Lab
8
TLEN 5460

IPSec VPNs (Site-to-site)

TELECOMMUNICATIONSYSTEMSLABORATORY

OBJECTIVES

Familiarize students with the concept of VPN services

Learn how to Configure site-to-site VPN services with IPSec

Reaffirm use of Access Lists and NAT

Virtual Private Networks

1. Introduction

Historically, companies with multiple sites used various technologies like leased lines, Frame Relay and ATM for inter-office connectivity. These methods were not scalable and turned out to be quite expensive to install and maintain. An alternative to private lines is the *Virtual Private Network (VPN)* – sites are connected to a public network and tunnels are used for secure site-to-site communication.

Tunneling uses the process of encapsulation to protect the data – the sender gateway encapsulates the native data packet in a standard tunnel protocol format and sends it over the Internet, the receiver gateway then de-encapsulates this packet and forwards the original datagram. GRE, L2TP and IPSec are examples of tunneling protocols.

The main objectives of tunneling protocols are to protect the data packet as it traverses the Internet. This is done by providing –

confidentiality– the information is stored in such a way that it can't bemeaningfully interpreted even if it were to be intercepted
integrity– ensures that the data hasn't been modified or in any waycompromised during it's journey across the Internet
authentication– ensures that the datagram actually came from thedevice it was expected to come from

Various tunneling protocols provide these features in different ways but the basic principles are the same.

Confidentiality is provided by means of data encryption. This can bedone by using a symmetric key process or an asymmetric key process. In symmetric key encryption, the same key is used at both ends to encrypt and decrypt the data. This method is extremely fast but the initial key exchange can be an issue. Eg: DES, 3DES, AES

In asymmetric key encryption, a pair of mathematically related keys – public and private – are used to encrypt and decrypt data, respectively. Data encrypted using the public key (freely available) can only be decrypted using the private key (available only to the authorized user). This method is slower than symmetric key encryption but key exchange is safer and more secure. Eg: RSA, Blowfish

The method of choice in the major tunneling protocols is the *Diffie-Hellman* key exchange method. This method combines the speed andefficiency of symmetric key encryption, at the same time, providing the security and ease of asymmetric key exchange and management. The DH algorithm makes use of 5 groups of very large prime numbers used as the modulus for the DH key generation process. The only requirement is that both of the peers must use the same DH group. The way this algorithm works is that both peers generate a pair of public-private keys in the same
DH group. They then exchange their public keys over the Internet. This does not pose a security threat as the public keys are meant to be freely available any way. Once this is done, each gateway combines its own private key and the peer's public key to make the session key. In this way, the peers have the same key (mathematically) without having transmitted it over the public network.

Data integrity is provided by using*one-way hash algorithms*– the data tobe transmitted is passed through a one-way hash function and the result
(always a fixed length value), along with the original packet, is transmitted to the peer. At the receiving end, the gateway separates the data from the hash and runs the same one-way hash function. The result obtained here is compared with the hash value received. Any discrepancy indicates that the datagram has been tampered with.

Authentication is provided by a method known as*Hashing MessageAuthentication Code* (HMAC). IT uses a process similar to the one-wayhash function but includes a pre-shared key (or a session key) in the datagram before passing it through the hash function. The only way, the receiver can get the correct hash value is to have the same pre-shared key. In this way, correctly computed hash values received from rogue

senders will not give the same value because of the absence of the pre-shared key in the hashed datagram.

Now that we've seen the generic working of a tunneling protocol, we will examine the most widely used specification for tunneling, IPSec.

2. IPSec

IP Security (IPSec) is an industry standard that defines how encryption, validation and authentication are handled by peer gateways. Two protocols are defined in the IPSec standard – ESP (Encapsulating Security Protocol) and AH (Authentication Header). The former provides services for encryption, data validation and authentication, while AH only does integrity checks and authentication.

> ***Encapsulating Security Protocol*** **(ESP)**runs directly on IP protocol 50and provides data confidentiality, integrity and anti-replay services. Encryption is done using symmetric key algorithms and data integrity is provided using one-way hash functions. During encapsulation, the ESP header is inserted between the original IP header and the new IP header for the tunnel interfaces. It contains the IP protocol value, *SecurityParameter Index* (SPI) and a sequence number. SPI is generated using acombination of the destination IP address and the security protocol used.
> IT is used to uniquely identify the security association for the tunnel. The sequence number is a monotonically increasing value and is used for anti-replay functionality.
>
> ***Authentication Header*** **(AH)**only provides for integrity and authenticationchecks in the same fashion as ESP. It runs on IP protocol 51. The AH header also contains SPI and sequence numbers along with header length and next header fields.

