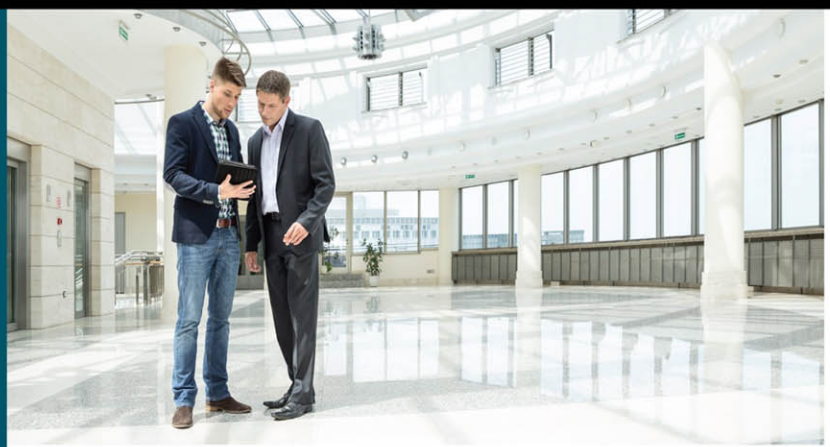

Cisco Intelligent WAN (IWAN)

Brad Edgeworth, CCIE No. 31574
Jean Marc Barozet
David Prall, CCIE No. 6508
Anthony Lockhart
Nir Ben-Dvora

ciscopress.com

Cisco Intelligent WAN (IWAN)

Brad Edgeworth, CCIE No. 31574
Jean-Marc Barozet
David Prall, CCIE No. 6508
Anthony Lockhart
Nir Ben-Dvora

Cisco Press
800 East 96th Street
Indianapolis, Indiana 46240 USA

Step 3. **Configure the local router's identity (optional).**

The local router's identity can be set based on an IP address with the command **identity local address** *ip-address*.

This command is not needed for pre-shared key authentication but is very helpful with the deployment of PKI authentication. The IP address specified should match the IP address used when registering the certificate (the recommended Loopback0 IP address).

Step 4. **Identify the FVRF for the tunnel end.**

The FVRF must be associated to the IKEv2 profile with the command **match fvrf** {*vrf-name* | **any**}. Using the **any** keyword allows either FVRF to be selected.

Step 5. **Define the local authentication method.**

The authentication method must be defined for connection requests that are received by remote peers. The command **authentication local** {**pre-share** | **rsa-sig**} defines the local authentication. Only one local authentication can be selected. The **pre-share** keyword is for pre-shared static keys, and **rsa-sig** is used for certificate-based authentication.

Step 6. **Define the remote authentication method.**

The authentication method must be defined for connection requests that are sent to remote peers. The command **authentication remote** {**pre-share** | **rsa-sig**} defines the remote authentication. Multiple remote authentication methods can be defined by repeating the command. The **pre-share** keyword is for pre-shared static keys, and **rsa-sig** is used for certificate-based authentication.

Step 7. **Define the IKEv2 keyring (required for pre-shared authentication).**

Pre-shared authentication requires that the IKEv2 keyring be associated to the IKEv2 profile. The command **keyring local** *keyring-name* associates the IKEv2 keyring.

Example 5-2 provides a sample IKEv2 profile that uses pre-shared key authentication.

Example 5-2 *Sample IKEv2 Profile*

```
crypto ikev2 profile DMVPN-IKE-PROFILE-INET
 match fvrf INET01
 match identity remote address 0.0.0.0
 authentication remote pre-share
 authentication local pre-share
 keyring local DMVPN-KEYRING-INET
```

The IKEv2 profile settings are displayed with the command **show crypto ikev2 profile** as shown in Example 5-3. Notice that the authentication, FVRF, IKE keyring, and identity IP address are displayed along with the IKE lifetime.

