

Direct Transport VPN Configuration

Michael J. Martin

Previous articles in this series on IPsec VPN configuration using Cisco routers covered building a VPN gateway and clients for client-to site connectivity as well as Cisco's EzVPN to support network-to-network VPN topologies. In this article, we will look at options for implementing static network-to-network IPsec links, or direct transport VPNs.

Static network-to-network IPsec policies are used to provide permanent, secure network connections between private networks, utilizing the unsecured public network for secure data transport. In previous articles we examined dynamically established split-tunneling and full-crypto IPsec policy configurations. Similar structural parallels are evident in static IPsec VPN configurations used to implement network-to-network security topologies. However, with dynamic client configurations the differences in policy structure denote what traffic is secured between the client and the IPsec gateway. With static IPsec policies the operational focus of the policy shifts to the "reachability" of the networks that need to exchange secured data. Through this lens, using static IPsec crypto policies on Cisco routers can be implemented using one of two approaches: direct and indirect transport VPNs.

Direct vs. Indirect Transport VPNs

The direct transport approach is used to secure directly routable IP traffic exchanged between networks. The key operating element of a direct transport VPN is that the exchange of data relies on a route reachability policy that is independent of the IPsec security policy. This is similar in structure to the client VPN split-tunnel policy, in which specific network-to-network security policies are defined on each of the IPsec peers. These static IPsec policies then secure specific IP data communications exchanges between specific networks as the traffic passes through IPsec peered gateway or transit routers. The big advantages to using direct VPNs are twofold. First, they are more efficient than indirect VPNs, adding only 28 bytes of additional packet overhead. Second, the secure and unsecured traffic flows are managed by a single routing policy.

While the direct approach allows for selective encryption of direct traffic exchanges, the indirect transport VPN approach uses a virtual interface to tunnel network-to-network data exchanges. Indirect VPN implementations operate using a full-crypto model. The tunneling protocol enables a virtual link to be established between two public router endpoints. IPsec peering is then established between the tunnel endpoint routers and the tunneling protocol traffic is then encrypted using an IPsec security policy.

That is where indirect VPNs outshine the direct VPN approach. They enable networks that do not have any direct IP reachability the capacity to exchange data securely, using a combination of a tunneling protocol and IPsec. This functionality does come with a price, however. Indirect VPN solutions add more than 60 bytes of overhead to each packet and the additional headache of having to manage both private and public routing policies.

Direct transport VPN Configuration

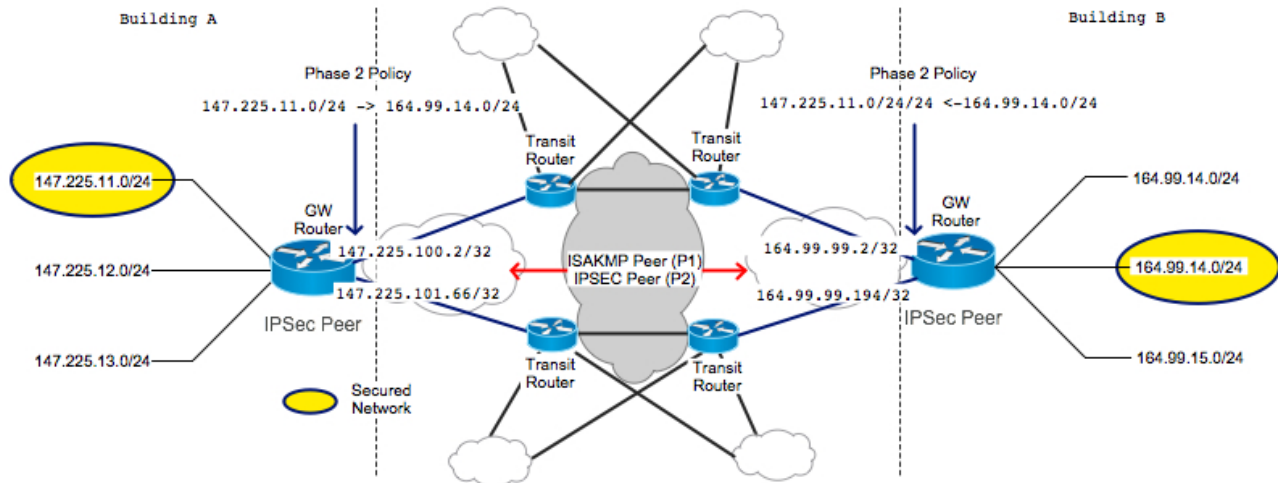
Direct transport VPNs are a good match in any scenario where additional security needs to be overlaid on existing IP network infrastructure. They could include the following:

- Augmenting application layer transaction security or network-specific data exchanges in a private campus or enterprise environment

Direct Transport VPN Configuration

Michael J. Martin

- Securing private data exchanges over carrier provided packet/cell/frame (MPLS) transport
- As a WAN transport alternative to connect a remote location to a network hub using a single connectivity path.



For example, let's say you have a large campus network, with two new research teams that need to securely exchange data, one at Building A and one at Building D. The diagram above illustrates how that could be diagrammed and will form the basis for our first configuration example. When building a direct transport VPN solution, one of the first things to consider is the degree of security required. In the illustration, the IPsec policy could be installed either on the gateway routers or the transit routers. If the policy requires LAN-to-LAN security, then the IPsec policy must be implemented on the GW router. Alternatively, if we are only concerned with encrypting the traffic as it leaves the "LAN" we could implement the policies on the transit routers. Both the "gateway" and "transit" scenarios have a degree of complexity in terms of configuration. Both options require IPsec/ISAKMP peer definitions for each path.

For our configuration example, we will secure the traffic on the gateway router. The IPsec configuration process has six steps. The Building A configuration is in blue, and the Building D configuration is in red.

Step 1. Define ISAKMP (Phase 1) policy:

```
build-a-gw1(config)#crypto isakmp policy 10
build-a-gw1(config-isakmp)# encr 3des
build-a-gw1(config-isakmp)# authentication pre-share
build-a-gw1(config-isakmp)# group 2
build-a-gw1(config-isakmp)# lifetime 3600
```

```
build-d-gw1(config)#crypto isakmp policy 10
build-d-gw1(config-isakmp)# encr 3des
build-d-gw1(config-isakmp)# authentication pre-share
build-d-gw1(config-isakmp)# group 2
build-d-gw1(config-isakmp)# lifetime 3600
```

Direct Transport VPN Configuration

Michael J. Martin

Step 2. Define ISAKMP policy pre-shared keys:

```
build-a-gw1(config)# crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0
build-a-gw1(config)# crypto isakmp keepalive 30
```

```
build-d-gw1(config)# crypto isakmp key cisco123
address 0.0.0.0 0.0.0.0
build-d-gw1(config)# crypto isakmp keepalive 30
```

Step 3. Build IPsec (Phase 2) transform set:

```
build-a-gw1(config)#crypto IPsec transform-set direct-trans-vpn
esp-3des esp-md5-hmac
```

```
build-d-gw1(config)#crypto IPsec transform-set direct-trans-vpn
esp-3des esp-md5-hmac
```

Step 4. Build traffic qualifier access list:

```
build-a-gw1(config)#ip access-list extended CRYPT
build-a-gw1(config-ext-nacl)#permit ip 172.30.69.0 0.0.0.255
172.30.41.0 0.0.0.255
```

```
build-d-gw1(config)#ip access-list extended CRYPT
build-d-gw1(config-ext-nacl)#permit ip 172.30.41.0 0.0.0.255
172.30.69.0 0.0.0.255
```

