Global Information Assurance Certification Paper

Building a Secure Enterprise Grade V^{3}PN

Practical: Track 1 – GIAC Security Essentials (GSEC)
GSEC practical requirements (v1.4b), Option 1

Author: Ian C. Rudy

Abstract

The rising popularity of Virtual Private Networks (VPN) has given birth to a new technology architecture from Cisco Systems that takes these same VPN concepts and encompasses the additional voice and video requirements into a technology called Voice and Video Enabled IPSec VPN or V^3PN. This V^3PN concept is in fact just a combination of Voice over IP (VoIP) technologies, Quality of Service (QoS) technologies, and Security services into one architectural view. The purpose of this paper is to demonstrate a secure, scalable, and redundant V^3PN architecture that can be used as a model for implementation in the Enterprise.

There are many possible deployment scenarios for these various technologies. I will cover one possible solution pattern and make some references to what other possible deployment scenarios might be used within certain requirements.

Business Case

The implementation of any Enterprise scale architecture usually requires the establishment of a solid business case. The International Engineering Consortium has stated that:

> While VPNs offer direct cost savings over other communications methods (e.g., leased lines and long-distance calls), they can also offer other advantages, including indirect cost savings as a result of reduced training requirements and equipment, increased flexibility, and scalability.[1]

The business case for the cost and operationally advantages for VPN already has been established numerous times over the last few years. The challenge in any VPN deployment is to provide the same or better communication services at a reduced cost. The majority of Service Providers establish Service Level Agreements (SLAs) when providing frame relay or leased line type services. The reluctance to establish business critical services over VPN has come from the lack of SLAs for end to end VPN connectivity. This concept is best described in an excerpt from Connected: An Internet Encyclopedia:

> Unreliable delivery has been a mixed blessing for the Internet. It certainly has lived up to its billing for producing a fault-tolerant network, but has created almost as many problems as it has solved.

[1] International Engineering Consortium. "Virtual Private Networks (VPNs)". URL:http://www.iec.org/online/tutorials/vpn/topic02.html (December 6 2003)

> TCP guarantees data delivery, but makes no guarantees about how long that delivery will take. In some applications, such as telephone calls, this is simply unacceptable. If the data arrives too late, it is useless. Worse, TCP will stop everything to ensure retransmission of the lost data, possibly disrupting other data that could have arrived on time. Some Internet protocols, such as ST, have been proposed to address this problem, but none have gained widespread acceptance and all are a far cry from the guaranteed bandwidth of a phone call.[2]

These Service Providers will, however, provide SLA agreements when establishing internet services across their own backbone network. This perceived lack of SLA agreements for VPN services is resolved if two sites are connected on a VPN connection across the Service Providers network. Implementing this model for connecting different sites and establishing connections via two different Service Providers allows for SLA guaranteed delivery and network redundancy. This model also provides the same type of service delivery method Enterprises have become accustomed to with frame relay and leased line services at a fraction of the cost, and allows for the addition of business critical services (e.g., VoIP) to the VPN-based network architecture.

Architecture

The primary architecture that will be discussed in this paper is based on site-to-site VPN connections using Generic Routing Encapsulation (GRE) over IPSec. This architecture was selected because of the advantages of being able to encapsulate Enterprise grade routing protocols, including Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP). The encapsulation of all network and routing traffic within a GRE tunnel provides less complexity in the IPSec configuration and increases the speed of failover and redundancy within the network. Also, this architecture only includes the use of hardware-based encryption mechanisms, because software-based encryption platforms currently cannot be scaled to the level needed by most enterprises. Figure 1 is a high level overview of a two remote site V^{3}PN implementation:

[2] Connected: An Internet Encyclopedia. “Unreliable Delivery Model”. URL:http://www.freesoft.org/CIE/Topics/11.htm (December 6 2003)

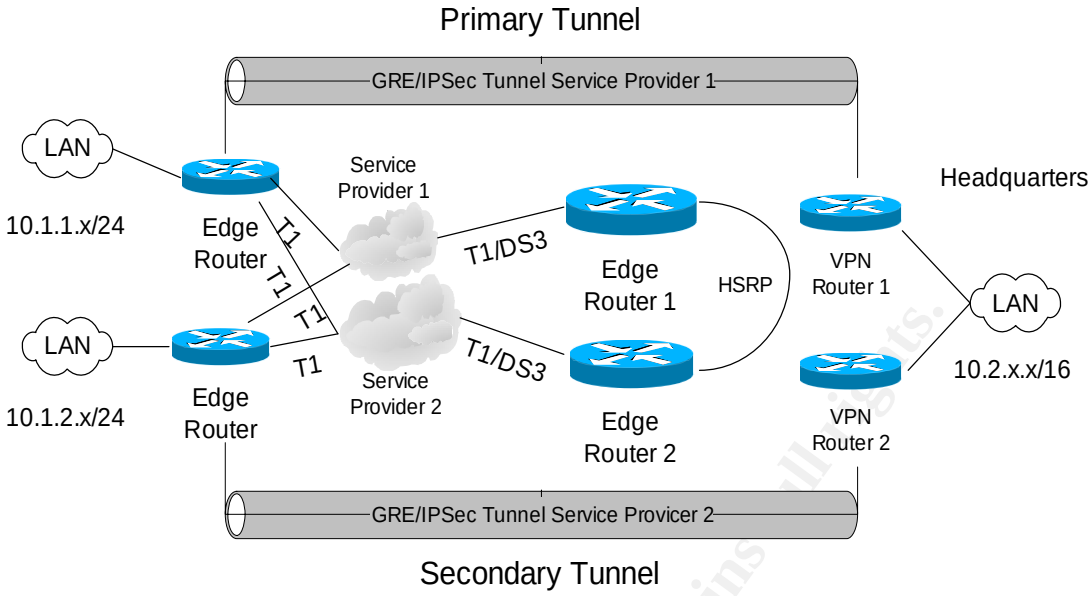

Figure 1. V[3]PN High Level Overview

The architecture will use the following Cisco recommended IPSec configuration guidance provided in the Voice and Video Enabled IPSec VPN (V[3]PN) Solution Reference Network Design whitepaper:

- Strong (3DES) encryption for both IKE and IPSec
- IP GRE with IPSec Tunnel mode
- Routing protocol (EIGRP)
- Diffie-Hellman Group 2 (1024 bit) for IKE
- Secure Hash Algorithm (SHA)-HMAC, a 160-bit rather than 128-bit with Message Digest 5 (MD5)-HMAC (both hash algorithms are truncated to 12 bytes in the ESP packet trailer).[3]

RFC 1701 defines the GRE protocol as "a protocol for performing encapsulation of an arbitrary network layer protocol over another arbitrary network layer protocol."[4] By using this core feature of encapsulating traffic within GRE, normal routing protocols can be encapsulated (e.g, OPSF and EIGRP) that normally would not be protected by a standard IPSec implementation. The ability to run a feature-rich routing protocol over an IPSec encrypted tunnel allows the bypass of the standard slow timeout mechanisms inherent in IPSec and quick failover to

[3] Cisco Systems. "Cisco Voice and Video Enabled IPSec VPN (V3PN) Solution Reference Network Design". URL:http://www.cisco.com/application/pdf/en/us/guest/netsol/ns241/c649/ccmigration_09186a0080146c8e.pdf (December 6 2003)

[4] Hanks, S., et al. "Generic Routing Encapsulation (GRE)". RFC 1701 IETF. October 1994 URL:http://www.ietf.org/rfc/rfc1701.txt (December 6 2003)

available routes and network interfaces. This is a core feature for deploying highly available V[3]PN networks.

