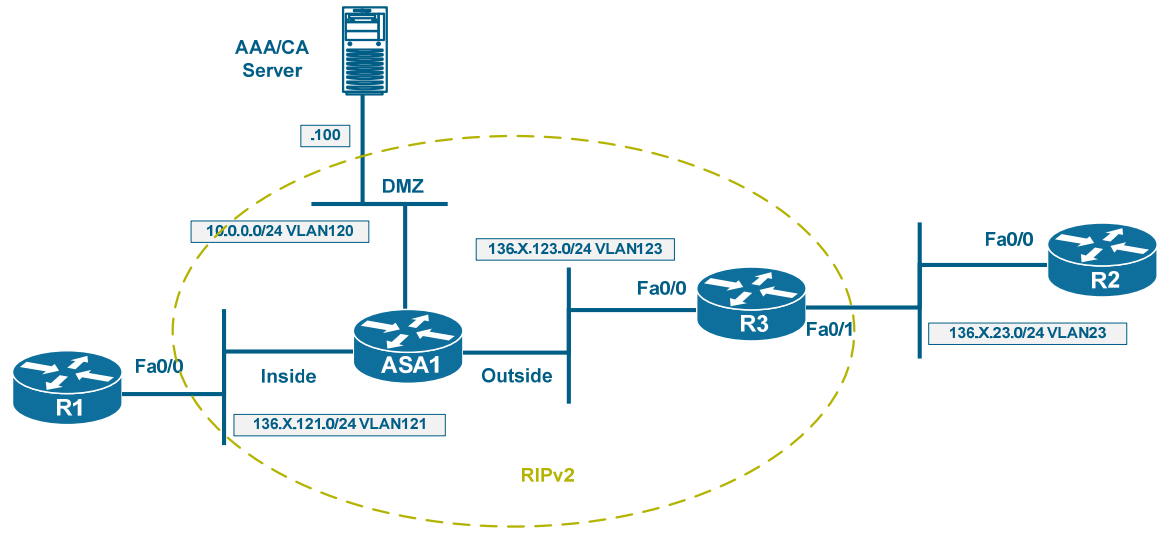


VPN

Note

Load the *IOS and ASA VPN* files to initialize your rack. Use the following diagram as your reference when working with the tasks below.



1.1 LAN-to-LAN VPN between IOS and ASA

- Configure a LAN-to-LAN IPsec tunnel between ASA1 and R3 using the following information:
 - Phase 1 settings:
 - Use 3DES encryption.
 - Use MD5 hash.
 - Use the default DH group in the ASA.
 - Use Pre-Shared keys authentication.
 - Use 3DES/MD5 for traffic encryption and integrity validation respectively.
- Only protect traffic between VLAN23 and VLAN121 subnets.

1.2 IPsec and NAT Interaction in ASA Firewall

- Enable NAT in the ASA firewall and translate all inside addresses using the IP address of the outside interface.
- Ensure the VPN traffic is not affected by this configuration.

1.3 Authentication using Digital Signatures

- Reconfigure the L2L tunnel between ASA1 and R3 to use digital certificates for ISAKMP authentication.
- Use the following URL for CA enrollment:
<http://10.0.0.100:80/certsrv/mscep/mscep.dll> and use the domain name INE.com for certificate DNS

VPN Solutions

1.1 LAN-to-LAN VPN between IOS and ASA

- Configure a LAN-to-LAN IPsec tunnel between ASA1 and R3 using the following information:
 - Phase 1 settings:
 - Use 3DES encryption.
 - Use MD5 hash.
 - Use DH Group2.
 - Use Pre-Shared keys authentication.
 - Use 3DES/MD5 for traffic encryption and integrity validation respectively.
- Only protect traffic between VLAN23 and VLAN121 subnets.

Configuration

Note

LAN-to-LAN IPsec VPN involves two devices in security negotiation. The result of this negotiation is an agreement to encrypt a certain set of traffic between the two endpoints. The negotiations proceed in two phases:

1) **IPsec Phase 1:** Devices authenticate each other using any configured method, .e.g a pre-shared password, digital signatures and so on. The both parties have first to negotiate the authentication method.. During the authentication phase, devices exchange their identities (e.g. IP addresses, hostnames, digital certificates) and prove that they are themselves. Further, devices establish a secure channel the called ISAKMP SA (Security Association) which is used to protect any further management communications.

The core procedure for establishing a secure channel is Diffie-Hellman (DH) key exchange (KE). This procedure allows a pair of devices to derive a common shared encryption key without letting any side party to eavesdrop it. DH KE involves discrete calculations on certain cyclic group. IPsec settings allows you selecting the group number, with the larger group being slower in computations but more secure.

You may often the term ISAKMP (Internet Security Association Key Management Protocol) and the term IKE (Internet Key Exchange) to be used interchangeably.

However, this is not strictly correct, as ISAKMP is an abstract framework, while IKE is its actual implementation.

The IPsec Phase 1 may run in two modes: Main Mode and Aggressive Mode. The first mode utilizes six messages exchange procedure. Prior to exchanging device identities and authenticating them, Main Mode ensures the DH KE produces a shared encryption key to protect the authentication phase. Aggressive Mode uses just three messages to establish the ISAKMP SA. This mode exchange device identities in parallel with the shared encryption key generation. This is less secure, but has some unique advantages, when using pre-shared keys for authentication. As we see later, IKE Main Mode with pre-shared keys has some limitations, because identities are exchanged only after the channel has been secured.

2) **IPsec Phase 2:** Two endpoints agree on the traffic they are going to encrypt and the cipher/hash functions to use. Both parties exchange the so-called Proxy Identities, which are in essence access-lists defining the traffic that each side wants to encrypt. Both sides check that their Proxy IDs are non-conflicting, i.e. they don't define mismatching traffic sets. The endpoints agree on the mode of encryption, which is usually "tunnel" mode, when the endpoint prepends additional header to route the tunneled packet to another device. The additional header is called ESP or encapsulated Security Payload. This header contains the IP addresses of the source/destination VPN endpoints and the original packet is encrypted and hidden behind. There is an option to use the AH (Authenticated Header) encapsulation, which does not encrypt traffic but only checksums the content. Thus AH ensures integrity but not confidentiality, which is rather rarely needed.

IPsec Phase 2 has only one mode of operations, called Quick Mode. This mode uses three messages to establish the IPsec SA. All Quick Mode communications and negotiations are protected by the ISAKMP SA.

