Configuring Internet Key Exchange for IPsec VPNs

This module describes how to configure the Internet Key Exchange (IKE) protocol for basic IP Security (IPsec) Virtual Private Networks (VPNs). IKE is a key management protocol standard that is used in conjunction with the IPsec standard. IPsec is an IP security feature that provides robust authentication and encryption of IP packets.

IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard.

IKE is a hybrid protocol, that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.)

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Creating IKE Policies

Restrictions for configuring AES IKE policy

- IPsec and long keys (the “k9” subsystem) must be supported.
- AES cannot encrypt IPsec and IKE traffic if an acceleration card is present.

```
enable
configure terminal
crypto isakmp policy 10
encryption aes 256
```

```
hash sha
authentication pre-share
group 14
end
```

Troubleshooting Tips

- Clear (and reinitialize) IPsec SAs by using the **clear crypto sa** EXEC command.

Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command in the Cisco IOS Security Command Reference.

- The default policy and default values for configured policies do not show up in the configuration when you issue the **show running-config** command. To display the default policy and any default values within configured policies, use the **show crypto isakmp policy** command.
- Any IPsec transforms or IKE encryption methods that the current hardware does not support should be disabled; they are ignored whenever an attempt to negotiate with the peer is made.

If a user enters an IPsec transform or an IKE encryption method that the hardware does not support, a warning message will be generated. These warning messages are also generated at boot time. When an encrypted card is inserted, the current configuration is scanned. If any IPsec transforms or IKE encryption methods are found that are not supported by the hardware, a warning message will be generated.

Configuring IKE Authentication

After you have created at least one IKE policy in which you specified an authentication method (or accepted the default method), you need to configure an authentication method. IKE policies cannot be used by IPsec until the authentication method is successfully configured.

Note Before configuring IKE authentication, you must have configured at least one IKE policy, which is where the authentication method was specified (or RSA signatures was accepted by default).

To configure IKE authentication, you should perform one of the following tasks, as appropriate:

Configuring RSA Keys Manually for RSA Encrypted Nonces

Note This task can be performed only if a CA is not in use.

```
enable
configure terminal
crypto key generate rsa general-keys modulus 360
crypto key generate ec keysize 256 label Router_1_Key
end
```

Optional Configuration using Named Key

```
enable
configure terminal
crypto key pubkey-chain rsa
named-key otherpeer.example.com
address 10.5.5.1
key-string
00302017 4A7D385B 1234EF29 335FC973
2DD50A37 C4F4B0FD 9DADE748 429618D5
18242BA3 2EDFBDD3 4296142A DDF7D3D8
08407685 2F2190A0 0B43F1BD 9A8A26DB
07953829 791FCDE9 A98420F0 6A82045B
90288A26 DBC64468 7789F76E EE21
quit
```

Optional Configuration using Addresses Key

```
enable
configure terminal
crypto key pubkey-chain rsa
addressed-key 10.1.1.2 encryption
address 10.5.5.1
key-string
00302017 4A7D385B 1234EF29 335FC973
2DD50A37 C4F4B0FD 9DADE748 429618D5
18242BA3 2EDFBDD3 4296142A DDF7D3D8
08407685 2F2190A0 0B43F1BD 9A8A26DB
07953829 791FCDE9 A98420F0 6A82045B
90288A26 DBC64468 7789F76E EE21
quit
```

Configuring Preshared Keys

Note Preshared keys do not scale well with a growing network.

Restrictoins for Mask Preshared Keys

- The SA cannot be established between the IPsec peers until all IPsec peers are configured for the same preshared key.
- The mask preshared key must be distinctly different for remote users requiring varying levels of authorization. A new preshared key should be configured for each level of trust and correct keys must be assigned to the correct parties. Otherwise, an untrusted party may obtain access to protected data.

```
enable
configure terminal
crypto isakmp identity address
crypto isakmp key sharedkeystring address 192.168.1.33 no-xauth
crypto isakmp key sharedkeystring address 10.0.0.1
end
```

Configuring IKE Mode Configuration

```
enable
configure terminal
ip local pool pool1 172.16.23.0 172.16.23.255
```

```
crypto isakmp client configuration address-pool local pool1
end
```

Configuration Examples for an IKE Configuration

Example: Creating IKE Policies

This section contains the following examples, which show how to configure an AES IKE policy and a 3DES IKE policy.

