**Pacific Tech Center**

Today’s topic: Cisco Lab - GETVPN

Goals: Our current goal today is to create a GETVPN configuration.

1. Setup the basic VPN Router lab

Diagram

Description automatically generated

Text

Description automatically generatedGraphical user interface, text, application, chat or text message

Description automatically generated

1. Validate that the lab is working correctly
   1. The DMVPN cloud is represented by a switch, the IP address of the switch is .10 for any of the segments.
   2. The basic lab only has static routes for the different segments, you should not be able to reach the loopback addresses of the opposite routes at first.

KS1: ***ping 200.1.2.2***

KS1: ***ping 200.1.3.3***

KS1: ***ping 200.1.3.3***

R3: ***ping 200.1.4.4***

Note: If any of the above ping attempts do not work, try to ping the switch IP from each router and make sure that’s working.

1. Now let’s protect the traffic using IPSEC, first setup the ISAKMP and IPSEC policies for each router. For simplicity of this lab, we will use the same information for all routers.

For complete control and clarification of what policy is being used, let’s disable the default policies first

***no crypto isakmp default policy***

Now let’s continue with this lab:

***crypto isakmp policy 1***

***encryption aes 128***

***hash sha256***

***group 14***

***lifetime 400***

***authentication pre-share***

***crypto isakmp key cisco address 0.0.0.0***

***crypto ipsec transform-set xf1 esp-aes esp-sha-hmac***

***crypto ipsec profile pf1***

***set transform-set xf1***

***set security-association lifetime seconds 7200***

1. Key Server specific configuration

***! RSA keys to sign regkey msg***

***crypto key generate rsa general-keys label mykeys mod 2048 export***

***! Group ID, server, SA type***

***crypto gdoi group group1***

***identity number 123***

***server local***

***! if this is a large deployment add***

***!sa receive-only***

***rekey transport unicast***

***rekey authentication mypubkey rsa mykeys***

***! Unicast rekey, in a large deployment maybe multicast is better,***

***! in order to change to multicase, perform the following***

***!no rekay transport unicast***

***rekey lifetime seconds 86400***

***rekey retransmit 10 number 2***

***address ipv4 { link address of the KS router KS1 = 200.1.1.1 }***

***! Policy downloaded to GMs***

***sa ipsec 1***

***profile pf1***

***match address ipv4 GET-VPN-ACL***

***replay counter window-size 5***

***redundancy***

***local priority {10 for KS1, 5 for KS2}***

***peer address ipv4 { link address of the other KS router, KS2 = 200.1.2.2 }***

***! if you only want to encrypt traffic for a specific IP address range***

***!as many entries as you need, or just encrypt everything when going from***

***!one group member to another***

***!access-list 199 permit ip any any***

***! For this lab the encryption domain will be anytime something from 10.1.0.0/16***

***! communicates with 10.1.0.0/16***

***ip access-list ext GET-VPN-ACL***

***10 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255***

1. Each Group member would be configured with the following

***crypto gdoi group group1***

***identity number 123***

***server address ipv4 200.1.1.1***

***server address ipv4 200.1.2.2***

***crypto map CM1 10 gdoi***

***set group group1***

***crypto map CM1 local-address gi0/0***

***int gi0/0***

***crypto map CM1***

1. Let’s do some validation

***!Commands on the KS servers***

***show crypto gdoi ks members***

***show crypto gdoi ks rekey***

***show crypto gdoi detail***

***! On Router 3 or 4***

***show crypto ipsec sa***

***show crypto gdoi ks coop***

***show crypto gdoi group gp1 ks coop version***

***! Commands on the GM routers***

***show crypto gdoi group gp1 ipsec sa***

***clear crypto gdoi group gp1 replay***

***debug crypto gdoi replay***

***show crypto gdoi group gp1***

***show crypto gdoi group gp1 ks replay***

1. Now let’s check some things in relation to the encryption domain
   1. You should not yet be able to communicate from User1 to User2 because the routing was not setup.
   2. Verify that there are no packets going through the encryption domain, from R3

***R3#show crypto ipsec sa | i local|remote|#pkts***

***Crypto map tag: CM1, local addr 200.1.3.3***

***local ident (addr/mask/prot/port): (10.1.0.0/255.255.0.0/0/0)***

***remote ident (addr/mask/prot/port): (10.1.0.0/255.255.0.0/0/0)***

***#pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0***

***#pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0***

***#pkts compressed: 0, #pkts decompressed: 0***

***#pkts not compressed: 0, #pkts compr. failed: 0***

***#pkts not decompressed: 0, #pkts decompress failed: 0***

***local crypto endpt.: 200.1.3.3, remote crypto endpt.: 0.0.0.0***

* 1. The MPLS cloud (switch) is preconfigured for eigrp 10 so let’s configure that

***R3:router eigrp 10***

***R3: network 10.1.0.0 0.0.255.255***

***R3: network 200.1.3.3 0.0.0.0***

***R3: eigrp router-id 200.1.3.3***

***R3: passive-interface gi0/1***

***R4:router eigrp 10***

***R4: network 10.1.0.0 0.0.255.255***

***R4: network 200.1.4.4 0.0.0.0***

***R3: eigrp router-id 200.1.4.4***

***R4: passive-interface gi0/1***

* 1. Now let’s go ahead and log into one of the two hosts and try to ping the other

***User1: ping 10.1.1.100***

* 1. Look at the encryption domain and you should now see packets encrypted/decrypted.

***R3#sh crypto ipsec sa | i local|remote|#pkts***

***Crypto map tag: CM1, local addr 200.1.3.3***

***local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)***

***remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)***

***#pkts encaps: 40438, #pkts encrypt: 40438, #pkts digest: 40438***

***#pkts decaps: 40438, #pkts decrypt: 40438, #pkts verify: 40438***

***#pkts compressed: 0, #pkts decompressed: 0***

***#pkts not compressed: 0, #pkts compr. failed: 0***

***#pkts not decompressed: 0, #pkts decompress failed: 0***

***local crypto endpt.: 200.1.3.3, remote crypto endpt.: 0.0.0.0***

1. This completes this lab