LAN-to-LAN VPN configuration in ASA firewall consists of the following steps:

1) Defining global ISAKMP (Phase 1) policy using the command `crypto isakmp policy <priority>`. You need to set the authentication method, the cipher to protect the ISAKMP SA and the hash function for integrity checks. Additionally, you may change the DH group number if you want, the default group is 2. After you have defined the ISAKMP policy, you should enable it on the interface where the VPN tunnel is to be terminated using the command `crypto isakmp enable <ifname>`. Notice that the default ISAKMP policy uses RSA-signatures for authentication, and that the policy-list is scanned from lower numbers to higher when matching the incoming proposals from the remote peer.

2) Create a tunnel-group for the LAN-to-LAN tunnel using the commands `tunnel-group <IP> type ipsec-l2l` and `tunnel-group <IP> ipsec-attributes`. The tunnel-group is essentially an object that defines the administrative policy to be applied to the LAN-to-LAN tunnel. It does NOT define the traffic to be protected; rather most attributes are related to IPsec Phase 1. Up to some point, it could be compared with ISAKMP profile concept in IOS routers. The tunnel group name must be the IP address of the remote endpoint, if you are using pre-shared keys for authentication. A symbolic name could be used in some cases, as we will see in separate tasks. When the firewall establishes a VPN tunnel it will look through the list of local tunnel-groups based on the remote endpoint IP address. At the very least, the tunnel group must specify the peer authentication information, such as pre-shared key, if the global ISAKMP policy uses pre-shared keys for authentication.

Notice that the concept of tunnel-group has been borrowed from VPN3000 concentrator series. The rest of the IPsec configuration in ASA firewall is very similar to Cisco IOS.

3) Define a crypto transform-set for IPsec Phase 2 using the command `crypto ipsec transform-set`. This command defines the security parameters for the IPsec tunnel, specifically the cipher, hash function and optionally the mode of the IPsec protection – tunnel or transport.

4) Define a subset of traffic for IPsec protection using an extended access-list. The syntax is `permit <local-ip> <local-mask> <remote-ip> <remote-mask>` and should mirror the entries configured in the remote endpoint.

5) Create a crypto-map using the command `crypto-map <NAME> <tag> {set | match}` that matches the above-created access-list, sets the remote peer and the transform-set. This completes the settings for IPsec Phase 2. Notice that setting the remote peer is important, since this is how the firewall binds the proxy IDs in the access-list to the tunnel group.

There are some optional parameters that are only supported by ASA firewall. You may use the command `crypto-map <NAME> <tag> set connection-type {answer-only|originate-only|bidirectional}` to specify the type of the VPN tunnel, similar to the types used in VPN3000. Answer-only entry will not attempt to initiate the VPN tunnel for outgoing traffic. Originate-only tunnel will not be established until there is outgoing traffic.

When you're done with the crypto-map configuration, apply the crypto-map to the interface where you expect the VPN tunnel to be terminated using the interface-level command `crypto-map`.

6) The last thing you may want to do is make sure that the command `sysopt connection permit-vpn` is enabled. When this option is enabled, the decrypted VPN traffic is NOT subject to access-list checks on the interface where the tunnel has terminated. For example, if you have this command disabled, and the tunnel terminates on the outside interface, then the decrypted traffic will be checked against the interface inbound access-list.

Now for the IOS part of IPSec configuration.

1) The first step is very similar to ASA configuration – you define an ISAKMP policy. Make sure you set the DH group to 2, when connecting to an ASA firewall (or configure the ASA firewall to use DH group 1) as the default DH group for IOS routers is group 1. Other settings must match the settings configured in the remote endpoint.

2) If you are using the pre-shared keys for authentication, you should define one using the global mode command `crypto isakmp key`. This differs from the tunnel-group settings used in the ASA firewall. In some advanced cases you may want to set additional Phase 1 settings using ISAKMP profiles. We will cover those in separate tasks.

3) Create a transform set, like you did in the ASA. Make sure the cipher and the hash match the values used in the ASA endpoint.

4) Create an extended access-list that defines the traffic to be encrypted. As usual, this access-list should mirror the access-list entries used in the remote endpoint.

5) Create a crypto map that matches the access-list created above, sets the peer IP address and configure the transform set to be applied to the traffic. This completes the configuration of IPSec Phase 2 settings.

As you can see, the configuration for ASA firewall and IOS router is very much similar. There are, however, differences, mostly related to the tunnel-group concept inherited by the ASA firewalls from VPN3000 concentrator code.

ASA1:

```
!  
! Configure & Enable ISAKMP policy  
!  
crypto isakmp policy 10  
  authentication pre-share  
  encryption 3des  
  hash md5  
!  
crypto isakmp enable outside  
  
!  
! Configure tunnel group for L2L tunnel  
!  
tunnel-group 136.1.123.3 type ipsec-l2l  
tunnel-group 136.1.123.3 ipsec-attributes  
  pre-shared-key CISCO  
  
!  
! Configure transform-set  
!  
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac  
  
!  
! Access-list to classify traffic for encryption  
!  
access-list VLAN121_TO_VLAN23 permit ip 136.1.121.0 255.255.255.0  
136.1.23.0 255.255.255.0  
  
!  
! Create a crypto-map  
!  
crypto map VPN 10 match address VLAN121_TO_VLAN23  
crypto map VPN 10 set peer 136.1.123.3  
crypto map VPN 10 set transform-set 3DES_MD5  
  
!  
! Apply crypto-map and enable VPN traffic to bypass ACLs  
!  
crypto map VPN interface outside  
sysopt connection permit-vpn
```

R3:

```
!  
! Configure ISAKMP policy  
!  
crypto isakmp policy 10  
  encryption 3des  
  auth pre-share  
  hash md5  
  group 2  
!  
crypto isakmp key CISCO address 136.1.123.12
```

```
!  
! Create transform-set  
!  
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac  
  
!  
! Create access-list to classify traffic for encryption  
!  
ip access-list extended VLAN23_TO_VLAN121  
  permit ip 136.1.23.0 0.0.0.255 136.1.121.0 0.0.0.255  
  
!  
! Create & apply crypto map  
!  
crypto map VPN 10 ipsec-isakmp  
  match address VLAN23_TO_VLAN121  
  set transform 3DES_MD5  
  set peer 136.1.123.12  
!  
interface FastEthernet 0/0  
  crypto map VPN
```

Verification

Note

To verify your configuration, send some traffic from R1 to R1's VLAN121 IP address.

```
Rack1R2#ping 136.1.121.1
```

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 136.1.121.1, timeout is 2 seconds:

```
.!!!!
```

Success rate is 80 percent (4/5), round-trip min/avg/max = 100/166/189 ms

Note

Now check the VPN tunnel stats in R3. First check the ISAKMP SA. Pay attention to the cipher/hash and the authentication mode used.

```
Rack1R3#show crypto isakmp sa detail
```

