Static Virtual Tunnel Interfaces
January 26, 2011
By Steve Means, CCSP, CCNP

Most engineers who have gone through CCSP certification or used the internet as a WAN via VPN tunnels are familiar with IPsec over GRE. GRE is encrypted by IPsec so that each protocol covers the limitations of the other. Although this allows for dynamic routing protocols to be used across the IPsec tunnel, the tradeoff is additional packet overhead, up to 24 bytes of it.

A less commonly known but much simpler way to allow dynamic routing over IPsec is through the use of static VTI's or virtual tunnel interfaces. Static VTI's also allow for per tunnel QoS, elimination of crypto ACLs and do away with the GRE header overhead issue.

The configuration is almost too easy. On the endpoint devices an isakmp policy must match. A transform set is created and is referenced in an IPsec Profile. A tunnel interface is created with mode ipsec ipv4, it's given an IP address and a source/destination. Finally the IPsec profile is referenced to protect the tunnel.

Assuming the tunnel source and destination is valid; the endpoints will attempt to establish an IPsec tunnel. If the isakmp policies and transform sets match, the tunnel will come up. At this point all that remains is to configure a dynamic routing protocol that advertises both the tunnel subnet and any networks that you want protected by the tunnel.

In our example R1 and R2 will each have a matching isakmp policy including pre-shared key. This is not shown for space savings in this short article. R1 is shown below; R2's config is identical except for the tunnel address, tunnel destination and the networks advertised in EIGRP.

```
crypto ipsec transform-set VTI esp-aes esp-sha-hmac
!
crypto ipsec profile VTI
 set transform-set VTI
!
interface Tunnel0
 ip address 192.168.1.1 255.255.255.0
 tunnel source FastEthernet0/0
 tunnel destination 24.234.2.2
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile VTI
!
router eigrp 150
no auto-summary
network 192.168.1.1
network 1.1.1.1
```

You'll know the configuration is working when the tunnels come up, the EIGRP neighbor relationship comes up, and finally when the network advertised in EIGRP show up in the routing table of the neighbor with a next hop of the tunnel.

```
R2#sho ip route
D       1.1.1.0 [90/297372416] via 192.168.1.1, 00:03:51, Tunnel0
```