Step 5. Define IPsec static crypto map:

```
build-a-gw1(config)#crypto map dir-trans-vpn-buld-a 10 IPsec-isakmp
build-a-gw1(config-crypto-map)#set transform-set direct-trans-vpn
build-a-gw1(config-crypto-map)# match address CRYPT
build-a-gw1(config-crypto-map)#set peer 164.99.99.2
build-a-gw1(config-crypto-map)#set peer 164.99.99.194
```

```
build-d-gw1(config)#crypto map dir-trans-vpn-buld-d 10 IPsec-isakmp
build-d-gw1(config-crypto-map)#set transform-set gre-vpn-transform
build-d-gw1(config-crypto-map)#match address CRYPT
build-d-gw1(config-crypto-map)#set peer 147.225.100.2
build-d-gw1(config-crypto-map)#set peer 147.225.101.66
```

Step 6. Install static crypto map:

```
build-a-gw1(config)#interface FastEthernet 0/0
build-d-gw1(config-if)#crypto map dir-trans-vpn-buld-a
```

```
build-a-gw1(config)#interface FastEthernet 0/1
build-a-gw1(config-if)#crypto map dir-trans-vpn-buld-a
```

```
build-d-gw1(config)#interface FastEthernet 0/0
build-d-gw1(config-if)#crypto map dir-trans-vpn-buld-d
```

```
build-d-gw1(config)#interface FastEthernet 0/1
build-d-gw1(config-if)#crypto map dir-trans-vpn-buld-d
```

Once the configuration is built and installed, there are two commands you can use to verify operation.

Direct Transport VPN Configuration

Michael J. Martin

The first command, <show crypto isakmp sa>, is used to verify the state of ISAKMP on the router:

```
build-a-gw1#sh crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state      conn-id slot status
147.225.100.2 164.99.99.2 QM_IDLE    2073    0 ACTIVE
```

```
build-d-gw1# sh crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst          src          state      conn-id slot status
164.99.99.2  147.225.100.2 QM_IDLE    2071    0 ACTIVE
```

According to Cisco's documentation there are eight ISAKMP states:

- MM_NO_STATE: The Internet Security Association and Key Management Protocol (ISAKMP) security association (SA) has been created but nothing else has happened yet.
- MM_SA_SETUP: The peers have agreed on parameters for the ISAKMP SA.
- MM_KEY_EXCH: The peers have exchanged Diffie-Hellman public keys and have generated a shared secret. The ISAKMP SA remains unauthenticated.
- MM_KEY_AUTH: The ISAKMP SA has been authenticated. If the router initiated this exchange, this state transitions immediately to QM_IDLE and a Quick Mode exchange begins.
- AG_NO_STATE: The ISAKMP SA has been created but nothing else has happened yet.
- AG_INIT_EXCH: The peers have done the first exchange in Aggressive Mode but the SA is not authenticated.
- AG_AUTH: The ISAKMP SA has been authenticated. If the router initiated this exchange, this state transitions immediately to QM_IDLE and a Quick Mode exchange begins.
- QM_IDLE: The ISAKMP SA is idle. It remains authenticated with its peer and may be used for subsequent Quick Mode exchanges.

A properly configured and active ISAKMP peer will be in QM_IDLE state. If the ISAKMP state of a peer is other than QM_IDLE, there is either an error in the configuration or the router is establishing the ISAKMP peering. The other handy command is <show crypto IPsec sa>:

```
build-a-gw1#show crypto IPsec sa

interface: FastEthernet0/0
  Crypto map tag: dir-trans-vpn-buld-a , local addr 147.225.100.2

  protected vrf: (none)
  local  ident (addr/mask/prot/port): (147.225.11.0/255.255.255.0/0/0)
  remote ident (addr/mask/prot/port): (164.99.14.0//255.255.255.0/0/0)
```

Direct Transport VPN Configuration

Michael J. Martin

```
current_peer 164.99.99.2 port 500
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 1711, #pkts encrypt: 1711, #pkts digest: 1711
  #pkts decaps: 291, #pkts decrypt: 291, #pkts verify: 291
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0
  #pkts not decompressed: 0, #pkts decompress failed: 0
  #send errors 2, #recv errors 0
```

```
local crypto endpt.: 147.225.100.2, remote crypto
endpt.: 164.99.99.2
path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0
current outbound spi: 0x0(0)
```

```
inbound esp sas:
spi: 0x9DD44A4(165495972)
  transform: esp-3des esp-md5-hmac ,
  in use settings ={Tunnel, }
  conn id: 141, flow_id: AIM-VPN/SSL-1:141,
crypto map: dir-trans-vpn-buld-a
  sa timing: remaining key lifetime (k/sec): (4545643/2916)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE
```