Codes: C - IKE configuration mode, D - Dead Peer Detection

K - Keepalives, N - NAT-traversal

X - IKE Extended Authentication

psk - Preshared key, rsig - RSA signature

renc - RSA encryption

C-id	Local	Remote	I-VRF	Encr	Hash	Auth	DH
	Lifetime	Cap.					
2	136.1.123.3	136.1.123.12		3des	md5	psk	2
	23:54:52						

Note

Now check IPsec Phase 2 SAs in R3. They should cover traffic between VLAN23 and VLAN121. Notice that the counters for encapsulated and de-capsulated packets are incrementing.

```
Rack1R3#show crypto ipsec sa
```

```
interface: Ethernet0/0
  Crypto map tag: VPN, local addr. 136.1.123.3

  protected vrf:
    local ident (addr/mask/prot/port): (136.1.23.0/255.255.255.0/0/0)
    remote ident (addr/mask/prot/port): (136.1.121.0/255.255.255.0/0/0)
    current_peer: 136.1.123.12:500
      PERMIT, flags={origin_is_acl,}
      #pkts encaps: 4, #pkts encrypt: 4, #pkts digest 4
      #pkts decaps: 4, #pkts decrypt: 4, #pkts verify 4
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts compr. failed: 0
      #pkts not decompressed: 0, #pkts decompress failed: 0
      #send errors 21, #rcv errors 0
```

Note

Notice the output below. It specifies the IPsec tunnel endpoints, which should be the IP addresses of the router and the ASA firewall. Further note that “Tunnel” mode is in use and ESP header (Encapsulated Security Payload) is used for packet tunneling. Make sure the transform set in the output matches the one required by the scenario.

If you are wondering about the meaning of “SPI” keyword, it stands for Security Parameters Index. This value is carried in IPsec header, and used by the receiving router to find the matching IPsec Phase 2 SA. Essentially, it is just an index in the array of SAs.

```
local crypto endpt.: 136.1.123.3, remote crypto endpt.: 136.1.123.12
  path mtu 1500, media mtu 1500
  current outbound spi: 482D0576

  inbound esp sas:
    spi: 0xB0A78AA3(2963770019)
      transform: esp-3des esp-md5-hmac ,
      in use settings = {Tunnel, }
      slot: 0, conn id: 2000, flow_id: 1, crypto map: VPN
      sa timing: remaining key lifetime (k/sec): (4455492/3285)
      IV size: 8 bytes
      replay detection support: Y

  inbound ah sas:

  inbound pcp sas:
```

```
outbound esp sas:
  spi: 0x482D0576(1210910070)
  transform: esp-3des esp-md5-hmac ,
  in use settings ={Tunnel, }
  slot: 0, conn id: 2001, flow_id: 2, crypto map: VPN
  sa timing: remaining key lifetime (k/sec): (4455492/3285)
  IV size: 8 bytes
  replay detection support: Y

outbound ah sas:

outbound pcp sas:
```

Note

Check ISAKMP SA status in the ASA firewall next. Notice the IKE Peer IP address and the "State" which should be "MM_ACTIVE" in the case of L2L tunnel (Main Mode, Active).

```
Rack1ASA1(config)# show crypto isakmp sa
```

```
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: 136.1.123.3
   Type      : L2L           Role      : responder
   Rekey     : no           State     : MM_ACTIVE
```

Note

Now check the IPsec Phase 2 SA in the ASA firewall. They should mirror the entries in the IOS router and have packet counters incrementing.

```
Rack1ASA1(config)# show cry ipsec sa
interface: outside
Crypto map tag: VPN, seq num: 10, local addr: 136.1.123.12

access-list VLAN121_TO_VLAN23 permit ip 136.1.121.0 255.255.255.0
136.1.23.0 255.255.255.0
local ident (addr/mask/prot/port):
(136.1.121.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port):
(136.1.23.0/255.255.255.0/0/0)
current_peer: 136.1.123.3
```

```
#pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
#pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 4, #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, #rcv errors: 0

local crypto endpt.: 136.1.123.12, remote crypto endpt.:
136.1.123.3

<snip>
```

Note

The next thing we're going to do is explore debugging output in the ASA firewall and the IOS router. We start with the firewall and clear all active IPsec sessions first. Then we enable ISAKMP and IPsec debugging in the ASA unit.

```
Rack1ASA1(config)# clear crypto isakmp
Rack1ASA1(config)# clear crypto ipsec sa

Rack1ASA1(config)# debug crypto isakmp 9
Rack1ASA1(config)# debug crypto ipsec 9

Rack1R2#ping 136.1.121.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 136.1.121.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/4/4 ms
```

Note

The following output demonstrates IKE main mode 6 messages exchange. The first portion of the debug output in the ASA shows the initial IKE message received from R3. Most important thing in this output is that the incoming SA proposal matches a local ISAKMP policy entry.

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE RECEIVED Message (msgid=0)
with payloads : HDR + SA (1) + VENDOR (13) + VENDOR (13) + VENDOR (13)
+ VENDOR (13) + NONE (0) total length : 164
%ASA-7-715047: IP = 136.1.123.3, processing SA payload
%ASA-7-713906: IP = 136.1.123.3, Oakley proposal is acceptable
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received NAT-Traversal RFC VID
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received NAT-Traversal ver 03 VID
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received NAT-Traversal ver 02 VID
%ASA-7-715047: IP = 136.1.123.3, processing IKE SA payload
%ASA-7-715028: IP = 136.1.123.3, IKE SA Proposal # 1, Transform # 1
acceptable Matches global IKE entry # 3
```

Note

The firewall prepares a response IKE packet, where it specifies the accepted policy and additional information.