The above requirements provide us the framework to begin creating the necessary configuration code to start building the GRE over IPSec tunnels. Figure 2 is a detailed drawing of the sample network configuration that I will be using.

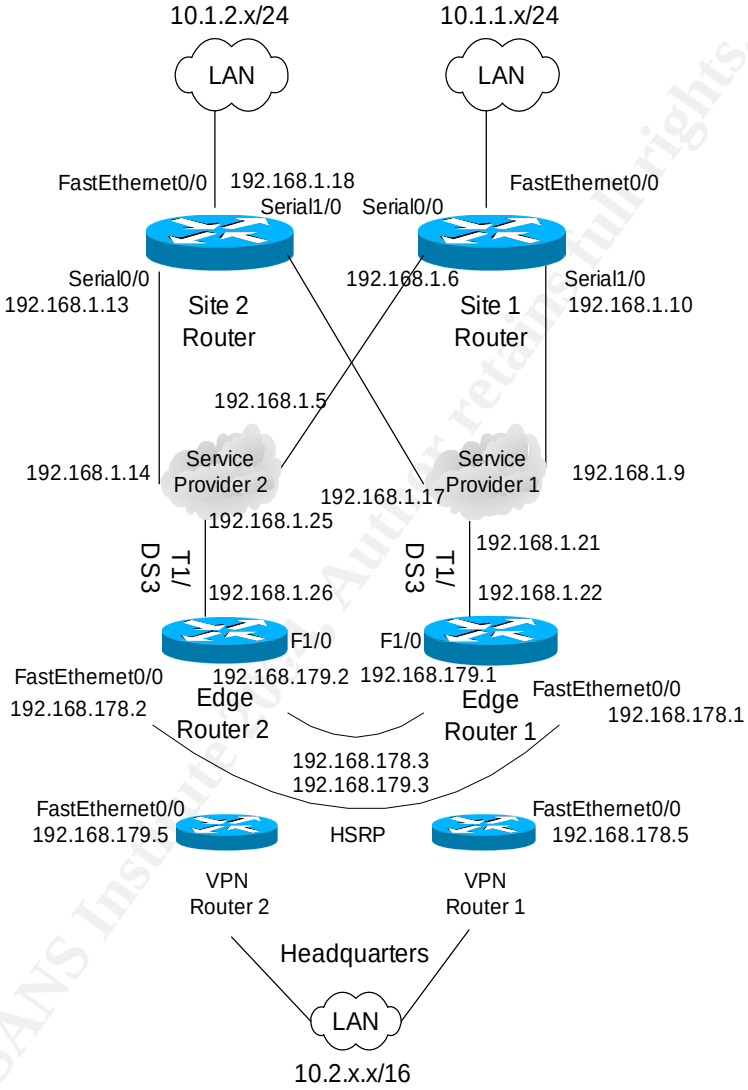

Figure 2. V[3]PN Detailed Architecture

Step 1

The first step is to create the IPSec configuration following the recommended configuration guidance from above.[3] The following is the necessary IOS code to add in order to complete this step:

Site 1&2 and HQ VPN 1&2 routers:

! ISAKMP policy using 3DES encryption, pre-shared keys, and Diffie-Hellman group 2 for IKE

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2

crypto ipsec transform-set vpncoreset esp-3des esp-sha-hmac
```

Site 1 router:
! Creation of crypto maps to identify interesting traffic to encrypt and encryption end points

```
crypto map site1vpnmap 10 ipsec-isakmp
 set peer 192.168.178.5
 set transform-set vpncoreset
 match address 101
crypto map site1vpnmap 20 ipsec-isakmp
 set peer 192.168.179.5
 set transform-set vpncoreset
 match address 102
```

Site 2 route:
! Creation of crypto maps to identify interesting traffic to encrypt and encryption end points

```
crypto map site2vpnmap 10 ipsec-isakmp
 set peer 192.168.178.5
 set transform-set vpncoreset
 match address 101
crypto map site2vpnmap 20 ipsec-isakmp
 set peer 192.168.179.5
 set transform-set vpncoreset
 match address 102
```

HQ VPN router 1:
! Creation of crypto maps to identify interesting traffic to encrypt and encryption end points

```
crypto map hqvpn1map 10 ipsec-isakmp
 set peer 192.168.1.10
 set transform-set vpncoreset
 match address 101
crypto map hqvpn1map 20 ipsec-isakmp
 set peer 192.168.1.18
```

```
 set transform-set vpncoreset
 match address 102

HQ VPN router 2:
! Creation of crypto maps to identify interesting traffic to encrypt and
encryption end points

crypto map hqvpn2map 10 ipsec-isakmp
 set peer 192.168.1.6
 set transform-set vpncoreset
 match address 101
crypto map hqvpn2map 20 ipsec-isakmp
 set peer 192.168.1.14
 set transform-set vpncoreset
 match address 102
```

The above code creates 1 tunnel from each site router to both of the HQ VPN routers. Thus, each site has 2 paths to follow to get to the HQ network (10.2.x.x/16). The crypto maps are defining the end point of the IPSec tunnel and the match address statement, denoting which Access Control List (ACL), is used to identify the traffic that is to be encrypted over this tunnel. The following is the ACL 101 on Site router 1:

```
access-list 101 permit gre host 192.168.1.10 host 192.168.178.5
```

A further analysis of this ACL statement indicates that the match address 101 statement is instructing the router to protect all GRE traffic sourced from the Serial 1/0 interface of the Site 1 router to the FastEthernet0/0 interface of the HQ VPN 1 Router.

Step 2

The next step is to create the GRE tunnel interfaces to encapsulate the internal network traffic thereby allowing the IPSec tunnel to encrypt it. The following is the required code to create the GRE tunnel interfaces:

```
Site 1 router:
! Creation of GRE Tunnel interfaces
interface Tunnel0
 description Primary Tunnel to HQ VPN 1
 ip address 10.3.255.6 255.255.255.252
 tunnel source Serial1/0
 tunnel destination 192.168.178.5
!
interface Tunnel1
 description Secondary Tunnel to HQ VPN 2
```

```
 ip address 10.3.255.10 255.255.255.252
 tunnel source Serial0/0
 tunnel destination 192.168.179.5

HQ VPN router 1:
! Creation of GRE Tunnel interfaces
interface Tunnel0
 description Primary Tunnel to Site 1 router
 ip address 10.3.255.5 255.255.255.252
 tunnel source FastEthernet0/0
 tunnel destination 192.168.1.10

HQ VPN router 2:
! Creation of GRE Tunnel interfaces
interface Tunnel0
 description Secondary Tunnel to Site 1 router
 ip address 10.3.255.9 255.255.255.252
 tunnel source FastEthernet0/0
 tunnel destination 192.168.1.6
```

The above code is creating a point-to-point GRE tunnel from the Serial1/0 interface of Site 1 router to FastEthernet0/0 of HQ VPN Router 1 and Serial0/0 interface of Site 1 router to FastEthernet0/0 of HQ VPN Router 2. It is important to note that the GRE tunnel interfaces emulate the IPSec tunnels created earlier. Once the GRE tunnel interfaces are created, it is necessary to add the shared crypto keys on each router in order to create a tunnel for each peer. The keys may be different for each peer, but must be matched on either end of the tunnel. The following commands will need to be issued on each router:

```
Site 1 router:
crypto isakmp key yourkeyhere address 192.168.178.5
crypto isakmp key yourkeyhere address 192.168.179.5

HQ VPN router 1:
crypto isakmp key yourkeyhere address 192.168.1.10

HQ VPN router 2:
crypto isakmp key yourkeyhere address 192.168.1.6
```

In order to activate the IPSec tunnels the crypto maps created must be applied to both the Serial and the Tunnel interfaces on Site routers 1 and 2 in addition to the FastEthernet and the Tunnel interfaces on both HQ VPN 1 and 2 routers. The following is the required code to activate the IPSec tunnels:

```
Site router 1:
! Activation of the IPSec tunnels
```

```
interface Serial1/0
 description Service Provider 1 Circuit
 ip address 192.168.1.10 255.255.255.252
 crypto map site1vpnmap
interface Tunnel0
 description Primary Tunnel to HQ VPN 1
 ip address 10.3.255.6 255.255.255.252
 tunnel source Serial1/0
 tunnel destination 192.168.178.5
 crypto map site1vpnmap
interface Tunnel1
 description Secondary Tunnel to HQ VPN 2
 ip address 10.3.255.10 255.255.255.252
 tunnel source Serial0/0
 tunnel destination 192.168.179.5
 crypto map site1vpnmap
```

HQ VPN router 1:

```
! Activation of the IPSec tunnels
interface FastEthernet0/0
 ip address 192.168.178.5 255.255.255.0
 duplex full
 speed auto
 crypto map hqvpn1map
interface Tunnel0
 description Primary Tunnel to Site 1
 ip address 10.3.255.5 255.255.255.252
 tunnel source FastEthernet0/0
 tunnel destination 192.168.1.10
 crypto map hqvpn1map
```

HQ VPN router 2:

```
! Activation of the IPSec tunnels
interface FastEthernet0/0
 ip address 192.168.179.5 255.255.255.0
 duplex full
 speed auto
 crypto map hqvpn2map
interface Tunnel0
 description Primary Tunnel to Site 1
 ip address 10.3.255.9 255.255.255.252
 tunnel source FastEthernet0/0
 tunnel destination 192.168.1.6
 crypto map hqvpn2map
```

The addition of the crypto map commands on each router should begin the IPSec negotiation of the tunnels. The issuing of the show crypto isakmp sa command on the HQ VPN router 1 will yield the following output:

dst	src	state	conn-id	slot
192.168.1.10	192.168.178.5	QM_IDLE	540	0

This command shows the state of the IPSec tunnel as QM_IDLE, which is the normal operational condition. The Site 1 router should display two connection tunnels as QM_IDLE because both the primary and secondary tunnels are created and active. At this point, we have a working tunnel from Site 1 router to both HQ VPN routers. The only traffic that will be passed over the tunnels at this point is traffic between the point to point GRE tunnel network (10.3.255.4/30 in the case of Tunnel 0 on Site 1 router). In order to encapsulate the internal LAN traffic of both Site 1 and HQ, a routing protocol must be configured to advertise these various network routes over the GRE Tunnels. I have selected EIGRP as the routing protocol of choice (OSPF can be configured in a similar way) and it is configured on each router as follows:

```
Site 1 router:
router eigrp 100
 network 10.0.0.0 0.0.0.255
 no auto-summary

HQ VPN router 1:
router eigrp 100
 network 10.0.0.0 0.0.0.255
 no auto-summary

HQ VPN router 2:
router eigrp 100
 network 10.0.0.0 0.0.0.255
 no auto-summary
```

In addition to activating EIGRP I also want to limit the size of the routing table broadcast across the network for increased routing performance. I will be using route summarization to only broadcast the top level network routes needed to each site. This is configured as follows on each tunnel interface:[3]

```
Site 1 router:
! Configure summary EIGRP route
interface Tunnel0
 description Primary Tunnel to HQ VPN 1
 ip address 10.3.255.6 255.255.255.252
 ip summary-address eigrp 100 10.1.1.0 255.255.255.0 5
 tunnel source Serial1/0
```

```
 tunnel destination 192.168.178.5
 crypto map site1vpnmap
interface Tunnel1
 description Secondary Tunnel to HQ VPN 2
 ip address 10.3.255.10 255.255.255.252
 ip summary-address eigrp 100 10.1.1.0 255.255.255.0 5
 tunnel source Serial0/0
 tunnel destination 192.168.179.5
 crypto map site1vpnmap

HQ VPN router 1:
! Configure summary EIGRP route
interface Tunnel0
 description Primary Tunnel to Site 1
 ip address 10.3.255.5 255.255.255.252
 ip summary-address eigrp 100 10.2.0.0 255.255.0.0 5
 tunnel source FastEthernet0/0
 tunnel destination 192.168.1.10
 crypto map hqvpn1map
```

The command show ip route on the Site 1 router should produce a listing of the directly connected routes and the advertised 10.2.x.x/16 route from both HQ VPN router 1 and 2. If the routing table is empty, it is necessary to issue the ip routing command on the router to enable routing (this won't show up in the configuration but is needed to enable routing). It is important to note that there are two entries listed for the next hop to the 10.2.0.0/16 network when issuing the show ip route command on the Site 1 router:

```
D       10.2.0.0/16 [90/297635072] via 10.3.255.5, 05:36:56, Tunnel0
                    [90/297635072] via 10.3.255.9, 05:36:56, Tunnel1
```

The reason for this is because the cost of the primary and secondary tunnels is the same. In order to control the routing behavior of the connection it is necessary to change the delay of the primary and secondary tunnels so that the primary tunnel appears more attractive to EIGRP.[3] The following is the code required to make this change:

```
Site 1 router:
! Configure Tunnel delay to broadcast Tunnel0 as the preferred route
interface Tunnel0
 description Primary Tunnel to HQ VPN 1
 ip address 10.3.255.6 255.255.255.252
 ip summary-address eigrp 100 10.1.1.0 255.255.255.0 5
 delay 1000
 tunnel source Serial1/0
 tunnel destination 192.168.178.5
```

```
 crypto map site1vpnmap
interface Tunnel1
 description Secondary Tunnel to HQ VPN 2
 ip address 10.3.255.10 255.255.255.252
 ip summary-address eigrp 100 10.1.1.0 255.255.255.0 5
 delay 1526
 tunnel source Serial0/0
 tunnel destination 192.168.179.5
 crypto map site1vpnmap
```

The delay parameter is used by EIGRP when calculating the cost of each route. This change makes Tunnel0 the preferred route, but allows for failover to Tunnel1 if Tunnel0 becomes unavailable. Because GRE tunnels are treated as virtual interfaces in IOS, the actual tunnel interface will never go down. However, the EIGRP routing updates traversing that tunnel will cease if the path that tunnel is built over becomes unavailable. EIGRP will automatically select the Tunnel1 interface to start forwarding traffic after the configured timeout period.

Another consideration when establishing these tunnels is the flow of traffic from each interface of the Site 1 router. Because it is important to balance the load of the traffic between providers and provide redundancy, it is necessary to configure different routing paths for the tunnel traffic in addition to the default gateway route. The Site routers may not have the necessary memory or processor to receive the full or even locally connected Service Provider BGP routes. As a result, it also is necessary to manually configure some static routes and default gateways. The traffic path desired for Tunnel0 to take is over Service Provider 1, so a static route is configured for traffic destined to HQ VPN Router 1 to travel over that interface as follows:

```
ip route 192.168.178.0 255.255.255.0 Serial1/0
```

The same considerations are necessary for routing from the Tunnel1 interface to HQ VPN Router 2, traveling over Service Provider 2. The required route configuration is as follows:

```
ip route 192.168.179.0 255.255.255.0 Serial0/0
```

The ability to provide redundancy and failover for all other internet based traffic requires choosing a primary internet interface and configuring a higher cost route as the failover mechanism on the second interface. In the present case, because primary VPN tunnel is on Serial1/0, I have designated interface Serial0/0 as the primary interface for internet traffic and configured the following routes:

```
ip route 0.0.0.0 0.0.0.0 Serial0/0 50
ip route 0.0.0.0 0.0.0.0 Serial1/0 60
```

The numbers after the interface denote the cost of the route, with lower numbers being the preferred path. In the above example all default traffic will exit through the Serial0/0 interface unless that interface becomes unavailable, which would cause traffic to be routed out of the Serial1/0 interface. Configuring the routing behavior of the remote site, as shown above, allows the control of which Service Provider is handling the primary and secondary traffic. This approach also allows the enforcement of any SLA agreements pertaining to traffic traversing each Service Provider network end to end.

The following section on configuring the HQ internet facing routers will show how to control this same routing behavior for the HQ VPN routers.

Step 3

The last architectural configuration step that needs to occur is to setup the Internet facing routers at the HQ location for Border Gateway Routing Protocol (BGP) routing and failover. RFC 1771 describes BGP as the following:

> The primary function of a BGP speaking system is to exchange network reachability information with other BGP systems. This network reachability information includes information on the list of Autonomous Systems (ASs) that reachability information traverses.[5]

BGP is the core routing protocol that runs the Internet. Various organizations share their routing information through the implementation of BGP and its components. By using features of the BGP protocol, the routing behavior of traffic destined for our Internet facing routers can be determined. Cisco provides a document entitled "How to Use HSRP to Provide Redundancy in a Multihomed BGP Network" that covers the configuration of a single BGP routed network in an active/standby configuration[6]. Based on this document, the necessary configuration to provide an active/active configuration for the BGP routing of two separate Class C networks can be extrapolated. The following is the code necessary to complete the BGP and Hot Standby Routing Protocol (HSRP) configuration on the two Internet facing HQ routers:

```
Edge Router 1:
! BGP Configuration with as-path prepend
router bgp 64512
```

[5] Rekhter, Y., et al. "A Border Gateway Protocol 4 (BGP-4)". RFC 1771 IETF. March 1995 URL:http://www.ietf.org/rfc/rfc1771.txt (December 6 2003)

[6] Cisco Systems. "How to Use HSRP to Provide Redundancy in a Multihomed BGP Network". Jun 6th, 2003 URL:http://www.cisco.com/en/US/tech/tk365/tk80/technologies_configuration_example09186a0080093f2c.shtml (December 8 2003)

```
 no synchronization
 network 192.168.178.0
 network 192.168.179.0
 neighbor 192.168.1.21 remote-as 64513
 neighbor 192.168.1.21 route-map aspath out
 neighbor 192.168.178.2 remote-as 64512
 neighbor 192.168.178.2 next-hop-self
 neighbor 192.168.179.2 remote-as 64512
 neighbor 192.168.179.2 next-hop-self
 no auto-summary

route-map aspath permit 10
 match ip address 1
 set as-path prepend 64512
route map aspath permit 20
 match ip address 2

access-list 1 permit 192.168.179.0
access-list 2 permit 192.168.178.0

! HSRP Configuration
interface FastEthernet0/0
 ip address 192.168.178.1 255.255.255.0
 standby 1 ip 192.168.178.3
 standby 1 priority 105
 standby 1 preempt
 standby 1 track Hssi0/0
interface FastEthernet1/0
 ip address 192.168.179.1 255.255.255.0
 standby 2 ip 192.168.179.3
 standby 2 preempt

Edge Router 2:
! BGP Configuration with as-path prepend
router bgp 64512
 no synchronization
 network 192.168.178.0
 network 192.168.179.0
 neighbor 192.168.1.25 remote-as 64514
 neighbor 192.168.1.25 route-map aspath out
 neighbor 192.168.178.1 remote-as 64512
 neighbor 192.168.178.1 next-hop-self
 neighbor 192.168.179.1 remote-as 64512
 neighbor 192.168.179.1 next-hop-self
 no auto-summary
```

```
route-map aspath permit 10
 match ip address 1
 set as-path prepend 64512
route-map aspath permit 20
 match ip address 2

access-list 1 permit 192.168.178.0
access-list 2 permit 192.168.179.0

! HSRP Configuration
interface FastEthernet0/0
 ip address 192.168.178.2 255.255.255.0
 standby 1 ip 192.168.178.3
 standby 1 preempt
interface FastEthernet1/0
 ip address 192.168.179.2 255.255.255.0
 standby 2 ip 192.168.179.3
 standby 2 priority 105
 standby 2 preempt
 standby 2 track Hssi0/0
```

To further describe the above code example it is necessary to start by examining the BGP networks advertised via both routers. The 192.168.178.0 network is being advertised without the additional as-path attribute, making it appear more attractive across the Internet. The majority of the traffic destined for the 192.168.178.0 network will arrive on Edge Router 1's HSSI0/0 interface. The converse is true for traffic destined for the 192.168.179.0 network. The majority of this traffic will arrive on Edge Router 2's HSSI0/0 interface. Examination of the HSRP code indicates that the default gateway for any host on the 192.168.178.0 network should be the virtual address broadcast via HSRP or 192.168.178.3. When the HSSI0/0 interface of Edge Router 1 is running it will be the primary router controlling that ip address via the priority 105 statement. The router with the higher priority is the one that controls the virtual address. The way HSRP works is that if the HSSI0/0 interface goes down the HSRP priority is decremented by 10 (via the standby 1 track HSSI0/0 command), thus reducing the priority to 95. Since the default priority is 100, Edge Router 2 would immediately assume control of the virtual 192.168.178.3 address because of the configured standby 1 preempt command. The converse is true for the 192.168.179.0 network in which Edge Router 2 is the primary router in control and preempted by Edge Router 1 if the HSSI0/0 interface on Edge Router 2 goes down. Using the features of both BGP and HSRP provides for a highly available Internet facing solution to front end the HQ VPN routers.

As demonstrated above, the architecture of a V^3PN network can be made highly available and redundant using the features of existing protocols such as GRE, BGP, HSRP, and EIGRP. The ability to encapsulate the EIGRP routing protocol

with GRE over IPSec tunnels provides standard network routing failover times of 10-20 seconds. The combination of HSRP and BGP at the Headquarters location also provides for high availability and redundancy of the HQ VPN routers.

Security

The establishment of a working GRE over IPSec architecture is the first step in creating an Enterprise grade V^{3}PN. However, the most critical step is establishing the necessary security configurations based on industry recommended guidelines. One such recommended guidance document is the NSA published Router Security Configuration Guide.[7] This document describes numerous configuration steps to secure production Cisco routers connected to the Internet. I will be using this document as a foundation for the following configuration changes on the remote site and core routers in this example, as well as direction provided by the 12.2 Cisco IOS Security Configuration Guide.[8]

The first security step is to lockdown access to only authorized machines and users via TACACS+ over an encrypted ssh channel. The following configuration code makes the necessary security changes:[8]