IPSec operates in two modes, this can be chosen by the peers during the tunnel setup and negotiation process. They are tunnel mode and transport mode.

> *Tunnel mode* is the most commonly implemented mode. A secureconnection is provided between two gateways, thus providing secure access to clients and networks behind these gateways. All IPSec computations, encapsulations and de-encapsulations are performed on the gateways. The end hosts have no knowledge of the presence of any tunneling protocols.

Transport mode is implemented between IPSec end systems. An IPSeccapable software must be present on the hosts to perform all the operations required for tunneling.

Actual tunnel establishment between peers is done using the Internet Key Exchange (IKE).

3. Internet Key Exchange

Internet Key Exchange (IKE) is a secure key management protocol used by IPSec to exchange information and proposals in a secure and dynamic fashion with little or no intervention from the users. The IKE proposals used to build a tunnel consist of two main phases (we will look at these phases in detail a little bit later). The following attributes are exchanged between IPSec peers during the IKE process –

encryption algorithms
hashing algorithms
Diffie-Hellman group
authentication method
pre-shared key
public keys
digital signatures

Once these attributes are exchanged between the peers, they can be used to secure future attribute exchanges which are actually used to protect the data being transported across the tunnel. When the tunnel is being built, objects called security associations (SA) are used to store the policy sets and keys used in that particular tunnel instance. An SA can be uniquely identified using the SPI, destination IP address and the tunnel protocol (ESP or AH).

Tunnel establishment is completed in two IKE phases.

IKE Phase 1 establishes a secure channel over which Phase 2negotiations can occur. This exchange contains the encryption algorithm, hashing algorithm, DH group and authentication method. IKE Phase 1 can run in 2 different modes – *main mode* and *aggressive mode*.
Main mode is used when both the IPSec peers have static IPs assigned to them.
Aggressive mode is used when one of the peers uses a dynamically assigned IP address. In this case, the dynamic peer must be the one that

initiates the tunnel establishment process. It should also have it's peer's address pre-configured along with the proposal list.

***IKE Phase** 2*is also called*quick mode*. Once Phase 1 is completed, theproposals to establish specific VPNs are exchanged between the peers. This involves the negotiations of SAs containing the encryption and authentication algorithms that will be used during actual data transfer. The attributes that are negotiated during Phase 2 are –

security protocol (ESP or AH)

tunnel mode or transport mode

optional DH group – this DH group is exchanged under the secure channel already established and is hence doubly-secured. This is called Perfect Forward Secrecy.

Proxy IDs – these are used to bind the VPN between the peers. The local proxy ID at one end corresponds to the remote proxy ID at the opposite end of the VPN, and vice versa.

A single Phase 1 channel can be used to generate multiple Phase 2 SAs or VPNs. A Phase 1 channel is bi-directional while a Phase 2 SA is always unidirectional. That's why we require 2 SAs for bidirectional communication over one tunnel.

The result of the IKE Phase 2 process is the creation of an IPSec VPN which can be used to securely transmit user datagrams.