inbound ah sas:

inbound pcp sas:

```
outbound esp sas:
spi: 0xDD5316D7(3713210071)
  transform: esp-3des esp-md5-hmac ,
  in use settings ={Tunnel, }
  conn id: 142, flow_id: AIM-VPN/SSL-1:142,
crypto map: dir-trans-vpn-buld-a
  sa timing: remaining key lifetime (k/sec): (4545643/2916)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE
```

outbound ah sas:

outbound pcp sas:

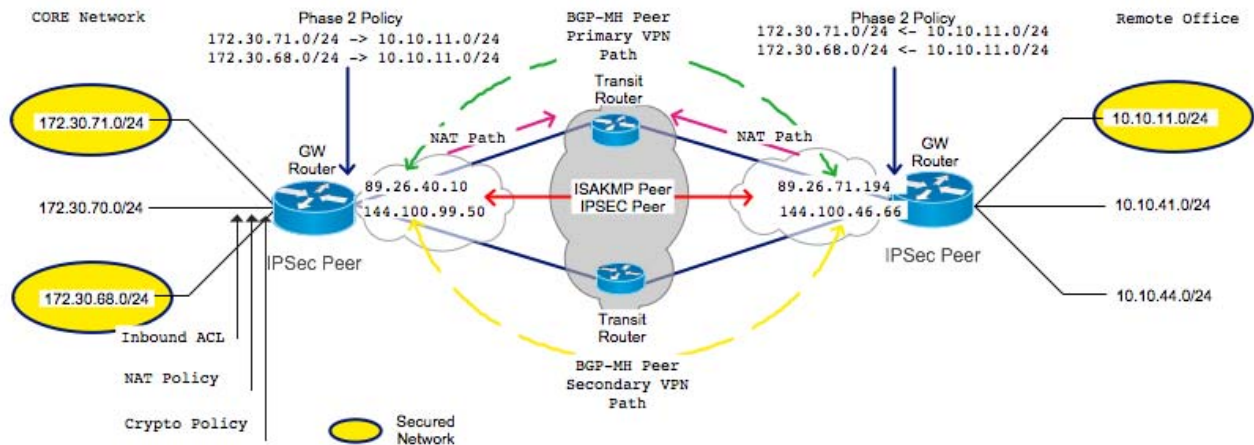
The command generates two data elements for each Phase 2 peering relationship, grouped by policy interface. The first element provides the encryption, compression, and error statistics for the crypto peering policy. The second element details the lifetime and mode information for the active inbound and outbound SAs associated with the policy. This command provides you with the data to see if the policy is actually working as advertised. If you have the ISAKMP peers successfully established but are not seeing the policy statistic counters increment, check your traffic qualifier access list. It's more than likely that there is an error in your traffic-matching logic.

Direct Transport VPN Configuration

Michael J. Martin

Hybrid Direct VPN Configuration

Our second configuration example is a hybrid of the direct and indirect VPN models. Like a direct VPN implementation, it uses only IPsec to transport data between secured networks. But like an indirect VPN implementation, this solution can be used to provide secure data exchange between networks that are non-routable or are not reachable without the aid of the security policy.



The diagram above illustrates our configuration scenario. In this example, we are providing Internet connectivity for all networks at the core and remote locations and enabling secure data exchange between some of the networks at each location. In this case, each location is using RFC-1918 private Internet addressing, so NAT translation will be used to provide Internet access. Like our first example, each site has two links terminating on a single gateway router, but in this case they are two different ISP links, both providing a default route path. One of the links is 10 Mbps and is used as the primary traffic path; the other is a 2 Mbps (downlink), 512Kbps (uplink) ADSL link used for VPN backup.

