- IPSEC Site-to-Site VPNs on an IOS Router -

Configuring an ISAKMP Policy Set

The first step in creating an IPSEC Tunnel is to globally enable ISAKMP:

RouterA(config)# *crypto isakmp enable*

Next, we must create an **ISAKMP policy**, which defines the algorithms and protocols to use when exchanging keys **(IKE Phase 1)**. To create the ISAKMP policy:

RouterA(config)# *crypto isakmp policy 1*
RouterA(config-isakmp)# *encryption des*
RouterA(config-isakmp)# *hash sha*
RouterA(config-isakmp)# *authentication rsa-sig*
RouterA(config-isakmp)# *group 1*
RouterA(config-isakmp)# *lifetime 86400*

The *crypto isakmp policy* command established the policy. The *1* is a priority number, as we can have multiple ISAKMP policies. The *lower* the number, the *higher* priority the policy is.

The above values for each IKE Phase 1 parameter are the **default** values. The following table details every possible option:

Parameter	*Values*
Encryption	des, 3des, aes
Hash	md5, sha
Authentication	pre-share, rsa-sig
Group	1, 2, 5

To view all configured ISAKMP policies:

RouterA# *show crypto isakmp policy*

* * *

Configuring Authentication (Pre-Shared Keys)

If we had specified a **pre-shared key** for authentication, we must now create that key string, and point it to the remote host's IP address (or hostname):

RouterA(config)# *crypto isakmp policy 1*
RouterA(config-isakmp)# *authentication pre-share*

RouterA(config)# *crypto isakmp key MYKEY address 77.1.1.1*

or

RouterA(config)# *crypto isakmp key MYKEY hostname REMOTEHOST*

Remember, both the shared key, and the ISAKMP policy must match on both peers for a session to be established.

Pre-shared keys are the **simplest** method of authentication. Much more configuration is required when using **RSA Digital Signatures**.

Configuring Authentication (Digital Signatures)

To use digital certificates from a Certificate Authority server for authentication:

RouterA(config)# *crypto isakmp policy 1*
RouterA(config-isakmp)# *authentication rsa-sig*

Notice the *authentication* method is now set to *rsa-sig*, instead of *pre-share*.

Next, CA-specific parameters must be configured. Since certificates are time-sensitive, the time and date should be accurately set on the IOS device:

RouterA(config)# *clock timezone est -5*
RouterA(config)# *clock set 12:00:00 16 february 2006*

Next, the hostname and domain name should be configured for the IOS device, as this information is included in the blank certificate sent to the CA:

RouterA(config)# *hostname RouterA*
RouterA(config)# *ip domain-name MYDOMAIN.COM*

* * *

Configuring Authentication (Digital Signatures) (continued)

Next, the RSA key must be generated. This will not work without a hostname and domain name configured:

RouterA(config)# *crypto key generate rsa*

The name for the keys will be RouterA.MYDOMAIN.COM
How many bits in the modulus [512]: 512

The number of bits in the modulus indicates the strength of encryption for the created key. This number can range from 360 to 2048, with a default of 512. Please note, the larger the specified number, the longer it will take to generate the keys.

To view the newly created key:

RouterA# *show crypto key mypubkey rsa*

Next, the Certificate Authority must be identified:

RouterA(config)# *crypto ca identity MYCERTIFICATE*
RouterA(ca-identity)# *enrollment url http://MYURL/certsrv/mscep/mscep.dll*

The *enrollment url* command points to the specific *mscep.dll* file on the Windows Certificate server.

Next, the CA must be authenticated, to ensure its validity:

RouterA(config)# *crypto ca authenticate MYCERTIFICATE*

Finally, the certificate must digitally signed:

RouterA(config)# *crypto ca enroll MYCERTIFICATE*

Please Note: the router will prompt for a password during this enrollment process. A random password cannot be used. Open the above *mscep.dll* file in a web browser, and it will display the required alphanumeric password. **It is case sensitive!**

The certificate will either be rejected or accepted, depending on whether the correct password was inputted. To view the newly acquired certificate:

RouterA# *show crypto ca certificates*

The remaining configuration necessary is identical to configuring an IPSEC tunnel with pre-shared keys.