```
%ASA-7-715046: IP = 136.1.123.3, constructing ISAKMP SA payload
%ASA-7-715046: IP = 136.1.123.3, constructing NAT-Traversal VID ver 02
payload
%ASA-7-715046: IP = 136.1.123.3, constructing Fragmentation VID +
extended capabilities payload
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE SENDING Message (msgid=0)
with payloads : HDR + SA (1) + VENDOR (13) + VENDOR (13) + NONE (0)
total length : 128
```

Note

The third message in row is the one received from R3 again. The most important field here is the KE (key-exchange header) which is used for Diffie-Hellman shared secret generation. The firewall processes the KE message and prepares a response. Notice that the firewall also processes NAT-D (NAT discovery) headers from the other node. Those headers contain the hashed values of the original IP addresses used by the initiator. This allows for detection of a NAT devices on the path between the two IPSec endpoints.

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE RECEIVED Message (msgid=0)
with payloads : HDR + KE (4) + NONCE (10) + VENDOR (13) + VENDOR (13) +
VENDOR (13) + VENDOR (13) + NAT-D (130) + NAT-D (130) + NONE (0) total
length : 296
%ASA-7-715047: IP = 136.1.123.3, processing ke payload
%ASA-7-715047: IP = 136.1.123.3, processing ISA_KE payload
%ASA-7-715047: IP = 136.1.123.3, processing nonce payload
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received Cisco Unity client VID
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received DPD VID
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715038: IP = 136.1.123.3, Processing IOS/PIX Vendor ID payload
(version: 1.0.0, capabilities: 00000f7f)
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received xauth V6 VID
%ASA-7-715047: IP = 136.1.123.3, processing NAT-Discovery payload
%ASA-7-713906: IP = 136.1.123.3, computing NAT Discovery hash
%ASA-7-715047: IP = 136.1.123.3, processing NAT-Discovery payload
%ASA-7-713906: IP = 136.1.123.3, computing NAT Discovery hash
%ASA-7-715046: IP = 136.1.123.3, constructing ke payload
%ASA-7-715046: IP = 136.1.123.3, constructing nonce payload
%ASA-7-715046: IP = 136.1.123.3, constructing Cisco Unity VID payload
%ASA-7-715046: IP = 136.1.123.3, constructing xauth V6 VID payload
%ASA-7-715048: IP = 136.1.123.3, Send IOS VID
%ASA-7-715038: IP = 136.1.123.3, Constructing ASA spoofing IOS Vendor
ID payload (version: 1.0.0, capabilities: 20000001)
%ASA-7-715046: IP = 136.1.123.3, constructing VID payload
%ASA-7-715048: IP = 136.1.123.3, Send Altiga/Cisco VPN3000/Cisco ASA GW
VID
```

 **Note**

In addition to generating a KE response, the local endpoint prepares its own NAT-D headers to be used in subsequent exchange.

```
%ASA-7-715046: IP = 136.1.123.3, constructing NAT-Discovery payload
%ASA-7-713906: IP = 136.1.123.3, computing NAT Discovery hash
%ASA-7-715046: IP = 136.1.123.3, constructing NAT-Discovery payload
%ASA-7-713906: IP = 136.1.123.3, computing NAT Discovery hash
```

 **Note**

Now a very important moment. In order to be able to generate the shared encryption key, the local endpoint must find the pre-shared key matching the remote peer. This is because the shared key is produced from the Diffie-Hellman generated key hashed with the pre-shared key configured for the remote endpoint. At this point of IKE exchange the firewall does not yet learned the IKE ID of the remote endpoint. Thus, the only way to find a matching pre-shared key is to scan all local tunnel groups based on the remote peer's IP address. This is a fundamental limitation of using the pre-shared keys for IKE Main Mode authentication – PSKs are always looked up based on IP addresses.

```
%ASA-7-713906: IP = 136.1.123.3, Connection landed on tunnel_group
136.1.123.3
%ASA-7-713906: Group = 136.1.123.3, IP = 136.1.123.3, Generating keys
for Responder...
```

 **Note**

We send our response to the peer. At this moment, the encrypted channel has been established, and the following exchange is fully protected.

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE SENDING Message (msgid=0)
with payloads : HDR + KE (4) + NONCE (10) + VENDOR (13) + VENDOR (13) +
VENDOR (13) + VENDOR (13) + NAT-D (130) + NAT-D (130) + NONE (0) total
length : 296
```

 **Note**

Now we receive the fifth message from our peer, containing its IKE ID. Formally, this message is used for authentication, based on the peer's ID. However, due to the nature of the shared key generation, the remote party has been already authenticated. Thus, the remote IKE ID is simply ignored in case of IKE Main Mode with PSK authentication.

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE RECEIVED Message (msgid=0)
with payloads : HDR + ID (5) + HASH (8) + NOTIFY (11) + NONE (0) total
length : 88
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing ID
payload
%ASA-7-714011: Group = 136.1.123.3, IP = 136.1.123.3, ID_IPV4_ADDR ID
received
136.1.123.3
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing hash
payload
%ASA-7-715076: Group = 136.1.123.3, IP = 136.1.123.3, Computing hash
for ISAKMP
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing notify
payload
```

 **Note**

Also, now both devices know if there is any NAT box in the path or not, based on the preceding NAT-D exchange.

```
%ASA-6-713172: Group = 136.1.123.3, IP = 136.1.123.3, Automatic NAT
Detection Status: Remote end is NOT behind a NAT device This
end is NOT behind a NAT device
%ASA-7-713906: IP = 136.1.123.3, Connection landed on tunnel_group
136.1.123.3
%ASA-4-713903: Group = 136.1.123.3, IP = 136.1.123.3, Freeing
previously allocated memory for authorization-dn-attributes
%ASA-7-715046: Group = 136.1.123.3, IP = 136.1.123.3, constructing ID
payload
%ASA-7-715046: Group = 136.1.123.3, IP = 136.1.123.3, constructing hash
payload
%ASA-7-715076: Group = 136.1.123.3, IP = 136.1.123.3, Computing hash
for ISAKMP
%ASA-7-715034: IP = 136.1.123.3, Constructing IOS keep alive payload:
proposal=32767/32767 sec.
%ASA-7-715046: Group = 136.1.123.3, IP = 136.1.123.3, constructing dpd
vid payload
```

 **Note**

The local endpoint sends its own IKE ID to the peer along with other information. This finishes the 6-message Main Mode exchange.

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE SENDING Message (msgid=0)
with payloads : HDR + ID (5) + HASH (8) + IOS KEEPALIVE (128) + VENDOR
(13) + NONE (0) total length : 92
%ASA-6-113009: AAA retrieved default group policy (DfltGrpPolicy) for
user = 136.1.123.3
%ASA-5-713119: Group = 136.1.123.3, IP = 136.1.123.3, PHASE 1 COMPLETED
%ASA-7-713121: IP = 136.1.123.3, Keep-alive type for this connection:
DPD
%ASA-7-715080: Group = 136.1.123.3, IP = 136.1.123.3, Starting P1 rekey
timer: 82080 seconds.
```

 **Note**

Now it's time to run IPsec Phase 2 negotiations (running in Quick Mode, QM) and establish the IPsec SA. We receive the initial proposal from our peer. This proposal contains the SA payload, that describes the security policy (cipher, hash) and the KE message, which is a part of the new DH exchange, to generate the new encryption key.

```
%ASA-7-714003: IP = 136.1.123.3, IKE Responder starting QM: msg id =
c244ac78
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE RECEIVED Message
(msgid=c244ac78) with payloads : HDR + HASH (8) + SA (1) + NONCE (10) +
ID (5) + ID (5) + NONE (0) total length : 164
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing hash
payload
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing SA
payload
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing nonce
payload
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing ID
payload
```

 **Note**

The first packet from the initiator also contains the ID payload. This payload describes the Proxy IDs that the remote end is willing to protect.