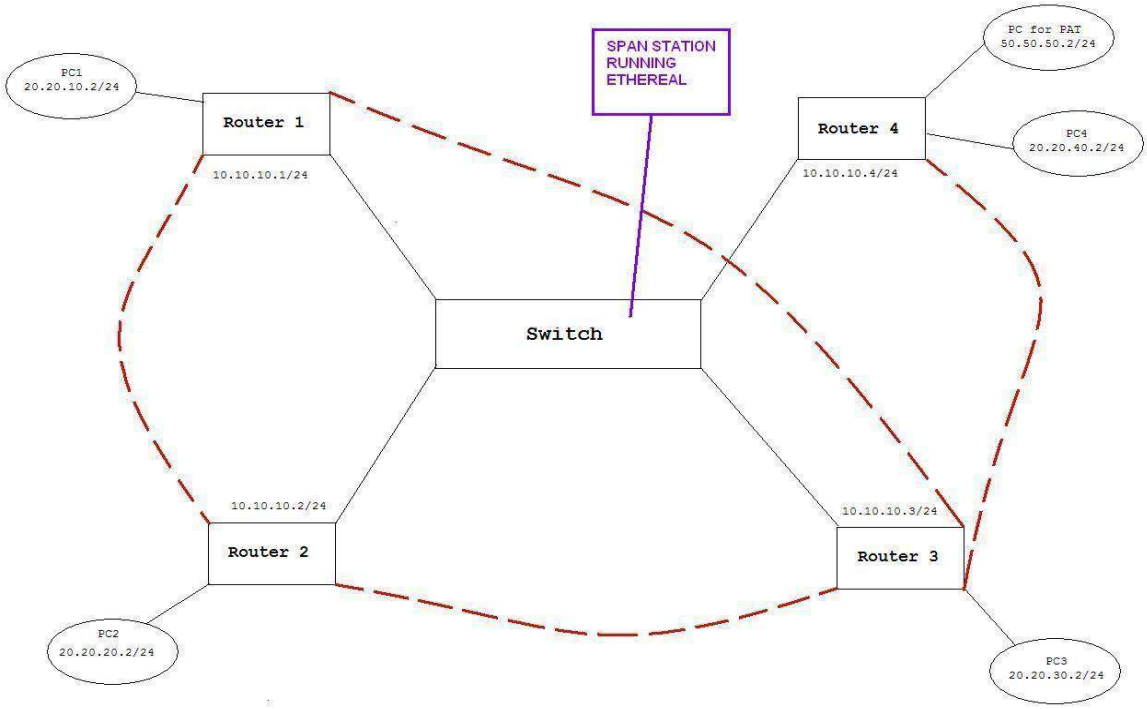
IPSec VPN lab configuration

Before connecting any routers, ensure that the IOS version on each one supports encryption. This can be done by running the following command -

```
Router#show version
```

The IOS name in the output must contain “K9”. Use the Cisco 3600 series of routers.

Connect the network as per the following diagram -



Lines with red dashes indicate the VPNs to be built.
Assign appropriate IPs to the interfaces where the PCs are connected and use those as the gateway.

Part 1 ~

On all routers running IPSec VPNs, the following steps are necessary irrespective of the number of VPNs built.

a) First create the ISAKMP policy that will be used –

```
crypto isakmp policy 1
 encryption aes 128          <<< use 128-bit AES encryption
 authentication pre-share    <<< use pre-shared keys – defined later
 group 2                     <<< use DH group 2
```

b) Then, create the IPSEC transform set to be used.

```
cryptoipsec transform-set aes_shaesp-aesesp-sha-hmac
```

aes_sha is the name of the transform set. You can use any name here. Ensure that the same one is used in the crypto map.
esp-aes is the encryption algorithm to be used to encrypt data packets.
esp-sha-hmac tells the peer to use Host Messahe Authentication for integrity checking.

The above 2 steps take care of basic IKE Phase 1 and Phase 2 negotiations. They have to be done only once for any number of VPNs built on the peer. The same policy can be used for all VPNs.

Now we have to configure the specifics of each individual VPN to be established. Following is the configuration for the VPN between Router 1 and Router 2. Corresponding configurations need to be entered for the other VPNs as well.

1. First, we need to define the pre-shared ISAKMP key to be used.

```
cryptoisakmp key lab1 address 10.10.10.2
```

lab1 is the pre-shared key to be used.
10.10.10.2 is the IP address of the VPN end-point. Be sure to use a 255.255.255.255 mask for the peer address.

2. A static route must be added for the destination of the VPN -

```
ip route 20.20.20.0 255.255.255.0 10.10.10.2
```

3. An access-list must be configured to allow interesting traffic to be encrypted and be transmitted over the secure VPN -

```
access-list 101 permit ip host 20.20.10.2 host 20.20.20.2
```

The source and destination are the PCs connected to the source and destination routers, respectively.

4. Then, we define the crypto map -

```
crypto map VPNS 10 ipsec-
   isakmp set peer 10.10.10.2
   set transform-set aes_sha
   match address 101
```

VPNS is the name of the crypto map. Any name can be used, but the same one must be applied to the outgoing interface.

10 is the entry number in the route map. We will see how other entries are added to the crypto map.
aes_sha is the transform set we had defined earlier.
is the access list defined to filter interesting traffic.

5. The crypto map is applied to the outgoing interface -

```
interface FastEthernet1/0
   ip address 10.10.10.1
   255.255.255.0 crypto map VPNS
```

Note : Only one crypto map can be applied to an interface.

To accommodate for multiple VPNs exiting the same interface, we use multiple entries in the same crypto map by using a different crypto map entry number in this case, 30

```
crypto map VPNS 30 ipsec-
   isakmp set peer 10.10.10.3
   set transform-set aes_sha
   match address 103
```

This is the crypto map entry for the VPN from Router 1 to Router 3.