Example 5-3 *Display of IKEv2 Profile Settings*

```
R12-DC1-Hub2# show crypto ikev2 profile
IKEv2 profile: DMVPN-IKE-PROFILE-INET
 Ref Count: 1
 Match criteria:
  Fvrf: INET01
  Local address/interface: none
  Identities:
   address 0.0.0.0
  Certificate maps: none
 Local identity: none
 Remote identity: none
 Local authentication method: pre-share
 Remote authentication method(s): pre-share
 EAP options: none
 Keyring: DMVPN-KEYRING-INET
 Trustpoint(s): none
 Lifetime: 86400 seconds
 DPD: disabled
 NAT-keepalive: disabled
 Ivrf: none
 Virtual-template: none
 mode auto: none
 AAA AnyConnect EAP authentication mlist: none
 AAA EAP authentication mlist: none
 AAA Accounting: none
 AAA group authorization: none
 AAA user authorization: none
```

IPsec Transform Set

The transform set identifies the security protocols (ESP) for encrypting traffic. It specifies the protocol ESP or authentication header that is used to authenticate the data.

Table 5-1 provides a matrix of common IPsec transforms that can be inserted into a transform set. Following are some guidelines:

- Select an ESP encryption transform for data confidentiality.
- Select an authentication header or ESP authentication transform for data confidentiality.

Table 5-1 *IPsec Transform Matrix*

Transform Type	Transform	Description
ESP encryption	esp-aes 128	ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm
	esp-aes 192	ESP with the 192-bit AES encryption algorithm
	esp-aes 256	ESP with the 256-bit AES encryption algorithm
	esp-gcm 128	ESP transform using the Galois Counter Mode (GCM) 128-bit cipher
	esp-gcm 192	ESP transform using the GCM 192-bit cipher
	esp-gcm 256	ESP transform using the GCM 256-bit cipher (next-generation encryption)
ESP authentication	esp-sha-hmac	ESP with the Secure Hash Algorithm (SHA) (HMAC variant) authentication algorithm
	esp-sha256-hmac	ESP with the 256-bit SHA2 (HMAC variant) authentication algorithm
	esp-sha384-hmac	ESP with the 384-bit SHA2 (HMAC variant) authentication algorithm
	esp-sha512-hmac	ESP with the 512-bit SHA2 (HMAC variant) authentication algorithm
Authentication header authentication	ah-md5-hmac	Authentication header with the MD5 (Message Digest Algorithm 5) authentication algorithm
	ah-sha-hmac	Authentication header with the SHA authentication algorithm

The transform set is created with the following steps:

Step 1. **Create the transform set and identify the transforms.**

The transform set and identification of transforms are accomplished with one command. Only one transform set can be selected for ESP encryption, ESP authentication, and authentication header authentication. The command is **crypto ipsec transform-set** *transform-set-name* [*esp-encryption-name*] [*esp-authentication-name*] [*ah-authentication-name*].

Suggested transform set combinations are

```
esp-aes 256 and esp-sha-hmac
esp-aes and esp-sha-hmac
```

Step 2. **Specify the ESP mode.**

The ESP mode is configured with the command **mode** {**transport** | **tunnel**}. The ESP tunnel mode is the default mode and does not provide any benefits while adding 20 bytes of overhead per packet. Use the ESP mode of transport.

Example 5-4 provides a sample IPsec transform set.

Example 5-4 *Sample IPsec Transform Set*

```
crypto ipsec transform-set AES256/SHA/TRANSPORT esp-aes 256 esp-sha-hmac
 mode transport
```

The transform set can be verified with the command **show crypto ipsec transform-set** as shown in Example 5-5.

Example 5-5 *Verification of the IPsec Transform Set*

```
R12-DC1-Hub2# show crypto ipsec transform-set
! Output omitted for brevity
Transform set AES256/SHA/TRANSPORT: { esp-256-aes esp-sha-hmac  }
   will negotiate = { Transport,  },
```

IPsec Profile

The IPsec profile combines the IPsec transform set and the IKEv2 profile. The IPsec profile is created with the following steps:

Step 1. **Create the IPsec profile.**

The IPsec profile is created with the command **crypto ipsec profile** *profile-name*. The configuration context is then placed in IPsec profile configuration submode.

Step 2. **Specify the transform set.**

The transform set is specified with the command **set transform-set** *transform-set-name*.

Step 3. **Specify the IKEv2 profile.**

The IKEv2 profile is specified with the command **set ikev2-profile** *ike-profile-name*.

Example 5-6 provides a sample IPsec profile configuration.

Example 5-6 *Sample IPsec Profile*

```
crypto ipsec profile DMVPN-IPSEC-PROFILE-INET
 set transform-set AES256/SHA/TRANSPORT
 set ikev2-profile DMVPN-IKE-PROFILE-INET
```

The command **show crypto ipsec profile** displays the components of the IPsec profile as shown in Example 5-7.