```
%ASA-7-714011: Group = 136.1.123.3, IP = 136.1.123.3,
ID_IPV4_ADDR_SUBNET ID received--136.1.23.0--255.255.255.0
%ASA-7-713035: Group = 136.1.123.3, IP = 136.1.123.3, Received remote
IP Proxy Subnet data in ID Payload:  Address 136.1.23.0, Mask
255.255.255.0, Protocol 0, Port 0
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing ID
payload
%ASA-7-714011: Group = 136.1.123.3, IP = 136.1.123.3,
ID_IPV4_ADDR_SUBNET ID received--136.1.121.0--255.255.255.0
%ASA-7-713034: Group = 136.1.123.3, IP = 136.1.123.3, Received local IP
Proxy Subnet data in ID Payload:  Address 136.1.121.0, Mask
255.255.255.0, Protocol 0, Port 0
%ASA-7-713906: Group = 136.1.123.3, IP = 136.1.123.3, QM IsRekeyed old
sa not found by addr
```

 **Note**

The firewall starts scanning the crypto map attached to the interface where the IPSec session terminates. The crypto map is scanned to find the matching peer IP address and extract the access-list associated with this peer. Additionally, the locally configured transform-set is extracted and compared to the remote proposal. In our case, everything matches OK, and the local endpoint may prepare and send an answer.

```
%ASA-7-713221: Group = 136.1.123.3, IP = 136.1.123.3, Static Crypto Map
check, checking map = VPN, seq = 10...
%ASA-7-713225: Group = 136.1.123.3, IP = 136.1.123.3, Static Crypto Map
check, map VPN, seq = 10 is a successful match
%ASA-7-713066: Group = 136.1.123.3, IP = 136.1.123.3, IKE Remote Peer
configured for crypto map: VPN
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing IPSec
SA payload
%ASA-7-715027: Group = 136.1.123.3, IP = 136.1.123.3, IPSec SA Proposal
# 1, Transform # 1 acceptable Matches global IPSec SA entry # 10
%ASA-7-713906: Group = 136.1.123.3, IP = 136.1.123.3, IKE: requesting
SPI!
%ASA-7-715006: Group = 136.1.123.3, IP = 136.1.123.3, IKE got SPI from
key engine: SPI = 0x0c27c77b
%ASA-7-713906: Group = 136.1.123.3, IP = 136.1.123.3, oakley
constucting quick mode
<snip>
```

 **Note**

The local endpoint prepares the quick-mode response, with the local proxy IDs and the accepted proposal.

```
%ASA-7-715001: Group = 136.1.123.3, IP = 136.1.123.3, constructing proxy ID
%ASA-7-713906: Group = 136.1.123.3, IP = 136.1.123.3, Transmitting Proxy Id:
  Remote subnet: 136.1.23.0 Mask 255.255.255.0 Protocol 0 Port 0
  Local subnet: 136.1.121.0 mask 255.255.255.0 Protocol 0 Port 0
%ASA-7-715046: Group = 136.1.123.3, IP = 136.1.123.3, constructing qm hash payload
%ASA-7-714005: Group = 136.1.123.3, IP = 136.1.123.3, IKE Responder sending 2nd QM pkt: msg id = c244ac78
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE SENDING Message (msgid=c244ac78) with payloads : HDR + HASH (8) + SA (1) + NONCE (10) + ID (5) + ID (5) + NONE (0) total length : 164
```

 **Note**

The final 3rd message of the QM exchange is received from the remote end. Now Phase 2 negotiations have been successfully terminated and we have IPsec SAs installed in both endpoints.

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE RECEIVED Message (msgid=c244ac78) with payloads : HDR + HASH (8) + NONE (0) total length : 48
%ASA-7-715047: Group = 136.1.123.3, IP = 136.1.123.3, processing hash payload
%ASA-7-713906: Group = 136.1.123.3, IP = 136.1.123.3, loading all IPSEC SAs
%ASA-7-715001: Group = 136.1.123.3, IP = 136.1.123.3, Generating Quick Mode Key!
%ASA-7-715001: Group = 136.1.123.3, IP = 136.1.123.3, Generating Quick Mode Key!
```

 **Note**

Let's review the same IKE message exchange at the other side of the tunnel, in R3. Enable debugging and generate some traffic that matches the VPN filter.

```
Rack1R3#clear crypto isakmp
Rack1R3#clear crypto sa
```

```
Rack1R3#debug crypto isakmp
Crypto ISAKMP debugging is on
```

```
Rack1R3#debug crypto ipsec
Crypto IPSEC debugging is on
```

```
Rack1R3#ping 136.1.121.1 source fastEthernet 0/1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 136.1.121.1, timeout is 2 seconds:
Packet sent with a source address of 136.1.23.3
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/7/8 ms
Rack1R3#
```

 **Note**

The first thing that the local router attempts to do is to find a local SA matching the traffic. Since there is no local SA to use, the local endpoint starts ISAKMP negotiations:

```
IPSEC(sa_request): ,
  (key eng. msg.) OUTBOUND local= 136.1.123.3, remote= 136.1.123.12,
  local_proxy= 136.1.23.0/255.255.255.0/0/0 (type=4),
  remote_proxy= 136.1.121.0/255.255.255.0/0/0 (type=4),
  protocol= ESP, transform= esp-3des esp-md5-hmac (Tunnel),
  lifedur= 3600s and 4608000kb,
  spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x0
ISAKMP:(0): SA request profile is (NULL)
ISAKMP: Created a peer struct for 136.1.123.12, peer port 500
ISAKMP: New peer created peer = 0x83E80BDC peer_handle = 0x80000005
ISAKMP: Locking peer struct 0x83E80BDC, refcount 1 for isakmp_initiator
ISAKMP: local port 500, remote port 500
ISAKMP: set new node 0 to QM_IDLE
ISAKMP: Find a dup sa in the avl tree during calling isadb_insert sa =
83EA4438
```