Corresponding configuration for the VPN on Router 2 (for VPN to Router 1) is -

```
cryptoisakmp policy 1
   encryption aes 128
   authentication pre-
   share group 2
cryptoipsec transform-set aes_shaesp-aesesp-sha-hmac
cryptoisakmp key lab1 address 10.10.10.1
ip route 20.20.10.0 255.255.255.0 10.10.10.1
access-list 101 permit ip host 20.20.20.2 host 20.20.10.2
crypto map VPNS 10 ipsec-
   isakmp set peer 10.10.10.1
   set transform-set aes_sha
   match address 101
interface FastEthernet1/0
   ip address 10.10.10.2
   255.255.255.0 crypto map VPNS
```

Similarly, configure the other VPNs in the network .i.e. from Router 1 to Router 3, Router 2 to Router 3, and from Router 3 to Router 4.

After the VPNs are configured, you should be able to ping from the PCs connected to each router to the other PCs. The first few pings will time out because the IKE process is still establishing. Once pings are successful in both directions over the VPN, you can view the SAs on the routers.

This can be done with the following command -

```
R3#sh crypto isakmpsa
dst             src             state          conn-id slot
10.10.10.1      10.10.10.3      QM_IDLE              4   0
10.10.10.4      10.10.10.3      QM_IDLE              3   0

R4#sh crypto isakmpsa
dst             src             state          conn-id slot
10.10.10.4      10.10.10.3      QM_IDLE              2   0


R1#sh crypto isakmpsa
dst             src             state          conn-id slot
10.10.10.1      10.10.10.3      QM_IDLE              1   0
```

An SA will be visible for each successful VPN established.

If you want to clear the SAs on the routers, use the following commands -

```
R1#clear crypto isakmp
R1#clear crypto sa
R1#sh crypto isakmpsa
dst             src             state          conn-id slot
10.10.10.1      10.10.10.3      MM_NO_STATE          1    0 (deleted)
10.10.10.3      10.10.10.1      QM_IDLE              2    0
```

The source and destination of the SA is determined by which peer initiated the VPN establishment process. The old SA is marked as (deleted). That entry is removed from the table when traffic is passing through the new SA.

Part 2 ~

Now, configure NAT on one of the routers.

This used to demonstrate how traffic is chosen to be encrypted and sent over the VPN. We will use 2 sources going to the same destination – in this case, the PCs connected to Router 4 are the sources and any other PCs can be used as the destinations. In addition, on Router4 add a loopback 100.100.100.100/30 that should be considered a public website… all other loopbacks should be able to access it w/o encryption by using PAT/NAT (i.e. w/o using their private IP addresses to route traffic)

We configure PAT on Router 4 for all communication that isn't encrypted and sent over the VPN.

The configuration is as follows -

```
access-list 2 permit 50.50.50.2

ipnat inside source list 2 interface FastEthernet1/0

overload interface
   FastEthernet1/0 ip address
   10.10.10.4 255.255.255.0 ipnat
   outside crypto map VPNS
interface FastEthernet2/0
   ip address 50.50.50.1
   255.255.255.0 ipnat inside
```

After PAT is configured, ping the same destination (in this case, 20.20.30.2). On successful pings from both sources, you should see the following outputs on Router 4.

```
R4#sh ipnat trans
Pro Inside global      Inside local       Outside local
                       Outside global
icmp 10.10.10.4:512    50.50.50.2:512     20.20.30.2:512
20.20.30.2:512
R4#sh crypto isakmpsa
dst             src             state          conn-id slot
10.10.10.4      10.10.10.3      QM_IDLE              2    0
```

This output shows the two traffic flows originating from the different sources and being treated differently. The differentiation occurs due to the configured access lists – one used in the crypto map and the other for PAT.

EXTRA CREDIT: 5 Points!!!!!!!!!!

Replace the Router R4 in the diagram above with a **Juniper** (JR 2320 series)

router and re-create the IPsec Tunnels

Part 3 ~

Configure SPAN on the switch that represents the core, and forward all traffic from the network to a PC running ethereal. Include in your report the switch configuration and the ethereal capture file (print to file), showing the encrypted and non-encrypted packets exchanged by all routers.