```
All routers:
! Enable encrypted passwords in config
service password-encryption
enable secret 5 *encrypted password*
! Create local admin account in case TACACS+ server is down
username rtradmin privilege 15 password 7 *encrypted password*
! Create AAA configuration enabling authentication and authorization
aaa new-model
aaa authentication login default group tacacs+ local
aaa authorization exec default local if-authenticated group tacacs+
aaa session-id common
! Configure access-list for authorized administration stations
access-list 10 permit 10.2.1.1 log
access-list 10 deny any log
! Configure TACACS+ server
tacacs-server host 10.2.2.1 port 49 key yourkeyhere
tacacs-server directed-request
```

[7] Antoine, Vanessa, et al. “Router Security Configuration Guide”. September 27 2002 URL:http://www.nsa.gov/snac/cisco/guides/cis-2.pdf (December 8 2003)

[8] Cisco Systems. “Cisco IOS Security Configuration Guide Release 12.2”. URL:http://www.cisco.com/application/pdf/en/us/guest/products/ps1835/c1069/ccmigration_09186a008011dff4.pdf (December 8 2003)

```
! Configure vty terminals for ssh enabled access only from authorized
administration stations
line vty 0 4
 access-class 10 in
 exec-timeout 5 0
 login authentication default
 transport input ssh
 transport output none
```

In order to enable ssh it is necessary to configure a hostname, domain name, and generate an RSA key pair based upon that information:[8]

```
All routers:
hostname yourhostnamehere
ip domain-name yourdomainnamehere
crypto key generate rsa
```

The next steps harden the router configuration and remove certain privilege command levels, establish login banners, configuring the aux and con port, disabling unnecessary services, and configure ftp for configuration file transfer:[7]

```
All routers:
line con 0
 transport input none
 login authentication default
 exec-timeout 5 0
 ! Disable aux port
line aux 0
 transport input none
 exec-timeout 0 1
 login authentication default
 no exec
! Change default privilege levels
privilege exec level 15 connect
privilege exec level 15 telnet
privilege exec level 15 rlogin
privilege exec level 15 show ip access-list
privilege exec level 15 show access-lists
privilege exec level 15 show logging
privilege exec level 1 show ip
! Configure login banner with appropriate message from legal
banner motd ^C NO UNAUTHORIZED ACCESS ALLOWED ^C
! Configure FTP commands
ip ftp username ftpuser
ip ftp password yourftppassword
ip ftp source-interface closest_interface_to_ftp_server
```

```
! Disable CDP
no cdp run
! Disable small-servers
no service tcp-small-servers
no service udp-small-servers
! Disable finger
no ip finger
no service finger
! Disable HTTP server
no ip http server
no ip http secure-server
! Disable bootp
no ip bootp server
! Disable booting from network config
no boot network
no service config
! Disable source routing
no ip source-route
! Disable proxy-arp on each interface under interface config
no ip proxy-arp
! Disable ip directed-broadcasts
no ip directed broadcast
! Disable ICMP messages under each interface
no ip unreachable
no ip redirect
no ip mask-reply
! Disable SNMP if not used or secure it with access-lists
no snmp-server
```

After hardening the router configuration, access control lists must be created on the Internet facing routers to protect them. Only the essential services needed into the exposed Internet facing interfaces will be allowed, in addition to a few required services for the GRE over IPSec tunnels. The following is an adapted configuration list:[7]

```
Site 1 router:
access-list 151 remark SITE 1 SERIAL1/0 INTERNET ACL
access-list 151 remark ANTI-SPOOFING
access-list 151 deny icmp any any redirect
access-list 151 deny ip 127.0.0.0 0.255.255.255 any
access-list 151 deny ip 224.0.0.0 31.255.255.255 any
access-list 151 deny ip any 10.0.0.0 0.255.255.255 log
access-list 151 deny ip any 172.16.0.0 0.15.255.255 log
! The next access-list will break the example configuration but since the
example addresses are not routable and will be replaced the following
statement is included for completeness
```

```
access-list 151 deny ip any 192.168.0.0 0.0.255.255 log
access-list 151 deny ip host 0.0.0.0 any
access-list 151 permit esp host 192.168.178.5 host 192.168.1.10
access-list 151 permit udp host 192.168.178.5 host 192.168.1.10 eq 500
access-list 151 remark ICMP permit out and allow response
access-list 151 permit icmp any any echo-reply
access-list 151 permit icmp any any traceroute
access-list 151 permit icmp any any packet-too-big
access-list 151 permit icmp any any time-exceeded
access-list 151 permit icmp any any unreachable
access-list 151 remark BIT BUCKET
access-list 151 deny ip any any log-input

access-list 152 remark SITE 1 SERIAL0/0 INTERNET ACL
access-list 152 remark ANTI-SPOOFING
access-list 152 deny icmp any any redirect
access-list 152 deny ip 127.0.0.0 0.255.255.255 any
access-list 152 deny ip 224.0.0.0 31.255.255.255 any
access-list 152 deny ip any 10.0.0.0 0.255.255.255 log
access-list 152 deny ip any 172.16.0.0 0.15.255.255 log
! The next access-list will break the example configuration but since the
example addresses are not routable and will be replaced the following
statement is included for completeness
access-list 152 deny ip any 192.168.0.0 0.0.255.255 log
access-list 152 deny ip host 0.0.0.0 any
access-list 152 permit esp host 192.168.179.5 host 192.168.1.6
access-list 152 permit udp host 192.168.179.5 host 192.168.1.6 eq 500
access-list 152 remark ICMP permit out and allow response
access-list 152 permit icmp any any echo-reply
access-list 152 permit icmp any any traceroute
access-list 152 permit icmp any any packet-too-big
access-list 152 permit icmp any any time-exceeded
access-list 152 permit icmp any any unreachable
access-list 152 remark BIT BUCKET
access-list 152 deny ip any any log-input
```

Edge Router 1:

```
access-list 151 remark HQ HSSI 0/0 INTERNET ACL
access-list 151 remark ANTI-SPOOFING
access-list 151 deny icmp any any redirect
access-list 151 deny ip 127.0.0.0 0.255.255.255 any
access-list 151 deny ip 224.0.0.0 31.255.255.255 any
access-list 151 deny ip any 10.0.0.0 0.255.255.255 log
access-list 151 deny ip any 172.16.0.0 0.15.255.255 log
```

```
! The next access-list will break the example configuration but since the
example addresses are not routable and will be replaced the following
statement is included for completeness
access-list 151 deny ip any 192.168.0.0 0.0.255.255 log
access-list 151 deny ip host 0.0.0.0 any
access-list 151 permit esp host 192.168.1.10 host 192.168.178.5
access-list 151 permit esp host 192.168.1.6 host 192.168.179.5
access-list 151 permit udp host 192.168.1.10 host 192.168.178.5 eq 500
access-list 151 permit udp host 192.168.1.6 host 192.168.179.5 eq 500
access-list 151 remark ICMP permit out and allow response
access-list 151 permit icmp any any echo-reply
access-list 151 permit icmp any any traceroute
access-list 151 permit icmp any any packet-too-big
access-list 151 permit icmp any any time-exceeded
access-list 151 permit icmp any any unreachable
access-list 151 remark BIT BUCKET
access-list 151 deny ip any any log-input
```

These access control lists prevent any traffic sourced from RFC 1918[9] address space inbound into the network, because these are non-routable addresses and should not appear on the Internet. Another access-list not shown in the example is to prevent addressed sourced with the assigned address block. For example, if 192.168.2.0/29 is allocated, the following access control list should be included:

```
access-list 151 deny ip 192.168.2.0 7.255.255.255 any
```

The next set of access control list commands allow the encapsulating security payload (ESP)[10] packets of IPSec and internet key exchange (IKE) packets from the HQ VPN Routers into the interface. The last set of ACL commands allow normal ICMP messages, including control messages for Path MTU discovery, which is covered in the Quality of Service section.

Beyond the standard GRE over IPSec configuration and in addition to the above security configuration there also is a requirement for this V^{3}PN configuration to provide normal day-to-day Internet access for the remote sites. In order to secure this communication and adhere with the corporate web surfing compliance software, a Context-Based Access Control (CBAC) firewall with IDS features and Websense URL filtering capabilities is implemented. Support for Websense filtering was added into the mainstream IOS with the advent of the

---

[9] Rekhter, Y., et al. "Address Allocation for Private Internets". RFC 1918 IETF. February 1996 URL:http://www.ietf.org/rfc/rfc1918.txt (December 8 2003)

[10]Atkinson, R. "IP Encapsulating Security Payload (ESP)". RFC 1827 IETF. August 1995 URL:http://www.ietf.org/rfc/rfc1827.txt (December 8 2003)

12.3 IOS code release.[11] The following code is based on the Cisco 12.2 Security Command Reference[8] and the recommended configuration of URL filtering[12]:

```
Site 1 router:
! CBAC Configuration
ip inspect name site1fw ftp
ip inspect name site1fw smtp
ip inspect name site1fw http java-list 51 urlfilter timeout 20
! No java filtering enabled
access-list 51 permit any
! Configuration of Websense URL Filtering
ip urlfilter allow-mode on
ip urlfilter cache 4500
ip urlfilter server vendor Websense 10.2.3.1
! Configuration of IDS
ip audit attack action drop
ip audit notify log
ip audit name Site1_IDS info list 99 action alarm
ip audit name Site1_IDS attack list 99 action drop
! Exclude internal network from triggering IDS
access-list 99 deny 10.1.1.0 0.0.0.255
access-list 99 permit any
```

The above configuration is configuring the IOS firewall feature to inspect ftp, smtp, and http traffic destined out of the network. A java-list command can specify that only certain internet addresses are allowed from java enabled applications. In this example, I have allowed all java enabled applications, but it can be configured for more stringent java controls. The ip urlfilter commands configure the integration with Websense and caches 4500 addresses in memory. The default mode is to deny web requests (ip urlfilter allow-mode off) if the Websense server is unavailable, but I have configured it to allow web requests even if the server is unavailable (ip urlfilter allow-mode on). Context-based access control works on external interfaces by creating temporary openings in the inbound access list (this must be an extended ACL) for the duration of the session.[8] With release 12.3 of Cisco IOS code the IDS audit engine now

---

[11] Cisco Systems. "Cisco IOS Security New-Features List, Release 12.3". August 13 2003 URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123tech/sc_ftlst.htm (December 8 2003)

[12] Cisco Systems. "Firewall Websense URL Filtering". April 11 2003 URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122y/122yu11/ftwebsen.htm (December 8 2003)

supports 101 signatures of the most common attacks[13]. In order to restrict the traffic that can go outbound to the internet, one more access-control list for outbound traffic needs to be configured:

```
Site 1 router:
access-list 199 permit tcp any any eq www
access-list 199 permit tcp any any eq 443
access-list 199 permit tcp any any eq smtp
access-list 199 permit tcp any any eq ftp
access-list 199 permit tcp any any eq ftp-data
access-list 199 permit esp host 192.168.1.10 host 192.168.178.5
access-list 199 permit esp host 192.168.1.6 host 192.168.179.5
access-list 199 permit udp host 192.168.1.10 host 192.168.178.5 eq 500
access-list 199 permit udp host 192.168.1.6 host 192.168.179.5 eq 500
access-list 199 permit icmp any any echo-request
access-list 199 deny ip any any log
```

In order to provide Internet access for internal clients using private address space, network address translation (NAT) between the internal network and the Internet needs to be provided. The addition of the following code completes the necessary NAT configuration:[14]

```
Site 1 router:
ip nat pool ovrld 192.168.2.1 192.168.2.1 prefix-length 24
ip nat inside source list 150 pool ovrld overload
access-list 150 permit ip 10.1.1.0 0.0.0.255 any
```

The final configuration step requires access control lists, firewall inspection rules, and IDS audit rules on the appropriate interfaces on the remote site router as follows:

```
Site 1 router:
interface Serial1/0
 description Connected to Service Provider 1
 ip address 192.168.1.10 255.255.255.252
 ip access-group 151 in
 ip access-group 199 out
 ip audit Site1_IDS in
```

[13] Cisco Systems. “Cisco IOS Software Release 12.3 New Features and Platforms”. URL:http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/prod_bulletin09186a0080199900.html (December 8 2003)

[14] Cisco Systems. “Cisco IOS Network Address Translation”. URL:http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ionetn/prodlit/1195_pp.pdf (December 8 2003)