To provide dynamic path failover, a dynamic routing protocol must be employed. Because RFC-1918 addressing is being used and there is no direct network connectivity to support an Interior Gateway Protocol (IGP), we will use Border Gateway Protocol (BGP) multihop to exchange routing information and set path preference. A distribute list is used to filter the BGP routing announcements, and the Cisco proprietary BGP metric weight is used to set path preference.

From a packet processing perspective, all of the LAN traffic passes through the NAT policy prior to the IPsec policy. So we'll use an explicit NAT traffic-matching access list to pass the traffic that must be encrypted to the IPsec policy. Once the traffic is qualified by the IPsec policy, the "non-routable" IP packets are encapsulated in an ESP frame that sources from the public interface of the IPsec peer gateway router to the public destination address of the other peer router. This configuration involves 11 steps that are outlined below. The core data center configuration is in green, and the remote office configuration is in orange.

Step 1. Define ISAKMP (Phase 1) policy:

```
CORE-VPN-GW(config)#crypto isakmp policy 10
CORE-VPN-GW(config-isakmp)# encr 3des
CORE-VPN-GW(config-isakmp)# authentication pre-share
```

Direct Transport VPN Configuration

Michael J. Martin

```
CORE-VPN-GW(config-isakmp)# group 2
CORE-VPN-GW(config-isakmp)# lifetime 3600
```

```
REMOTE-VPN-GW(config)#crypto isakmp policy 10
REMOTE-VPN-GW(config-isakmp)# encr 3des
REMOTE-VPN-GW(config-isakmp)# authentication pre-share
REMOTE-VPN-GW(config-isakmp)# group 2
REMOTE-VPN-GW(config-isakmp)# lifetime 3600
```

Step 2. Define ISAKMP policy pre-shared keys:

```
CORE-VPN-GW(config)# crypto isakmp key cisco123
address 0.0.0.0 0.0.0.0
```

```
REMOTE-VPN-GW(config)# crypto isakmp key cisco123
address 0.0.0.0 0.0.0.0
```

Step 3. Build IPsec (Phase 2) transform set:

```
CORE-VPN-GW(config)#crypto IPsec transform-set
CRT-ESP-COMP esp-3des esp-md5-hmac comp-lzs
```

```
REMOTE-VPN-GW(config)#crypto IPsec transform-set
RRT-ESP-COMP esp-3des esp-md5-hmac comp-lzs
```

Step 4. Build traffic-qualifier access list:

```
CORE-VPN-GW(config)#
CORE-VPN-GW(config-crypto-map)#permit ip 172.30.71.0 0.0.0.255
10.10.11.0 0.0.0.255
CORE-VPN-GW(config-crypto-map)#permit ip 172.30.68.0 0.0.0.255
10.10.11.0 0.0.0.255
```

```
REMOTE-VPN-GW(config)#
REMOTE-VPN-GW(config-crypto-map)#permit ip 10.10.11.0 0.0.0.255
172.30.71.0 0.0.0.255
REMOTE-VPN-GW(config-crypto-map)#permit ip 10.10.11.0 0.0.0.255
172.30.68.0 0.0.0.255
```

Step 5. Build NAT-qualifier access list:

```
CORE-VPN-GW(config)#ip access-list extended NAT
CORE-VPN-GW(config-ext-nacl)#deny ip 172.30.68.0 0.0.0.255 10.10.11.0
0.0.0.255
CORE-VPN-GW(config-ext-nacl)#deny ip 172.30.71.0 0.0.0.255 10.10.11.0
0.0.0.255
CORE-VPN-GW(config-ext-nacl)#permit ip 172.30.68.0 0.0.0.255 any
CORE-VPN-GW(config-ext-nacl)#permit ip 172.30.71.0 0.0.0.255 any
CORE-VPN-GW(config-ext-nacl)#permit ip 172.30.70.0 0.0.0.255 any
```

```
REMOTE-VPN-GW(config)#ip access-list extended NAT
REMOTE-VPN-GW(config-ext-nacl)#deny ip 10.10.11.0 0.0.0.255
172.30.68.0 0.0.0.255
REMOTE-VPN-GW(config-ext-nacl)#deny ip 10.10.11.0 0.0.0.255
172.30.71.0 0.0.0.255
```