Part 4 ~

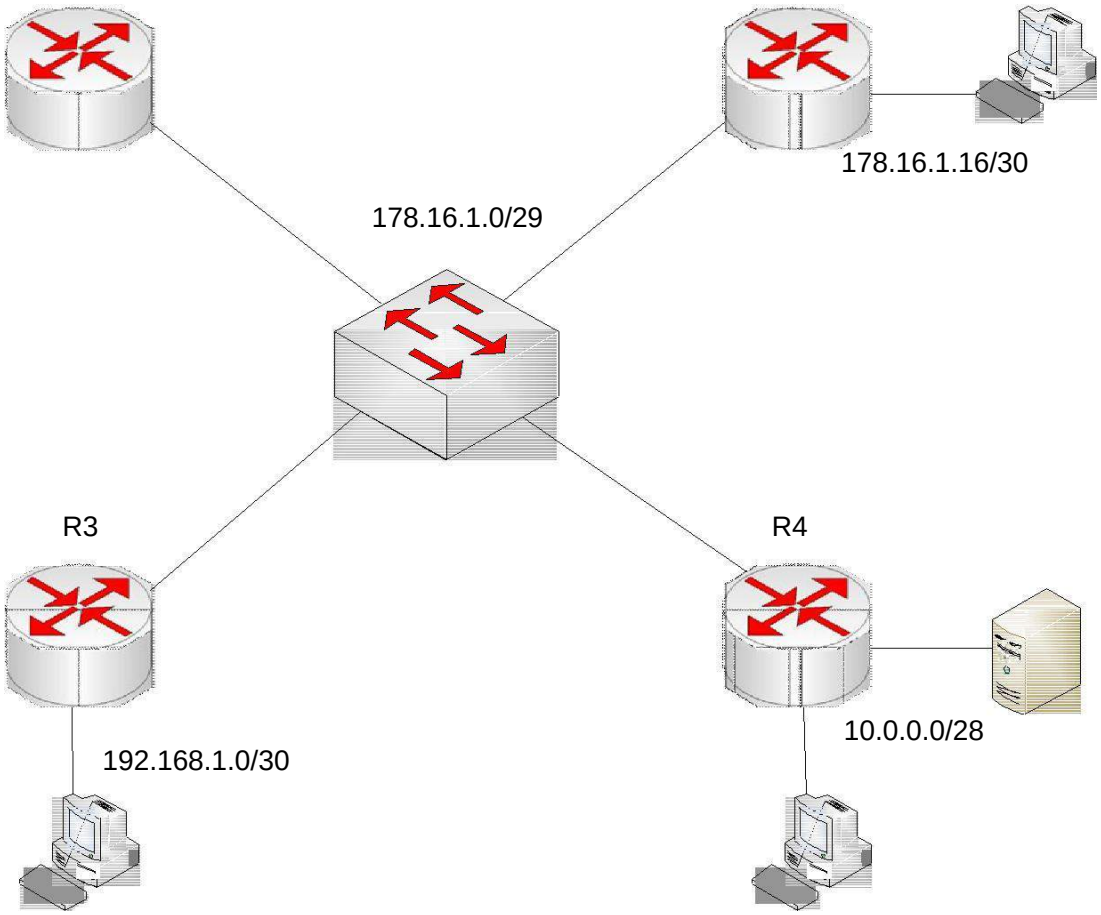
In this part you need to create a “Remote VPN” or “Easy VPN” (for instance for clients/employees using their personal Laptops to connect to the company Intranet.
In this instance, the clients on the 178.16.1.16 /30 network need to access the server on R4 using remote VPN.

Easy VPN works pretty much like the CU Boulder VPN (allowing you the access to CU resources from your home)

Steps:

1. The client needs to VPN into the server using the public IP address on R4. (Users belonging to the public network 178.168.1.16/30 should be able to remote VPN into the 10.0.0.0 network)
2. (The users in the private network 192.168.1.0/30 network should also be able to remote VPN into the 10.0.0.0 network after their addresses are translated to a public address by the Router R3)
3. You would have to download Cisco VPN client, which will allow you to establish the Remote VPN connection **(It should be already there on your Lab desktops)**

Please follow the guided instructions below to complete this objective:



Enable AAA for authentication of remote users

aaa new-model
aaa authentication login userauthen local
aaa authorization network groupauthor local
aaa session-id common

Configure the Usernameand Password that the remote users should have to connect

username cisco password 0 lab

Createthe ISAKMP policy to be used

```
cryptoisakmp policy 3
 hash md5
 authentication pre-share
 group 2
```

Assign the group Username and Password to be used by connecting users

```
cryptoisakmp client configuration group San
 key Jose
 poolippool
```