```
 ip inspect site1fw out
 ip nat outside
 crypto map site1vpnmap
interface Serial0/0
 description Connected to Service Provider 2
 ip address 192.168.1.6 255.255.255.252
 ip access-group 152 in
 ip access-group 199 out
 ip audit Site1_IDS in
 ip inspect site1fw out
 ip nat outside
 crypto map site1vpnmap
interface FastEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
```

Through the use of industry standard configuration guidance, technologies such as access control list, context-based access control, IDS, and network address translation one can take the standard GRE over IPSec architecture configured above and make it into a secure Enterprise grade implementation. With feature support for URL filtering, pushing Internet access out to the site locations continues to support essential Enterprise core security requirements.

Quality of Service

Now that the GRE over IPSec tunnels have been configured and secured, the router configuration and added additional security enhancements made, the last core technology implementation of Quality of Service (QoS) will be discussed. The actual configuration of QoS will differ across every organization because of the different applications, business critical services, and network bandwidth requirements. My intent is to provide a framework for establishing a solid QoS strategy and provide techniques I've discovered through the actual implementation of QoS. The key to a successful QoS implementation lies in understanding the network infrastructure and the key business services involved. Having a full understanding of these elements requires less technology and more communication with the various customers using the network serves, whether they are internal or external customers.

The first recommendation I have addressing the nuts and bolts of configuring QoS is to survey the key business leaders around the organization and ask them what they consider to be the most critical business systems. If the results are similar to my experience, most will point to a handful of key systems that support their specific line of business. Once a quick hit list of the key business applications have been assembled, I think it is important to install a network sniffer and start to model the traffic each of the identified applications uses. It is essential to determine what IP protocols and ports they use, discover who and

where the largest consumer of these systems resides, and model important issues (e.g., response time). I would model these over the course of a few weeks and pay attention to the spikes in traffic activity. Then it is important to compile this information and begin to identify the network requirements necessary to begin building different traffic profiles.

Since the very definition of V^{3}PN includes Voice over IP (VoIP) traffic, I will take this particular class of traffic and demonstrate how to create a QoS traffic profile and integrate it into the existing architecture, described above. I also will demonstrate how these techniques can be used to solve some mainstream problems in rather interesting ways.

Typical VoIP traffic does not consume large amounts of bandwidth compared to other applications. It does, however, require very consistent network performance (jitter) and low latency. A typical G.729 conversation only requires about 31.2 Kbps (on Ethernet) of bandwidth.[15] Nonetheless, with the additional overhead of GRE encapsulation and IPSec encapsulation this per call bandwidth number may increase to around 56 Kbps.[3] This still is not a huge amount of bandwidth compared to other Enterprise applications. Thus, with bandwidth numbers acceptable, the other key concepts behind success VoIP will be explored.

Latency is an easy concept to grasp; it is the time it takes a packet to travel from point A to point B across the network. Introducing IPSec compression does have some effect on latency, but usually if the end-to-end ping is consistently under 150ms one-way this time in actual practice is adequate. On the other hand, jitter is a critical element to near toll quality VoIP applications and is defined by searchNetworking as:

> In voice over IP (VoIP), jitter is the variation in the time between packets arriving, caused by network congestion, timing drift, or route changes. A jitter buffer can be used to handle jitter.[16]

In addition to considering the bandwidth and the latency of a network, network consistency also needs to be addressed. Unfortunately, most of the time it is consistency that determines the success of the VoIP implementation. The problem is in the core way voice is converted from sound and transmitted over the wire as data. A digital signal processor (DSP) takes a sampling of the voice over a period of time (20ms for G.729[3]) and converts that to a packet stream (50

---

[15] Cisco Systems. "Voice Over IP - Per Call Bandwidth Consumption". June 6 2003 URL:http://www.cisco.com/warp/public/788/pkt-voice-general/bwidth_consume.html (December 8 2003)

[16] searchNetworking.com. "jitter". Jun 18,2003 URL:http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci213534,00.html (December 8 2003)

packets per second for G.729[3]) and sends it across the wire. If the DSP at the other end receives only a partial deliver of that packet stream outside of its acceptable delay window, then only part of the voice conversation can be played. Thus, what occur are breaks in the conversation, pops, clicks, and in the worst case scenario, silence. In a G.729 configuration, it only takes 2 consecutive dropped packets to become noticeable audibly.[3] This interruption, usually a noticeable one to the end user, is why I consider jitter to be one of the most important factors in a successful VoIP deployment.

An additional key concept to remember when implementing VoIP is Call Admission Control (CAC). CAC is the mechanism to limit the amount of VoIP calls traversing the network at one time.[3] This is fundamental because if your network is configured to only handle 10 VoIP calls at a time and you allow the 11th onto the network, all 11 calls will suffer in voice quality. This is analogous to trying to park a huge Cadillac into a compact car parking spot, some of the car is going to be shaved off trying to force it into the spot. Voice traffic is similar, too many calls on the wire and some of them are going to be lost or delayed in transit, thus resulting in pops, clicks, and dead air.

So now that it has been established how not to configure VoIP, methods that are more appropriate will be addressed. Most modern day VoIP phones and systems mark the voice packets exiting them with the type of service (ToS) field in the ip header set to 5, which denotes mission-critical traffic. Maybe because I'm paranoid or just don't trust all implementations, I make sure that when the packet enters the edge router on the network it is reclassified with a ToS setting of 5. Let's assume that this particular VoIP implementation uses UDP ports 2300-2399 for voice communication (you should be able to identify what ports your particular vendor might be using for voice bearer traffic). In order to configure and mark these packets it is necessary to create a QoS class map to identify this traffic:

```
class-map match-any VoIP
 match access-group 2000
access-list 2000 permit udp any any range 2300 2399
```

This access list is capturing any traffic destined for UDP ports 2300 through 2399 entering the router, and marking them as part of the class VoIP. Next, a QoS policy map to do something with these packets that we have accumulated into the VoIP class needs to be configured:

```
policy-map Mark_VoIP
 class VoIP
  set precedence 5
 class class-default
  set precedence 1
```

The Mark_VoIP policy map marks every packet that matches the class VoIP with an ip precedence (ToS) value of 5. It also marks any other packet not matching any other classes with the ip precedence of 1. In order for this policy to work, it needs to be applied to an interface in either an inbound or outbound configuration to start analyzing traffic:

```
interface FastEthernet0/0
 service-policy input Mark_VoIP
```

That will activate the policy map for all packets inbound into the FastEthernet0/0 interface of the router. The statistics of the packets marked can be displayed to ensure your policy is working by issuing the **show policy map interface** command.

Now that the packet is marked with the correct ip precedence level, which is preserved after the packet is encrypted via IPSec[3], bandwidth by ip precedence values can be allocated. Again, another class map is created to gather all of the packets with an ip precedence of 5 into a QoS class:

```
class-map match-any Prec_5
 match ip precedence 5
```

Note in the configuration above all of the traffic that did not match the VoIP class was set to an ip precedence of 1. It is good practice to control rogue traffic invading your Prec_5 class using this method. Now the decision is needed to determine how much bandwidth to allocate to the new Prec_5 class traffic. When provisioning QoS bandwidth it is a good rule of thumb to not allocate more than 75% of the bandwidth to classes.[3] This is done to prevent throttling of critical network traffic, for example routing updates, from being starved for bandwidth. Assume we want to provision enough bandwidth to have 10 G.729 VoIP calls in session at one time. Using the bandwidth numbers from above, we determine that 10 calls X 56 Kbps (GRE and IPSec overhead) = 560 Kbps of bandwidth we need to allocate on our interface. Another policy map could be configured and applied to the necessary interface:

```
policy-map Outbound
 class Prec_5
  priority 560
class class-default
 fair-queue
interface Serial1/0
 service-policy output Outbound
```

The priority statement reserves 560 Kbps of bandwidth for the VoIP calls and the interface will forward this traffic before it services any of the other queues. The fair-queue statement under the default class services the remaining queues fairly

to prevent bandwidth starvation. That is basically all there is to the configuration of QoS. The configuration of Call Admission Control is usually handled by the VoIP management device (which can be the router if configured for VoIP calls as well) and not the edge router. It is important that your network QoS expectations are in line with your CAC settings. As a good rule I always provision my VoIP class for the worst case scenario (i.e., I have an office with 20 VoIP phones I assume that 90% may be on at one time) until I can collect more historical data to further tune the class. Class tuning is an ongoing exercise as new applications are introduced into the network, but can also be a great tool to manage bandwidth and help the decision making process on increasing bandwidth when needed. The priority statement in the VoIP class map should handle the jitter on the network, but it is a good idea to setup a monitor (like Chariot[3]) when the network is first built to make sure jitter and latency is consistent over time.

Another application for QoS concepts is what I like to refer to as reverse QoS. Using the same code and configuration you can also throttle down applications that are unwanted and hard to identify through normal means (like Peer to Peer applications). The following is an example of how to police P2P applications on the network using QoS:

```
class-map match-any P2P_Sharing
 match protocol fasttrack
 match protocol gnutella
 match protocol kazaa2
policy Outbound
 class P2P_Sharing
  police cir 8000
    conform-action drop
    exceed-action drop
    violate-action drop
```

This example is using Network Based Application Recognition (NBAR)[17] to identify various P2P clients and drop all the traffic matched by this class. The following is a description of the abilities of NBAR:

> NBAR addresses IP QoS classification requirements by classifying application-level protocols so that QoS policies can be applied to the classified traffic. NBAR addresses the ongoing need to extend the classification engine for the many existing and emerging application protocols by providing an extensible Packet Description Language (PDL). NBAR can determine which protocols and applications are currently

[17] Cisco Systems. "Network-Based Application Recognition". Jun 26,2000 URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121limit/121e/121e2/nbar2e.htm (December 8 2003)

running on a network so that an appropriate QoS policy can be created based upon the current traffic mix and application requirements.[17]

The last element of QoS to examine is the use of Path MTU discovery (PMTUD)[18] and manually setting the maximum transmission units (MTU) of the tunnel interfaces. Path MTU discovery is almost always enabled on a host, thus all TCP/IP packets from the host will have the Do not Fragment (DF) flag set.[18] In a typical operation, when the host sends a full maximum segment size (MSS) packet with the DF flag set, PMTUD works to try and reduce the MSS to a value it can send on the GRE over IPSec tunnel.[18] Since we are running GRE + IPSec in Tunnel mode the resulting Tunnel interface MTU needs to be set to 1420 bytes.[18] The is configuration appears on each tunnel interface:

```
Site 1 router:
interface Tunnel0
 description Primary Tunnel to HQ VPN 1
 ip address 10.3.255.6 255.255.255.252
 ip summary-address eigrp 100 10.1.1.0 255.255.255.0 5
 delay 1000
 ip mtu 1420
 tunnel path-mtu-discovery
 tunnel source Serial1/0
 tunnel destination 192.168.178.5
interface Tunnel1
 description Secondary Tunnel to HQ VPN 2
 ip address 10.3.255.10 255.255.255.252
 ip summary-address eigrp 100 10.1.1.0 255.255.255.0 5
 delay 1526
 ip mtu 1420
 tunnel path-mtu-discovery
 tunnel source Serial0/0
 tunnel destination 192.168.179.5
 crypto map site1vpnmap
```

A further analysis of the tunnel path-mtu-discovery command yields the following flow of events:

1. GRE will copy the DF bit from the data IP header to the GRE IP header.

2. If the DF bit is set in the GRE IP header and the packet will be "too large" after IPsec encryption for the IP MTU on the physical outgoing

[18] Cisco Systems. “IP Fragmentation and PMTUD“. July 2 2003 URL:http://www.google.com/search?q=firebird&sourceid=mozilla-search&start=0&start=0&ie=utf-8&oe=utf-8 (December 8 2003)

interface, then IPsec will drop the packet and notify the GRE tunnel to reduce its IP MTU size.

3. IPsec does PMTUD for its own packets and if the IPsec PMTU changes (if it is reduced), then IPsec doesn't immediately notify GRE, but when another "too large" packet comes thorough, then the process in step 2 occurs.

4. GRE's IP MTU is now smaller, so it will drop any data IP packets with the DF bit set that are now too large and send an ICMP message to the sending host.[18]

The recommended guidance is to implement both the ip mtu command and path-mtu-discovery on the tunnel interfaces.[18] Proper configuration of MTU settings and PMTUD are critical to efficient network operations. Special caution needs to be given when traffic is traversing a GRE over IPSec tunnel and a NAT device. I have found that NAT devices will break the path-mtu-discovery and certain network transmissions will fail. This behavior is also why it is critical for the network to be able to pass the necessary ICMP control messages successfully across the network infrastructure.

Quality of Service can be a powerful tool to efficiently add and control network bandwidth across the corporation. It can be extended to add voice and video services onto the GRE over IPSec tunnel architecture, and used to control harmful network based traffic, such as worms and Peer to Peer applications. To fully used the power of QoS requires careful planning and creation of Enterprise wide SLA agreements for critical business services traversing the network infrastructure.

Conclusion

Creating a secure, reliable, and redundant V^3PN is indeed possible and desirable. This paper is a guide to adapt the various technological components presented, as well as others that can be incorporated into this architecture, to harness the power, flexibility, and cost savings realized by migrating network infrastructure to a secure Enterprise grade V^3PN.

References

1. International Engineering Consortium. “Virtual Private Networks (VPNs)”. URL:http://www.iec.org/online/tutorials/vpn/topic02.html (December 6 2003)
2. Connected: An Internet Encyclopedia. “Unreliable Delivery Model”. URL:http://www.freesoft.org/CIE/Topics/11.htm (December 6 2003)
3. Cisco Systems. “Cisco Voice and Video Enabled IPSec VPN (V3PN) Solution Reference Network Design”. URL:http://www.cisco.com/application/pdf/en/us/guest/netsol/ns241/c649/ccmigration_09186a0080146c8e.pdf (December 6 2003)
4. Hanks, S., et al. “Generic Routing Encapsulation (GRE)”. RFC 1701 IETF. October 1994 URL:http://www.ietf.org/rfc/rfc1701.txt (December 6 2003)
5. Rekhter, Y., et al. “A Border Gateway Protocol 4 (BGP-4)”. RFC 1771 IETF. March 1995 URL:http://www.ietf.org/rfc/rfc1771.txt (December 6 2003)
6. Cisco Systems. “How to Use HSRP to Provide Redundancy in a Multihomed BGP Network”. Jun 6th, 2003 URL:http://www.cisco.com/en/US/tech/tk365/tk80/technologies_configuration_example09186a0080093f2c.shtml (December 8 2003)
7. Antoine, Vanessa, et al. “Router Security Configuration Guide”. September 27 2002 URL:http://www.nsa.gov/snac/cisco/guides/cis-2.pdf (December 8 2003)
8. Cisco Systems. “Cisco IOS Security Configuration Guide Release 12.2”. URL:http://www.cisco.com/application/pdf/en/us/guest/products/ps1835/c1069/ccmigration_09186a008011dff4.pdf (December 8 2003)
9. Rekhter, Y., et al. “Address Allocation for Private Internets”. RFC 1918 IETF. February 1996 URL:http://www.ietf.org/rfc/rfc1918.txt (December 8 2003)
10. Atkinson, R. “IP Encapsulating Security Payload (ESP)”. RFC 1827 IETF. August 1995 URL:http://www.ietf.org/rfc/rfc1827.txt (December 8 2003)
11. Cisco Systems. “Cisco IOS Security New-Features List, Release 12.3”. August 13 2003 URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123tech/sc_ftlst.htm (December 8 2003)
12. Cisco Systems. “Firewall Websense URL Filtering”. April 11 2003 URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122limit/122y/122yu11/ftwebsen.htm (December 8 2003)
13. Cisco Systems. “Cisco IOS Software Release 12.3 New Features and Platforms”. URL:http://www.cisco.com/en/US/products/sw/iosswrel/ps5187/prod_bulletin09186a0080199900.html (December 8 2003)

14. Cisco Systems. “Cisco IOS Network Address Translation”. URL:http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ionetn/prodlit/1195_pp.pdf (December 8 2003)
15. Cisco Systems. “Voice Over IP - Per Call Bandwidth Consumption”. June 6 2003 URL:http://www.cisco.com/warp/public/788/pkt-voice-general/bwidth_consume.html (December 8 2003)
16. searchNetworking.com. “jitter”. Jun 18,2003 URL:http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci213534,00.html (December 8 2003)
17. Cisco Systems. “Network-Based Application Recognition”. June 26 2000 URL:http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121limit/121e/121e2/nbar2e.htm (December 8 2003)
18. Cisco Systems. “IP Fragmentation and PMTUD“. July 2 2003 URL:http://www.google.com/search?q=firebird&sourceid=mozilla-search&start=0&start=0&ie=utf-8&oe=utf-8 (December 8 2003)

Last Updated: May 28th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event