Direct Transport VPN Configuration

Michael J. Martin

```
REMOTE-VPN-GW(config-ext-nacl)#permit ip 10.10.11.0 0.0.0.255 any
REMOTE-VPN-GW(config-ext-nacl)#permit ip 10.10.41.0 0.0.0.255 any
REMOTE-VPN-GW(config-ext-nacl)#permit ip 10.10.44.0 0.0.0.255 any
```

Step 6. Build NAT policy:

```
CORE-VPN-GW(config)#ip nat inside source list NAT interface
FastEthernet 0/0
```

```
REMOTE-VPN-GW(config)#ip nat inside source list NAT interface
FastEthernet 0/0
```

Step 7. Install NAT configuration:

```
CORE-VPN-GW(config)#int fa 0/0
CORE-VPN-GW(config-if)#ip nat outside
CORE-VPN-GW(config)#int fa 0/1
CORE-VPN-GW(config-if)#ip nat inside
```

```
REMOTE-VPN-GW(config)#int fa 0/0
REMOTE-VPN-GW(config-if)#ip nat outside
REMOTE-VPN-GW(config)#int fa 0/1
REMOTE-VPN-GW(config-if)#ip nat inside
```

Step 8. Build static crypto map:

```
CORE-VPN-GW(config)#crypto map CORE-VPN 10 IPsec-isakmp
CORE-VPN-GW(config-crypto-map)#set transform-set CRT-ESP-COMP
CORE-VPN-GW(config-crypto-map)#match address CRYPT
CORE-VPN-GW(config-crypto-map)#set peer 89.26.71.194
CORE-VPN-GW(config-crypto-map)#set peer 144.100.46.66
```

```
REMOTE-VPN-GW(config)#crypto map REM-VPN 10 IPsec-isakmp
REMOTE-VPN-GW(config-crypto-map)#set transform-set RRT-ESP-COMP
REMOTE-VPN-GW(config-crypto-map)#match address CRYPT
REMOTE-VPN-GW(config-crypto-map)#set peer 89.26.40.10
REMOTE-VPN-GW(config-crypto-map)#set peer 144.100.99.50
```

Step 9. Install static crypto map:

```
CORE-VPN-GW(config)#interface FastEthernet0/0
CORE-VPN-GW(config-if)#description primary ISP path 10Mbit/s
CORE-VPN-GW(config-if)#crypto map CORE-VPN
CORE-VPN-GW(config)#interface FastEthernet0/1
CORE-VPN-GW(config)#description secondary ISP path 2/512 ADSL
CORE-VPN-GW(config-if)#crypto map CORE-VPN
```

```
REMOTE-VPN-GW(config)#interface FastEthernet0/0
REMOTE-VPN-GW(config-if)#description primary ISP path 10Mbit/s
REMOTE-VPN-GW(config-if)#crypto map REM-VPN
REMOTE-VPN-GW(config)#interface FastEthernet0/1
REMOTE-VPN-GW(config-if)#description secondary ISP path 2/512 ADSL
REMOTE-VPN-GW(config-if)#crypto map REM-VPN
```

Step 10. Build BGP distribute list access list:

```
CORE-VPN-GW(config)#ip access-list standard VPN-BGP
CORE-VPN-GW(config-std-nacl)#permit 172.30.71.0 0.0.0.255
```

Direct Transport VPN Configuration

Michael J. Martin

```
CORE-VPN-GW(config-std-nacl)#permit 172.30.68.0 0.0.0.255
```

```
REMOTE-VPN-GW(config)#ip access-list standard VPN-BGP
REMOTE-VPN-GW(config-std-nacl)#permit 10.10.11.0 0.0.0.255
```