 **Note**

By default, IKE Main Mode is selected for negotiations. Based on the ISAKMP policy, the locally configured keys are looked up to find the one matching the remote peer's IP address. The first packet will not be sent out until a local pre-shared key is found. The initial proposal contains the list of local ISAKMP policies, and suggests the responder to select the best one.

```
ISAKMP:(0):Can not start Aggressive mode, trying Main mode.
ISAKMP:(0):found peer pre-shared key matching 136.1.123.12
ISAKMP:(0): constructed NAT-T vendor-rfc3947 ID
ISAKMP:(0): constructed NAT-T vendor-07 ID
ISAKMP:(0): constructed NAT-T vendor-03 ID
ISAKMP:(0): constructed NAT-T vendor-02 ID
ISAKMP:(0):Input = IKE_MSG_FROM_IPSEC, IKE_SA_REQ_MM
ISAKMP:(0):Old State = IKE_READY New State = IKE_I_MM1

ISAKMP:(0): beginning Main Mode exchange
ISAKMP:(0): sending packet to 136.1.123.12 my_port 500 peer_port 500
(I) MM_NO_STATE
ISAKMP:(0):Sending an IKE IPv4 Packet.
ISAKMP (0:0): received packet from 136.1.123.12 dport 500 sport 500
Global (I) MM_NO_STATE
ISAKMP:(0):Input = IKE_MSG_FROM_PEER, IKE_MM_EXCH
ISAKMP:(0):Old State = IKE_I_MM1 New State = IKE_I_MM2
```

 **Note**

We just received a response to our initial message with the ISAKMP policy selected by the peer. The response also contains other useful information such as vendor IDs.

```
ISAKMP:(0): processing SA payload. message ID = 0
ISAKMP:(0): processing vendor id payload
ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
ISAKMP:(0): vendor ID is NAT-T v2
ISAKMP:(0): processing vendor id payload
ISAKMP:(0): processing IKE frag vendor id payload
ISAKMP:(0): Support for IKE Fragmentation not enabled
ISAKMP:(0):found peer pre-shared key matching 136.1.123.12
ISAKMP:(0): local preshared key found
ISAKMP : Scanning profiles for xauth ...
```

 **Note**

The local endpoint attempts to match the policy selected by the remote endpoint against the list of the local rules. If a match is found, the system may process further to Key Exchange step.

```
ISAKMP:(0):Checking ISAKMP transform 1 against priority 10 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 2
ISAKMP:      auth pre-share
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of 0x0 0x1 0x51 0x80
ISAKMP:(0):atts are acceptable. Next payload is 0
ISAKMP:(0):Acceptable atts:actual life: 0
ISAKMP:(0):Acceptable atts:life: 0
ISAKMP:(0):Fill atts in sa vpi_length:4
ISAKMP:(0):Fill atts in sa life_in_seconds:86400
ISAKMP:(0):Returning Actual lifetime: 86400
ISAKMP:(0)::Started lifetime timer: 86400.

ISAKMP:(0): processing vendor id payload
ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
ISAKMP:(0): vendor ID is NAT-T v2
ISAKMP:(0): processing vendor id payload
ISAKMP:(0): processing IKE frag vendor id payload
ISAKMP:(0): Support for IKE Fragmentation not enabled
ISAKMP:(0):Input = IKE_MSG_INTERNAL, IKE_PROCESS_MAIN_MODE
ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM2
```

 **Note**

The third message is sent out, with the KE payload and other information.

```
ISAKMP:(0): sending packet to 136.1.123.12 my_port 500 peer_port 500
(I) MM_SA_SETUP
ISAKMP:(0):Sending an IKE IPv4 Packet.
ISAKMP:(0):Input = IKE_MSG_INTERNAL, IKE_PROCESS_COMPLETE
ISAKMP:(0):Old State = IKE_I_MM2  New State = IKE_I_MM3
```

 **Note**

Received the KE response from out peer. At this point, we should be able to generate the session key based on the DH key-exchange and the pre-shared key configured for the peer.

```
ISAKMP (0:0): received packet from 136.1.123.12 dport 500 sport 500
Global (I) MM_SA_SETUP
ISAKMP:(0):Input = IKE_MSG_FROM_PEER, IKE_MM_EXCH
ISAKMP:(0):Old State = IKE_I_MM3 New State = IKE_I_MM4

ISAKMP:(0): processing KE payload. message ID = 0
ISAKMP:(0): processing NONCE payload. message ID = 0
ISAKMP:(0):found peer pre-shared key matching 136.1.123.12
ISAKMP:(1004): processing vendor id payload
ISAKMP:(1004): vendor ID is Unity
ISAKMP:(1004): processing vendor id payload
ISAKMP:(1004): vendor ID seems Unity/DPD but major 144 mismatch
ISAKMP:(1004): vendor ID is XAUTH
ISAKMP:(1004): processing vendor id payload
ISAKMP:(1004): speaking to another IOS box!
ISAKMP:(1004): processing vendor id payload
ISAKMP:(1004):vendor ID seems Unity/DPD but hash mismatch
ISAKMP:received payload type 20
ISAKMP:received payload type 20
ISAKMP:(1004):Input = IKE_MSG_INTERNAL, IKE_PROCESS_MAIN_MODE
ISAKMP:(1004):Old State = IKE_I_MM4 New State = IKE_I_MM4
```

 **Note**

Fifth and Sixth messages constitute IKE ID exchange and peer authentication. However, like mentioned before, the IDs are not actually used for authentication in Main Mode, since the session key generated already assumes mutual authentication. The message below contains the local endpoint ID.

```
ISAKMP:(1004):Send initial contact
ISAKMP:(1004):SA is doing pre-shared key authentication using id type
ID_IPV4_ADDR
ISAKMP (0:1004): ID payload
    next-payload : 8
    type         : 1
    address      : 136.1.123.3
    protocol     : 17
    port        : 500
    length      : 12
ISAKMP:(1004):Total payload length: 12
ISAKMP:(1004): sending packet to 136.1.123.12 my_port 500 peer_port 500
(I) MM_KEY_EXCH
```

```
ISAKMP:(1004):Sending an IKE IPv4 Packet.  
ISAKMP:(1004):Input = IKE_MSG_INTERNAL, IKE_PROCESS_COMPLETE  
ISAKMP:(1004):Old State = IKE_I_MM4 New State = IKE_I_MM5
```

 **Note**

The message received in response contains the IKE ID of the remote box, in this case it's the IP address of the firewall.

```
ISAKMP (0:1004): received packet from 136.1.123.12 dport 500 sport 500  
Global (I) MM_KEY_EXCH  
ISAKMP:(1004): processing ID payload. message ID = 0  
ISAKMP (0:1004): ID payload  
    next-payload : 8  
    type          : 1  
    address       : 136.1.123.12  
    protocol      : 17  
    port          : 0  
    length        : 12  
ISAKMP:(0):: peer matches *none* of the profiles  
ISAKMP:(1004): processing HASH payload. message ID = 0  
ISAKMP:received payload type 17  
ISAKMP:(1004): processing vendor id payload  
ISAKMP:(1004): vendor ID is DPD  
ISAKMP:(1004):SA authentication status:  
    authenticated  
ISAKMP:(1004):SA has been authenticated with 136.1.123.12  
ISAKMP: Trying to insert a peer 136.1.123.3/136.1.123.12/500/, and  
inserted successfully 83E80BDC.  
ISAKMP:(1004):Input = IKE_MSG_FROM_PEER, IKE_MM_EXCH  
ISAKMP:(1004):Old State = IKE_I_MM5 New State = IKE_I_MM6  
  
ISAKMP:(1004):Input = IKE_MSG_INTERNAL, IKE_PROCESS_MAIN_MODE  
ISAKMP:(1004):Old State = IKE_I_MM6 New State = IKE_I_MM6  
  
ISAKMP:(1004):Input = IKE_MSG_INTERNAL, IKE_PROCESS_COMPLETE  
ISAKMP:(1004):Old State = IKE_I_MM6 New State = IKE_P1_COMPLETE
```

