ASA Site-To-Site VPN Packet Tracer Lab

This Packet Tracer lab has been provided to help you gain a better understanding of Cisco ASA security appliance. Specify the configuring and troubleshooting of the ASA Site-To-Site VPN capability. The ability to configure and troubleshoot a Site-To-Site VPN using the Cisco ASA security appliance has become an essential part of a network engineer’s job as many networks today encompass multiple sites.

The inclusion of the ASA 5505 in the latest version 6.1.1 of Cisco’s Packet Tracer has allowed students studding for Cisco certification to model networks employing basic security using the ASA. The functionality of the ASA 5505 is limited in the above version of Packet Tracer due to two factors. First there is only a basic license available, this limits the DMZ capability. Second the command set is limited; there is no IP protocol available within the access-list command only TCP, UDP and ICMP. The nat command is limited to dynamic and static which does not allow the user to separate VPN traffic from the nat process. Additionally the show commands are limited and there is no debug command, this limits the ability to troubleshoot issues.

Before we begin our lab we need to get a better understanding of site-to site VPNs, and why do we need them? If we have two sites that are geography separated and we need to communicate between them we have two choices. First we can purchases a dedicated line between the two sites but this is very costly. Second we can use the Internet that we already have access to and employ a VPN. A site-to-site VPN is a Virtual Private Network that allow us to tunnel through the internet creating a private network connection been our two sites.

There are three things needed to the creation of Site-To-Site ISAKMP VPN.

1. Phase 1 has to complete.

2. Phase 2 has to complete.

3. The Traffic has to be allowed to pass.

# VPN Phase 1 (ISAKMP)

This stage brings up the first secure tunnel (eventually there will be three tunnels) and for it to establish the firewalls need to agree what they are going to do to bring up the tunnel, and then Secure the tunnel. Both firewalls need a matching Phase 1 Policy to continue.

## A Phase 1 policy consists of:

1. The Authentication method (either a pre shared key or an RSA signature is usual).

2. The Encryption method (DES, 3DES, AES, AES-192, or AES-256).

3. The Hashing Method (MD5 or SHA).

4. The Diffie Helman Group (1, 2 or 5 usually).

5. Lifetime (In seconds before phase 1 should be re-established - usually 86400 seconds [1 day]).

## MESSAGE 1

The Initiator sends policies that it proposes to use, for phase 1 to the other ASA.

## MESSAGE 2

Providing the responder has a matching policy it will accept one of those proposed by the initiator and send it back in message 2.

Now the two ends have agreed HOW they will establish phase 1, they then need to agree on a "Shared Key" both ends must use the same shared key, but the shared key can’t be sent between them because the network link is not secure. To do this they use a Diffie Hellman key exchange, this uses a mathematical process called modular exponentiation, a simple example of how that works (The math's involved in a real key exchange are much more complicated!).

# How Diffie Hellman works (simply)

Problem Site A and Site B need to use the same secret key (which will be a big long number). they can’t send that number to each other because if they do it will be seen.

## Solution:

Both sites pick a random number, and they have a common number, this common number can be passed between sites, in our example Site A chooses 4 and Site B chooses 5

Both sites use the common number and raise it by the power of the random number they are using so Site A arrives at 16 and Site B at 32.

The sites then send the number they have arrived at, to the other site.

Each site uses the other sites total and raises it to the power of their original random number, this results in them both having the same key, with only the numbers 2, 16 and 32 being passed between them.

# Back to our VPN Tunnel

The next two messages are the initiator and responder swapping their Diffie Hellman information, each side produces a DH Public Key, and mathematically computes a long number called a "Nonce"

## MESSAGE 3

The initiator generates a "Public Key" also called the DH Public Value or Xa It also generates a Nonce or Ni and sends both of them to the responder.

## MESSAGE 4

The responder generates a "Public Key" also called the DH Public Value or Xb It also generates a Nonce or Nr and sends both of them to the initiator.

At this point both the initiator and the responder can calculate the DH Shared secret key, they then use the DH Secret Key, the "Shared Secret" that is manually entered onto both peers, and the Nonce from the other peer to create 3 DIGITAL KEYS, and because of the nature of Diffie Hellman each end will produce the same keys.

Key 1 = SKEYID\_d Used to work out any future IPsec keying

Key 2 = SKEYID\_a Used for data integrity and authentication (IKE)

Key 3 = SKEYID\_e Used to encrypt all further IKE traffic.

## MESSAGE 5

The initiator now sends its ID to the responder (this is either its IP address or a hostname). It also sends a "Hash" this authenticates the initiator to the responder as its made from the SKEYID, the pre-shared key and other information only known to the two peers.