Step 11. Build BGP policy:

```
CORE-VPN-GW(config)#router bgp 65020
CORE-VPN-GW(config-router)#network 172.30.71.0 mask 255.255.255.0
CORE-VPN-GW(config-router)#network 172.30.68.0 mask 255.255.255.0
CORE-VPN-GW(config-router)#neighbor 89.26.71.194 remote-as 65010
CORE-VPN-GW(config-router)#neighbor 89.26.71.194 ebgp-multihop
CORE-VPN-GW(config-router)#neighbor 89.26.71.194 weight 62000
CORE-VPN-GW(config-router)#neighbor 89.26.71.194 distribute-list
VPN-BGP out
CORE-VPN-GW(config-router)#neighbor 89.26.71.194 soft-
reconfiguration inbound
CORE-VPN-GW(config-router)#neighbor 144.100.46.66 remote-as 65010
CORE-VPN-GW(config-router)#neighbor 144.100.46.66 ebgp-multihop
CORE-VPN-GW(config-router)#neighbor 144.100.46.66 weight 62000

CORE-VPN-GW(config-router)#neighbor 144.100.46.66 distribute-list
VPN-BGP out
CORE-VPN-GW(config-router)#neighbor 144.100.46.66 soft-
reconfiguration inbound

REMOTE-VPN-GW(config)#router bgp 65010
REMOTE-VPN-GW(config-router)#network 10.10.11.0 mask 255.255.255.0
REMOTE-VPN-GW(config-router)#neighbor 89.26.40.10 remote-as 65020
REMOTE-VPN-GW(config-router)#neighbor 89.26.40.10 ebgp-multihop
REMOTE-VPN-GW(config-router)#neighbor 89.26.40.10 weight 63000
REMOTE-VPN-GW(config-router)#neighbor 89.26.40.10 soft-reconfiguration i
REMOTE-VPN-GW(config-router)#neighbor 89.26.40.10 distribute-list
VPN-BGP out
REMOTE-VPN-GW(config-router)#neighbor 144.100.99.50 remote-as 65020
REMOTE-VPN-GW(config-router)#neighbor 144.100.99.50 ebgp-multihop
REMOTE-VPN-GW(config-router)#neighbor 144.100.99.50 weight 63000
REMOTE-VPN-GW(config-router)#neighbor 144.100.99.50 soft-
reconfiguration i
REMOTE-VPN-GW(config-router)#neighbor 144.100.99.50 distribute-list
VPN-BGP out
```

Once the BGP process has been established, both of the VPN route candidates will be stored in the router's BGP database. But only one will appear in the routing table with the remote peer router's address listed as the gateway:

```
CORE-VPN-GW#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
```

Direct Transport VPN Configuration

Michael J. Martin

o - ODR, P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

```
10.0.0.0/24 is subnetted, 1 subnets
B    10.10.11.0 [20/0] via 89.26.40.10, 01:04:14
89.0.0.0/30 is subnetted, 1 subnets
C    89.26.40.8 is directly connected, FastEthernet0/0
144.100.0.0/30 is subnetted, 1 subnets
C    144.100.99.48 is directly connected, FastEthernet3/0
172.30.0.0/24 is subnetted, 4 subnets
C    172.30.71.0 is directly connected, FastEthernet2/0
C    172.30.70.0 is directly connected, FastEthernet1/1
C    172.30.68.0 is directly connected, FastEthernet1/2
S*  0.0.0.0/0 is directly connected, FastEthernet0/0
CORE-VPN-GW#
```

While this hybrid direct VPN variation works, the options for managing the IP routing information are limited to either static routes or BGP multihop, where the traditional "direct" configuration relies on routing policy that is independent of the security policy. Routing policy limitations aside, this approach should be more efficient in terms of packet overhead and does not present the packet size and segment size issues that accompany tunnel-based VPNs.

Both of the configuration examples above are very traditional static map-based IPsec configurations. They are secure and efficient, but lack scalability. That is where indirect VPN solutions provide a big advantage over direct VPNs. In our next article, we'll examine indirect VPN configuration.

Now that you can configure a direct transport VPN, you'll want to learn about the indirect type. Go to the main page of this series on Cisco IPsec VPN configuration to read the next article and all the others in the series.