

For FlexVPN, there is no need to use the ***ip unnumbered*** command when you use the negotiated address on the client. But, it is important to use it on the Hub when you use EIGRP. When you use the configuration mode for routing, EIGRP is not needed.

For SVTI, IPv6 uses link−local addresses for adjacency, and there is no need to use the ***ipv6 unnumbered*** command.

For DVTI, IPv6 cannot be configured manually. The ***ipv6 unnumbered*** command is recommended for the Hub, and the ***ipv6 address negotiated*** command is recommended on the Spoke.

# Related information

- ***Cisco IOS 15.3 FlexVPN configuration guide***
- ***Cisco IOS 15.3 Command References***
- ***Technical Support & Documentation − Cisco Systems***

Updated: Sep 18, 2013 Document ID: 116346