## MESSAGE 6

Message 6 is basically the mirror of Message 5, the responder sends it’s ID (IP or Hostname) Back the initiator with its "Hash" and authenticates itself back to the initiator.

At this point both peers recalculate the hash they have received from the other peer, and they should both come out the same, if this happens then the IKE SA's are established and phase 1 is complete.

# VPN Phase 2

Once Phase 1 has completed the second stage of the VPN can start. Like phase 1 this state also requires messages to be sent between the peers, IPsec usually executes in "Quick mode" this means that there are only 3 MESSAGES.

## MESSAGE 1

The Initiator sends another Hash to the responder, this is similar to the one used in phase 1 but also includes info within this message to guarantee integrity.

The Phase 2 proposal includes:

1. Encapsulation method either ESP or AH.

2. Hashing method (Integrity checking) either SHA-HMAC or MD5-HMAC.

3. Diffie Hellman Group (1, 2, or 5).

4. The SPI - This number is the LABEL for the end of the tunnel the initiator will use for outbound traffic.

Tunnel mode (Tunnel or Transport), A timeout in seconds is specified, as is the ID (usually the subnet of both ends of the tunnel).

## MESSAGE 2

The Responder replies with its own "Hash" with the accepted proposal and its own SPI for outgoing encrypted traffic from the responder, and finally its own Key Exchange Payload.

Once this is complete both peers generate new DH secret keys and combine them with the SKEYID\_d key from phase 1 to create keys for IPsec encryption.

## MESSAGE 3

The final Message is sent from imitator to responder, and serves to inform the responder that its previous message was received.

Once phase 2 is complete IPsec SA's have been established and the tunnel is up.

# Learning Objectives:

* Configure ISAKMP Policy
* Creating an IKEv1 Transform Set
* Configure an ACL for interesting traffic
* Define a Tunnel Group
* Create a Crypto Map and Apply It To an Interface

# Lab Tasks:

All router and server are already configured with ip address and routes and will require no further configuration.

## Basic configuration:

1. Configure the Vlan interfaces with the IP address and subnet mask as shown in the network diagram.
2. Configure DHCP on the inside interface as shown in the network diagram.
3. Configure a default route on each of the ASAs.
4. Enable DHCP on the local and remote PCs and insure they get a proper ip address.
5. On HQ-ASA configure the following objects:

**object network Remote-subnet**

**subnet 192.168.2.0 255.255.255.0**

**object network inside-subnet**

**subnet 192.168.1.0 255.255.255.0**

1. On BR-ASA configure the following objects:

**object network Remote-subnet**

**subnet 192.168.1.0 255.255.255.0**

**object network inside-subnet**

**subnet 192.168.2.0 255.255.255.0**

## Create The ISAKMP Policy:

First we need to define an ISAKMP policy. ISAKMP is used to establish the initial asymmetrically encrypted channels between the two endpoints so that they can securely negotiate a pair of one-way IPsec security associations (SAs).

1. Enter the following on the HQ and Branch ASA.

**crypto ikev1 enable outside**

**crypto ikev1 policy 1**

**encryption 3des**

**hash md5**

**group 2**

**lifetime 86400**

## Create the IPSEC Transform Set:

An IPsec transform set establishes the encryption and authentication (HMAC) methods to be employed by the IPsec SAs. While it is possible to enable several options, both sides of our VPN will be configured to support only 3des and md5. Our transform set is named VPN-TRANSFORM.

1. Enter the following on both ASA devices.

**crypto ipsec ikev1 transform-set VPN-TRANSFORM esp-3des esp-md5-hmac**

## Create an ACL to Match Traffic:

Next we need to create an access list to match plain (unencrypted) traffic which should be encrypted and routed through the IPsec tunnel between the two LANs. This access list will be referenced by the crypto map we'll create later. In the real world, crypto map ACLs can be quite complex. For our purposes, however, we only need to match traffic going between the 192.168.1.0/24 and 1192.168.2.0/24 networks.

1. On HQ-ASA enter the following ACL:

**access-list VPN-INTERESTING-TRAFFIC extended permit icmp object inside-subnet object Remote-subnet**

**access-list VPN-INTERESTING-TRAFFIC extended permit tcp object inside-subnet object Remote-subnet**

1. On BR-ASA enter the following ACL:

**access-list VPN-INTERESTING-TRAFFIC extended permit icmp object inside-subnet object Remote-subnet**

**access-list VPN-INTERESTING-TRAFFIC extended permit tcp object inside-subnet object Remote-subnet**

## Create The Tunnel Groups:

A tunnel group holds tunnel configuration parameters, namely the connection type and authentication method. Since we're using pre-shred key authentication, we need to name our tunnel group as the IP address of the remote peer. Also, notice that we must define the connection type (ipsec-l2l) before we can configure the pre-shared key.