 **Note**

The six message IKE MM exchange has completed. Now it is time for 3 messages of Quick Mode. The initial message from the local node contains the local proxy IDs (not shown in the output) and our security policy (cipher, hash, etc).

```
ISAKMP:(1004):beginning Quick Mode exchange, M-ID of 946616940
ISAKMP:(1004):QM Initiator gets spi
ISAKMP:(1004): sending packet to 136.1.123.12 my_port 500 peer_port 500
(I) QM_IDLE
ISAKMP:(1004):Sending an IKE IPv4 Packet.
ISAKMP:(1004):Node 946616940, Input = IKE_MESG_INTERNAL, IKE_INIT_QM
ISAKMP:(1004):Old State = IKE_QM_READY New State = IKE_QM_I_QM1
ISAKMP:(1004):Input = IKE_MESG_INTERNAL, IKE_PHASE1_COMPLETE
ISAKMP:(1004):Old State = IKE_P1_COMPLETE New State = IKE_P1_COMPLETE
```

 **Note**

The next message is the response from the peer. Now the debug output shows the transform set selected by the peer and other SA parameters. Additionally, you can see the Proxy IDs sent by the remote peer.

```
ISAKMP (0:1004): received packet from 136.1.123.12 dport 500 sport 500
Global (I) QM_IDLE
ISAKMP:(1004): processing HASH payload. message ID = 946616940
ISAKMP:(1004): processing SA payload. message ID = 946616940
ISAKMP:(1004):Checking IPsec proposal 1
ISAKMP: transform 1, ESP_3DES
ISAKMP:   attributes in transform:
ISAKMP:     SA life type in seconds
ISAKMP:     SA life duration (basic) of 3600
ISAKMP:     SA life type in kilobytes
ISAKMP:     SA life duration (VPI) of 0x0 0x46 0x50 0x0
ISAKMP:     encaps is 1 (Tunnel)
ISAKMP:     authenticator is HMAC-MD5
ISAKMP:(1004):atts are acceptable.
IPSEC(validate_proposal_request): proposal part #1
IPSEC(validate_proposal_request): proposal part #1,
(key eng. msg.) INBOUND local= 136.1.123.3, remote= 136.1.123.12,
local_proxy= 136.1.23.0/255.255.255.0/0/0 (type=4),
remote_proxy= 136.1.121.0/255.255.255.0/0/0 (type=4),
protocol= ESP, transform= NONE (Tunnel),
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x0
```

 **Note**

The local endpoint matches the QM response message against the crypto map configured on the interface to find a match. Since the match was successful, IPSec SAs are created and user traffic is now encrypted.

```
Crypto mapdb : proxy_match
  src addr      : 136.1.23.0
  dst addr      : 136.1.121.0
  protocol      : 0
  src port      : 0
  dst port      : 0
ISAKMP:(1004): processing NONCE payload. message ID = 946616940
ISAKMP:(1004): processing ID payload. message ID = 946616940
ISAKMP:(1004): processing ID payload. message ID = 946616940
ISAKMP:(1004): Creating IPSec SAs
  inbound SA from 136.1.123.12 to 136.1.123.3 (f/i) 0/ 0
  (proxy 136.1.121.0 to 136.1.23.0)
  has spi 0xD00FF993 and conn_id 0
  lifetime of 3600 seconds
  lifetime of 4608000 kilobytes
  outbound SA from 136.1.123.3 to 136.1.123.12 (f/i) 0/0
  (proxy 136.1.23.0 to 136.1.121.0)
  has spi 0xD7A26A and conn_id 0
  lifetime of 3600 seconds
  lifetime of 4608000 kilobytes
```

 **Note**

The last packet finished the QM negotiations. Now both parties may encrypt and exchange traffic.

```
ISAKMP:(1004): sending packet to 136.1.123.12 my_port 500 peer_port 500
(I) QM_IDLE
ISAKMP:(1004):Sending an IKE IPv4 Packet.
ISAKMP:(1004):deleting node 946616940 error FALSE reason "No Error"
ISAKMP:(1004):Node 946616940, Input = IKE_MESG_FROM_PEER, IKE_QM_EXCH
ISAKMP:(1004):Old State = IKE_QM_I_QM1 New State =
IKE_QM_PHASE2_COMPLETE
```

1.2 IPsec and NAT Interaction in ASA Firewall

- Enable NAT in the ASA firewall and translate all inside addresses using the IP address of the outside interface.
- Ensure the VPN traffic is not affected by this configuration.

Configuration

Note

ASA firewall code applies NAT translation before IPsec encryption. Thus, with NAT enabled you might find that traffic is not matching your proxy-ID access-list because of having source IP addresses translated. As it has been discussed in the ASA Firewall section of the workbook, the solution is to use the NAT exemption rules. Alternatively, you may disable NAT control, if it has been enabled, and configure policy NAT rules so that only the relevant traffic is translated.

The same NAT and IPSec order of operations takes place for Cisco IOS. However, in Cisco IOS you may simply configure the NAT access-lists so that VPN traffic is not translated.

```
ASA1:
nat-control
!
! NAT for inside users
!
nat (inside) 1 0 0
global (outside) 1 interface

!
! Exemption access-list
!
access-list EXEMPT permit ip 136.1.121.0 255.255.255.0 136.1.23.0
255.255.255.0
nat (inside) 0 access-list EXEMPT
```

Verification

Note

Telnet from R1 to R3 – this traffic is not VPN protected. Make sure you see a translation entry for the connection.

```
Rack1R1#telnet 136.1.123.3
Trying 136.1.123.3 ... Open
```

User Access Verification

```
Password:
R3>
Rack1AS>12
[Resuming connection 12 to asal ... ]
```

```
Rack1ASA1(config)# show x
1 in use, 10 most used
PAT Global 136.1.123.12(1024) Local 136.1.121.1(11072)
```