Example 5-7 *Verification of the IPsec Profile*

```
R12-DC1-Hub2# show crypto ipsec profile
! Output omitted for brevity
IPSEC profile DMVPN-IPSEC-PROFILE-INET
        IKEv2 Profile: DMVPN-IKE-PROFILE-INET
        Security association lifetime: 4608000 kilobytes/3600 seconds
        Responder-Only (Y/N): N
        PFS (Y/N): N
        Mixed-mode : Disabled
        Transform sets={
                AES256/SHA/TRANSPORT:  { esp-256-aes esp-sha-hmac  } ,
```

Encrypting the Tunnel Interface

Now that all the required IPsec components have been configured, the IPsec profile is associated to the DMVPN tunnel interface with the command **tunnel protection ipsec profile** *profile-name* [**shared**].

The **shared** keyword is required for routers that terminate multiple encrypted DMVPN tunnels on the same transport interface. The command shares the IPsec *security association database (SADB)* among multiple DMVPN tunnels. Because the SADB is shared, a unique tunnel key must be defined on each DMVPN tunnel interface to ensure that the encrypted/decrypted traffic aligns to the proper DMVPN tunnel.

Note The topology in this book does not terminate multiple DMVPN tunnels on the same transport interface. The **shared** keyword is not required, nor is the tunnel key.

Example 5-8 provides a sample configuration for encrypting a DMVPN tunnel interface. After the configuration in this section is applied to R12, R22, R31, R41, and R52, the DMVPN tunnels are protected with IPsec.

Example 5-8 *Enabling IPsec Tunnel Protection*

```
interface Tunnel200
 tunnel protection ipsec profile DMVPN-IPSEC-PROFILE-INET
```

IPsec Packet Replay Protection

The Cisco IPsec implementation includes an anti-replay mechanism that prevents intruders from duplicating encrypted packets by assigning a unique sequence number to each encrypted packet. When a router decrypts the IPsec packets, it keeps track of the packets it has received. The IPsec anti-replay service rejects (discards) duplicate packets or old packets.

The router identifies acceptable packet age according to the following logic. The router maintains a sequence number window size (default of 64 packets). The minimum sequence number is defined as the highest sequence number for a packet minus the window size. A packet is considered of age when the sequence number is between the minimum sequence number and the highest sequence number.

At times, the default 64-packet window size is not adequate. Encryption is where the sequence number is set, and this happens before any QoS policies are processed. Packets can be delayed because of QoS priorities, resulting in out-of-order packets (low-priority packets are queued, whereas high-priority packets are immediately forwarded). The sequence number increases on the receiving router because the high-priority packets shift the window ahead, and when the lower-priority packets arrive later, they are discarded.

Increasing the anti-replay window size has no impact on throughput or security. An additional 128 bytes per incoming IPsec SA are needed to store the sequence number on the decryptor. The window size is increased globally with the command **crypto ipsec security-association replay window-size** *window-size*. Cisco recommends using the largest window size possible for the platform, which is 1024.

Dead Peer Detection

When two routers establish an IPsec VPN tunnel between them, it is possible that connectivity between the two routers can be lost for some reason. In most scenarios, IKE and IPsec do not natively detect a loss of peer connectivity, which results in network traffic being blackholed until the SA lifetime expires.

The use of *dead peer detection (DPD)* helps detect the loss of connectivity to a remote IPsec peer. When DPD is enabled in on-demand mode, the two routers check for connectivity only when traffic needs to be sent to the IPsec peer and the peer's liveliness is questionable. In such scenarios, the router sends a DPD R-U-THERE request to query the status of the remote peer. If the remote router does not respond to the R-U-THERE request, the requesting router starts to transmit additional R-U-THERE messages every retry interval for a maximum of five retries. After that the peer is declared dead.

DPD is configured with the command **crypto ikev2 dpd** [*interval-time*] [*retry-time*] **on-demand** in the IKEv2 profile. As a general rule, the interval time is set to twice that of the routing protocol timer (2*20), and the retry interval is set to 5 seconds. In essence, the total time is (2*20(*routing-protocol*)) + (5*5(*retry-count*)) = 65 seconds. This exceeds the holdtime of the routing protocol and engages only when the routing protocol is not operating properly.