Create the IPSEC transform set to be used

```
cryptoipsec transform-set mysetesp-des esp-md5-hmac

crypto dynamic-map dynmap 10
 set transform-set myset
 reverse-route
```

Assign features to the Crypto map named "clientmap"

```
crypto map clientmap client authentication list userauthen
crypto map clientmapisakmp authorization list groupauthor
crypto map clientmap client configuration address initiate
crypto map clientmap client configuration address respond
crypto map clientmap 10 ipsec-isakmp dynamic dynmap
```

Associate the Crypto map with the public interface of the router

Assign the IP pool to be used by the remote users

ip local pool ippool (IP Addresses to be alloted to connected users)

Configure PAT for internal users to be able to reach outside the network

http://www.cisco.com/warp/public/473/41.html

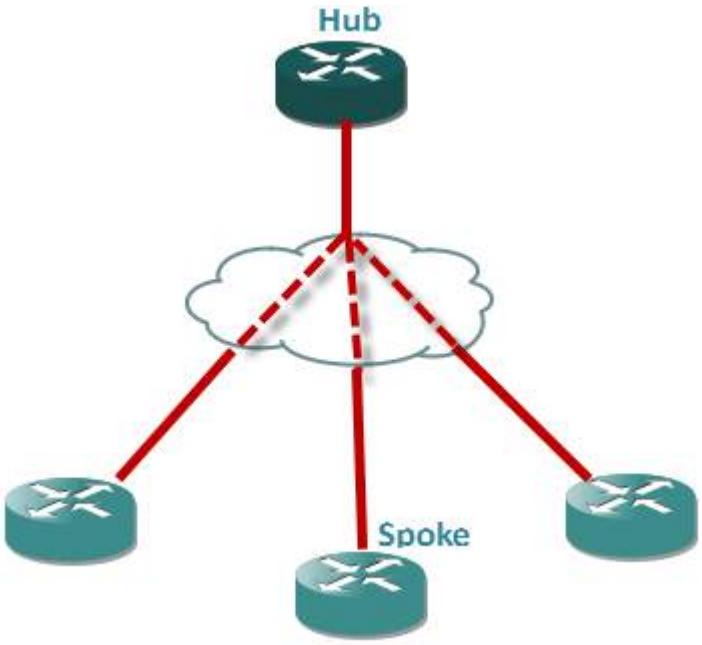
REMEMBER TO RESTORE EQUIPMENT TO ORIGINAL CONDITION

DMVPN

OBJECTIVE

1. Configure the crypto and tunnel on the Spoke router & HUB router, which effectively deploys DMVPN between the Spoke and Hub.

2. Demonstrate the various crypto and NHRP configurations, and the use of various "show" commands to troubleshoot and verify Spoke-Hub connectivity & tunnel formation.



192.168.0.0/24 **192.168.2.0/24**

192.168.1.0/24

Configure Phase 1 DMVPN tunnels using NHRP, IPSec, andmGRE, between 3 spokes and hub

Step 1.

Configure the outbound interface of the all the routers with public IPs.

Instead of the cloud use a Switch.

Use IPs as 100.100.x.0/24, where x is rack number. All outbound routers are in the same subnet

Check the connectivity. Report your outputs.

Step 2

Configure mGRE tunnel interface at the HUB as tunnel 1.

Proceed to configure NHRP on the HUB tunnel interface.

Check outputs of the

1. showdmvpn

2. showipnhrp

Report your observations if any.

Step 3

Create different internal networks on each of the spoke routers as described in the topology.

Create one GRE tunnel interface towards the hub on each of the Spoke routers.

Configure NHRP to simulate NBMA mapping on the spokes?

Report how the configuration on the spoke is different than Hub routers and why. Configure a routing protocol if necessary to bring the tunnels UP and get full connectivity across each site.

Check outputs of the following commands

1. showdmvpn

2. showipnhrp

3. showip route

Report your observations now and how it is different from Step 2, using "*debug ipnhrp*".

Connect a PC to Spoke 1 and ping Spoke 3 private IP subnet. Use wireshark and report a screenshot showing the actual NHRP packets. Do a traceroute to confirm the forwarding path.

Step 4

Configure IPSec on top for each tunnel to include for security.

Check outputs of the following commands and report them:

Show crypto isakmpsa

Show crypto ipsecsa

Make sure that the packets are encrypted now.

Verify the DMVPN operation at the spoke now:

Show dmvpn

REFERENCE:http://www.cisco.com/en/US/tech/tk583/tk372/technologies_configuration_example09186a008014bcd7.shtml