Note Cisco no longer recommends using 3DES; instead, you should use AES. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Example: Creating an AES IKE Policy

The following example is sample output from the **show running-config**command. In this example, the AES 256-bit key is enabled.

```
Current configuration : 1665 bytes
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname "Router1"
!
!
ip subnet-zero
!
!
no ip domain lookup
!
ip audit notify log
ip audit po max-events 100
!
crypto isakmp policy 10
 encryption aes 256
 authentication pre-share
 lifetime 180
crypto isakmp key cisco123 address 10.0.110.1
!
!
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport

.
.
.
```

Example: Creating 3DES IKE Policies

This example creates two IKE policies, with policy 15 as the highest priority, policy 20 as the next priority, and the existing default priority as the lowest priority. It also creates a preshared key to be used with policy 20 with the remote peer whose IP address is 192.168.224.33.

```
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
!
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
!
crypto isakmp key 1234567890 address 192.168.224.33
```

In the example, the encryption DES of policy default would not appear in the written configuration because this is the default value for the encryption algorithm parameter.

If the **show crypto isakmp policy**command is issued with this configuration, the output is as follows:

```
Protection suite priority 15
encryption algorithm:3DES - Triple Data Encryption Standard (168 bit keys)
hash algorithm:Message Digest 5
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#2 (1024 bit)
lifetime:5000 seconds, no volume limit
Protection suite priority 20
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:preshared Key
Diffie-Hellman group:#1 (768 bit)
lifetime:10000 seconds, no volume limit
Default protection suite
encryption algorithm:DES - Data Encryption Standard (56 bit keys)
hash algorithm:Secure Hash Standard
authentication method:Rivest-Shamir-Adleman Signature
Diffie-Hellman group:#1 (768 bit)
lifetime:86400 seconds, no volume limit
```

Note that although the output shows “no volume limit” for the lifetimes, you can configure only a time lifetime (such as 86,400 seconds); volume-limit lifetimes are not configurable.

Example: Configuring IKE Authentication

The following example shows how to manually specify the RSA public keys of two IPsec peer-- the peer at 10.5.5.1 uses general-purpose keys, and the other peer uses special-usage keys:

```
crypto key pubkey-chain rsa
 named-key otherpeer.example.com
 address 10.5.5.1
 key-string
 005C300D 06092A86 4886F70D 01010105
 00034B00 30480241 00C5E23B 55D6AB22
 04AEF1BA A54028A6 9ACC01C5 129D99E4
 64CAB820 847EDAD9 DF0B4E4C 73A05DD2
 BD62A8A9 FA603DD2 E2A8A6F8 98F76E28
 D58AD221 B583D7A4 71020301 0001
 quit
 exit
 addressed-key 10.1.1.2 encryption
 key-string
 00302017 4A7D385B 1234EF29 335FC973
 2DD50A37 C4F4B0FD 9DADE748 429618D5
 18242BA3 2EDFBDD3 4296142A DDF7D3D8
 08407685 2F2190A0 0B43F1BD 9A8A26DB
 07953829 791FCDE9 A98420F0 6A82045B
 90288A26 DBC64468 7789F76E EE21
 quit
 exit
 addressed-key 10.1.1.2 signature
 key-string
 0738BC7A 2BC3E9F0 679B00FE 53987BCC
 01030201 42DD06AF E228D24C 458AD228
 58BB5DDD F4836401 2A2D7163 219F882E
 64CE69D4 B583748A 241BED0F 6E7F2F16
 0DE0986E DF02031F 4B0B0912 F68200C4
 C625C389 0BFF3321 A2598935 C1B1
 quit
 exit
 exit
```

Verifying IKE Policies

```
Router# show crypto isakmp policy

Global IKE policy
Protection suite of priority 1
        encryption algorithm:   Three key triple DES
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #5 (1536 bit)
        lifetime:               86400 seconds, no volume limit
```

Verifying RSA Keys

```
Router# show crypto key pubkey-chain rsa
Codes: M - Manually configured, C - Extracted from certificate

Code Usage         IP-Address/VRF         Keyring          Name
C    Signing                            default         cn=Cisco Licensing Root CA,o=Cisco
C    Signing                              default          cn=CA
C    Signing                              default          cn=Cisco Root CA M1,o=Cisco
C    Signing                              default          cn=Cisco Root CA 2048,o=Cisco
Systems
C    Signing                             default         cn=Cisco Manufacturing CA,o=Cisco
 Systems
C    Signing                              default          ou=Class 3 Public Primary
Certification Authority,o=VeriSign, Inc.,c=US
C    Signing                              default          cn=Cisco Root CA M2,o=Cisco
C    Signing                              default          cn=Cisco Manufacturing CA
SHA2,o=Cisco
C    Signing                              default          cn=Licensing Root - DEV,o=Cisco
```