1. Configure the HQ-ASA with the following, use cisco as the pre-shared-key:

**tunnel-group 209.165.200.234 type ipsec-l2l**

**tunnel-group 209.165.200.234 ipsec-attributes**

**ikev1 pre-shared-key cisco**

1. The tunnel group configuration on BR-ASA is identical except that its name changes to 209.165.200.226 (HQ-ASA's outside interface):

**tunnel-group 209.165.200.226 type ipsec-l2l**

**tunnel-group 209.165.200.226 ipsec-attributes**

**ikev1 pre-shared-key cisco**

## Create and Apply a Crypto Map:

Finally, we need to create a crypto map (named CRYPTO-MAP) to tie together the IPsec transform set, access list, and tunnel group configured in the previous steps. First we match LAN-to-LAN traffic using our access list:

1. Enter the following command line on HQ-ASA:

**crypto map CRYPTO-MAP 1 match address VPN-INTERESTING-TRAFFIC**

1. Now we set the VPN peer and IPsec transform set to use:

**crypto map CRYPTO-MAP 1 set peer 209.165.200.234**

**crypto map CRYPTO-MAP 1 set ikev1 transform-set VPN-TRANSFORM**

**crypto map CRYPTO-MAP 1 set security-association lifetime seconds 86400**

1. The corresponding crypto map on BR-ASA looks like this:

**crypto map CRYPTO-MAP 1 match address VPN-INTERESTING-TRAFFIC**

**crypto map CRYPTO-MAP 1 set peer 209.165.200.226**

**crypto map CRYPTO-MAP 1 set ikev1 transform-set VPN-TRANSFORM**

**crypto map CRYPTO-MAP 1 set security-association lifetime seconds 86400**

1. Finally we need to apply the crypto map to the outside interface on each firewall:

**crypto map CRYPTO-MAP interface outside**

1. Now let’s test our new site-to-site VPN. From the local PC enter ping –n 10 192.168.2.200.

**Note:** The ping was not successful why? Let’s see.

1. From the enable command prompt enter the following command.

**HQ-ASA#show crypto isakmp sa**

IKEv1 SAs:

Active SA: 1

Rekey SA: 0 (A tunnel will report 1 Active and 1 Rekey SA during rekey)

Total IKE SA: 1

1 IKE Peer: 209.165.200.234

Type : L2L Role : Initiator

Rekey : no State : QM\_IDLE

There are no IKEv2 SAs

**Note:** It appears that phase 1 of our tunnel is up.

1. From the enable command prompt enter the following command.

**HQ-ASA#show crypto ipsec sa**

interface: outside

Crypto map tag: CRYPTO-MAP, seq num: 1, local addr 209.165.200.226

permit icmp object inside-subnet object Remote-subnet

local ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/1/0)

remote ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/1/0)

current\_peer 209.165.200.234

#pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 0

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

#pkts compressed: 0, #pkts decompressed: 0

#pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0

#pre-frag successes: 0, #pre-frag failures: 0, #fragments created: 0

#PMTUs sent: 0, #PMTUs rcvd: 0, #decapsulated frgs needing reassembly: 0

#send errors 1, #recv errors 0

local crypto endpt.: 209.165.200.226/0, remote crypto endpt.:209.165.200.234/0

path mtu 1500, ip mtu, ipsec overhead 78, media mtu 1500

current outbound spi: 0x4F3C14DE(1329337566)

current inbound spi: 0x4B083C78(1329337566)

inbound esp sas:

spi: 0x4B083C78(1258830968)

transform: esp-3des esp-md5-hmac no compression

in use settings ={L2L, Tunnel, }

slot: 0, conn id: 2006, crypto map: CRYPTO-MAP

sa timing: remaining key lifetime (k/sec): (4525504/85875)

IV size: 16 bytes

replay detection support: N

Anti replay bitmap:

0x00000000 0x0000001F

outbound esp sas:

spi: 0x4F3C14DE(1329337566)

transform: esp-3des esp-md5-hmac no compression

in use settings ={L2L, Tunnel, }

slot: 0, conn id: 2007, crypto map: CRYPTO-MAP

sa timing: remaining key lifetime (k/sec): (4525504/85875)

IV size: 16 bytes

replay detection support: N

Anti replay bitmap:

0x00000000 0x00000001

**Note:** the local and remote address are correct also the packets are being encrypted but no packets are decrypted.

1. Repeat the above command on BR-ASA.

**BR-ASA#show crypto ipsec sa**

interface: outside

Crypto map tag: CRYPTO-MAP, seq num: 1, local addr 209.165.200.234

permit icmp object inside-subnet object Remote-subnet

local ident (addr/mask/prot/port): (192.168.2.0/255.255.255.0/1/0)

remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/1/0)

current\_peer 209.165.200.226

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

#pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 0

#pkts compressed: 0, #pkts decompressed: 0

#pkts not compressed: 0, #pkts comp failed: 0, #pkts decomp failed: 0

#pre-frag successes: 0, #pre-frag failures: 0, #fragments created: 0

#PMTUs sent: 0, #PMTUs rcvd: 0, #decapsulated frgs needing reassembly: 0

#send errors 0, #recv errors 0

local crypto endpt.: 209.165.200.234/0, remote crypto endpt.:209.165.200.226/0

path mtu 1500, ip mtu, ipsec overhead 78, media mtu 1500

current outbound spi: 0x4B083C78(1258830968)

current inbound spi: 0x4F3C14DE(1258830968)

inbound esp sas:

spi: 0x4F3C14DE(1329337566)

transform: esp-3des esp-md5-hmac no compression

in use settings ={L2L, Tunnel, }

slot: 0, conn id: 2006, crypto map: CRYPTO-MAP

sa timing: remaining key lifetime (k/sec): (4525504/84957)

IV size: 16 bytes

replay detection support: N

Anti replay bitmap:

0x00000000 0x0000001F

outbound esp sas:

spi: 0x4B083C78(1258830968)

transform: esp-3des esp-md5-hmac no compression

in use settings ={L2L, Tunnel, }

slot: 0, conn id: 2007, crypto map: CRYPTO-MAP

sa timing: remaining key lifetime (k/sec): (4525504/84957)

IV size: 16 bytes

replay detection support: N

Anti replay bitmap:

0x00000000 0x00000001

**Note:** the local and remote address are correct also the packets are not encrypted but packets are decrypted.

1. Now let’s examine our access-list. From HQ-ASA enter the following command.

**HQ-ASA#sh access-list**

access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096) alert-interval 300

access-list VPN-INTERESTING-TRAFFIC; 2 elements; name hash: 0x92da53ba

access-list VPN-INTERESTING-TRAFFIC line 1 extended permit icmp object inside-subnet object Remote-subnet(hitcnt=3) 0x9546e289

access-list VPN-INTERESTING-TRAFFIC line 1 extended permit icmp 192.168.1.0 255.255.255.0 192.168.2.0 255.255.255.0(hitcnt=0) 0xd2b2e154

access-list VPN-INTERESTING-TRAFFIC line 2 extended permit tcp object inside-subnet object Remote-subnet(hitcnt=0) 0xb2688ae3

access-list VPN-INTERESTING-TRAFFIC line 2 extended permit tcp 192.168.1.0 255.255.255.0 192.168.2.0 255.255.255.0(hitcnt=0) 0xcadde50a

1. Repeat the above command on BR-ASA.

BR-ASA#show access-list

access-list cached ACL log flows: total 0, denied 0 (deny-flow-max 4096) alert-interval 300

access-list VPN-INTERESTING-TRAFFIC; 2 elements; name hash: 0x0838491a

access-list VPN-INTERESTING-TRAFFIC line 1 extended permit icmp object inside-subnet object Remote-subnet(hitcnt=0) 0x14ac3011

access-list VPN-INTERESTING-TRAFFIC line 1 extended permit icmp 192.168.2.0 255.255.255.0 192.168.1.0 255.255.255.0(hitcnt=0) 0xc1e6b847

access-list VPN-INTERESTING-TRAFFIC line 2 extended permit tcp object inside-subnet object Remote-subnet(hitcnt=0) 0x97934b5a

access-list VPN-INTERESTING-TRAFFIC line 2 extended permit tcp 192.168.2.0 255.255.255.0 192.168.1.0 255.255.255.0(hitcnt=0) 0xe0db4d89

**Note:** This all seems to be correct so what is the problem? Well we have told the ASA’s what traffic to transfer and how to do it, but we have not told it what to do with traffic once we get it. To solve this problem we need to create another access-list and send the filtered data to the inside interface.

1. On the HQ-ASA enter the following commands:

access-list VPN-PRIVATE-TRAFFIC extended permit tcp object Remote-subnet object inside-subnet

access-list VPN-PRIVATE-TRAFFIC extended permit icmp object Remote-subnet object inside-subnet

!

access-group VPN-PRIVATE-TRAFFIC out interface inside

1. On the HQ-ASA enter the following commands:

access-list VPN-PRIVATE-TRAFFIC extended permit tcp object Remote-subnet object inside-subnet

access-list VPN-PRIVATE-TRAFFIC extended permit icmp object Remote-subnet object inside-subnet

!

access-group VPN-PRIVATE-TRAFFIC out interface inside

1. Now let’s try again to ping 192.168.2.200.

Note: Now the pings are successful.

This concludes this packet tracer lab, I hope you found is informative.