* * *

Configuring the IPSEC Transform Set

Next, we must create a **transform set policy**, which defines the security policy to apply to **traffic** during IKE Phase 2:

RouterA(config)# *crypto ipsec transform-set MYSET ah-sha-hmac esp-des*

ESP and AH can be used concurrently. The following table details every possible option:

AH Transforms	ESP Encryption Transforms	ESP Authentication Tranforms
ah-md5-hmac	*esp-des*	*esp-md5-hmac*
ah-sha-hmac	*esp-3des*	*esp-sha-hmac*
	esp-null	

Thus, if we solely want ESP, with 3DES for encryption, and md5 for authentication:

RouterA(config)# *crypto ipsec transform-set MYSET esp-3des esp-md5-hmac*

* * *

Specifying Interesting Traffic

Next, we must specify what traffic is **interesting**. In other words, we must identify what traffic can *initiate* the IPSEC tunnel, and can be *sent across* the tunnel. We utilize an access-list to identify interesting traffic:

RouterA(config)# *access-list 100 permit ip 192.168.1.0 0.0.0.255 10.1.1.0 0.0.0.255*

Thus, we are specifying that traffic from hosts on the 192.168.1.x network, destined to the 10.1.1.x network, can both initiate and utilize the IPSEC tunnel. We will reference this access-list later in our configuration.

The configured interesting traffic on RouterB will be the **exact reverse** of that configured on RouterA:

RouterB(config)# *access-list 100 permit ip 10.1.1.0 0.0.0.255 192.168.1.0 0.0.0.255*

Configuring the IPSEC Crypto Map

Next, we must create a **crypto map**, which defines all previously configured IPSEC SA parameters, including the interesting traffic, the SA peer, and the IKE transform-set.

RouterA(config)# *crypto map MYTUNNEL 1 ipsec-isakmp*
RouterA(config-crypto-map)# *match address 100*
RouterA(config-crypto-map)# *set security-association lifetime seconds 1800*
RouterA(config-crypto-map)# *set peer 77.1.1.1*
RouterA(config-crypto-map)# *set transform-set MYSET*

We apply this crypto map to an interface, which will allow the IPSEC communication process to begin:

RouterA(config)# *interface s0/0*
RouterA(config-if)# *crypto map MYTUNNEL*

To view the configured crypto map:

RouterA# *show crypto map*

* * *

GRE Tunnels over IPSec Tunnels

Generic Routing Encapsulation (GRE) is a tunneling protocol that does *not* perform security functions, such as encryption or hashing.

However, GRE supports protocols other than IP (such as IPX or Appletalk), and supports multicast traffic, including that of routing protocols (such as RIP, OSPF, or EIGRP). GRE masks the original IP header, and introduces its own IP and GRE headers.

To configure a basic GRE tunnel:

RouterA(config)# *interface Tunnel0*
RouterA(config-if)# *ip address 1.1.1.1 255.0.0.0*
RouterA(config-if)# *tunnel source s0/0*
RouterA(config-if)# *tunnel destination 77.1.1.1*

RouterB(config)# *interface Tunnel0*
RouterB(config-if)# *ip address 1.1.1.2 255.0.0.0*
RouterB(config-if)# *tunnel source s0/0*
RouterB(config-if)# *tunnel destination 66.1.1.1*

The *tunnel source*, on the local router, must be pointed to as the *tunnel destination*, on the remote router.

The multi-protocol functionality of GRE can be used in conjunction with the security functionality of IPSec. GRE is most often used with **transport mode** IPSec. GRE tunneling occurs *before* IPSec security functions are applied to a packet.

IPSec configuration remains the same, *except* for the access-list for interesting traffic, which will reference only GRE traffic.

(Reference: http://www.cisco.com/univercd/cc/td/doc/solution/p2pgre_x.pdf)

* * *

Troubleshooting IPSEC

Various commands can be used to troubleshoot both IPSEC and ISAKMP:

RouterA# *show crypto isakmp sa*
RouterA# *show crypto ipsec sa*
RouterA# *show crypto isakmp policy*
RouterA# *show crypto isakmp transform-set*
RouterA# *show crypto map*

RouterA# *debug crypto isakmp*
RouterA# *debug crypto ipsec*

To manually tear down an ISAKMP or IPSEC SA:

RouterA# *clear crypto isakmp*
RouterA# *clear crypto sa*

* * *