Note

Now initiate some traffic between the protected subnets. Make sure the traffic makes it through and no additional NAT translation entries are being created.

```
Rack1R1#ping 136.1.23.2
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 136.1.23.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 96/165/189
ms
```

```
Rack1R1#telnet 136.1.23.2
Trying 136.1.23.2 ... Open
```

User Access Verification

```
Password:
R2>
```

```
Rack1ASA1(config)# show xlate
0 in use, 10 most used
```

 **Note**

Even though there are not xlates created, we still see the connection across the firewall.

```
Rack1ASA1(config)# show connection
```

```
5 in use, 43 most used
```

```
TCP out 136.1.23.2:23 in 136.1.121.1:11073 idle 0:00:52 bytes 111 flags
```

```
UIO
```

1.3 Authentication using Digital Signatures

- Reconfigure the L2L tunnel between ASA1 and R3 to use digital certificates for ISAKMP authentication.
- Use the following URL for CA enrollment:
<http://10.0.0.100:80/certsrv/mscep/mscep.dll> and use the domain name INE.com for certificate DNs

Configuration

Note

Authentication using pre-shared keys does not scale well for networks with many IPsec tunnels, as it requires setting a separate key for every pair of devices. One may use wildcard-pre-shared keys to resolve this issue. Wildcard PSKs are configured using the netmask based address scope, e.g. `crypto isakmp key CISCO address 150.1.0.0 255.255.0` will use PSK value of CISCO for the whole subnet 150.1.0.0/16. However, this solution is less secure, as it reuses the same key for multiple endpoints.

PKI or Public Key Infrastructure offers a scalable way to authenticate all communicating endpoints in secure manner. Every router that needs to participate in IPsec VPN is issued a digital certificate by the Certification Authority (CA). A typical digital certificate binds the identity information of a router (e.g. hostname or IP address) to the router's public key by means of CA digital signature. This involves the use of public key cryptography algorithms, such as RSA. Based on this binding, any device that trusts the CA certificate, i.e. trusts the signature of the CA, would accept the identity inside the signed certificate. Next, in order to authenticate the side that presents the digital certificate, the challenging party may ask the responder to encrypt a random piece of information using the responder's private key. Then, the challenge information is decrypted using the public key from the certificate. If the original challenge and the decrypted challenge are the same, then the identity of the remote party has been established and validated.

The above procedure allows all routers that trust the same CA to authenticate each other. More than that, since certificates contain information about endpoints identity, no communicating party may ever deny that it has participated in information exchange, even if they try.

The certificate-based authentication is also called digital-signature or RSA-signature based, as RSA is the most common underlying public-key encryption algorithm.

To configure the authentication based on digital signatures, perform the following steps:

1) Generate RSA private and public keys in every router or firewall that should participate in VPN. This is needed to create a key that uniquely identifies the router. Notice that IOS routers and ASA firewall require you to configure a domain-name before generating the RSA keys. (Notice that by generating the keys you automatically activate SSH protocol support in IOS router). The command to create the keys is **crypto key generate rsa**.

2) Configure the CA trustpoint in routers and firewalls. The devices use the CA trustpoint to enroll with the CA for certificates and validate certificates presented by peers. You define a trustpoint using the command **crypto ca trustpoint** in both IOS routers and ASA firewalls. Cisco devices use HTTP-based SCEP protocol to interact with CA over the network and thus you must define the URL to access the CA. The command to define an URL is **enrollment url** under the CA trustpoint configuration mode. By using SCEP, the devices may request CA digital certificate (which is often self-signed) and store it locally. Many CAs define CRL (Certificate Revocation List) URL to allow the endpoints to poll the CA for the list of invalidated (revoked) certificates. If your CA does not support this functionality or you don't need it, specify the **cr1 optional** command when configuring a trustpoint.

3) After you retrieved the CA certificate, you must authenticate it – i.e. tell the system that this certificate is trusted and could be used for validating other certificates. Use the command **crypto ca authenticate** for certificate retrieval and authentication. Before you authenticate the CA trustpoint, make sure the time is coordinated between the CA and the router. Most often, you may want to use NTP protocol to accomplish this.

4) After you have authenticated the CA certificate you may enroll with the CA using the command **crypto ca enroll**. When you issue this command, the router or the firewall will send a request to the CA using SCEP protocol. The request contains the public key of the router and convenient identity information, such as router's hostname and domain name. Depending on the CA configuration, the certificate request will be kept pending until the administrator approves it, or the certificate could be issued automatically. In our topology, the CA is configured for automatic certificate issuing. You may want to use the command **debug crypto pki transactions** if you are running into any issues enrolling with the CA.

5) After you have obtained the certificate, you may configure ISAKMP policy for authentication based on RSA signatures. For ASA firewall, you should also configure the respective tunnel-group to use the particular trustpoint for certificate validation:

```
tunnel-group <IPAddr> ipsec-attributes
  trust-point IE1
```

6) All other IPsec settings remain the same. Only the ISAKMP (Phase 1) is affected by certificate configuration.

Finally, remember to permit SCEP (transported in HTTP) and NTP protocol across the firewall, if the scenario needs this. In our case, we configure the access-list in the ASA to permit R3 communicating with the CA server.

ASA1:

```
!
! Trustpoint configuration
!
crypto ca trustpoint IE1
  enrollment url http://10.0.0.100:80/certsrv/mscep/mscep.dll
  crl optional
!
ntp server 10.0.0.100
!
crypto ca authenticate IE1
domain-name INE.com
crypto key generate rsa general-keys modulus 512
crypto ca enroll IE1

!
! L2L VPN. ISAKMP Configuration
!
crypto isakmp policy 10
  authentication rsa-sig
  encryption 3des
  hash md5
!
crypto isakmp enable outside

!
! Configure tunnel group for L2L tunnel
!
tunnel-group 136.1.123.3 type ipsec-l2l
tunnel-group 136.1.123.3 ipsec-attributes
  trust-point IE1

!
! Configure transform-set
!
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac
!
! Access-list to classify traffic for encryption
!
access-list VLAN121_TO_VLAN23 permit ip 136.1.121.0 255.255.255.0
136.1.23.0 255.255.255.0
!
```

```

! Configure crypto-map
!
crypto map VPN 10 match address VLAN121_TO_VLAN23
crypto map VPN 10 set peer 136.1.123.3
crypto map VPN 10 set transform-set 3DES_MD5
!
! Apply crypto-map and enable VPN traffic to bypass ACLs
!
crypto map VPN interface outside
sysopt connection permit-vpn

!
! Allow NTP and SCEP from R3
!
access-list OUTSIDE_IN permit tcp any host 10.0.0.100 eq 80
access-list OUTSIDE_IN permit udp any host 10.0.0.100 eq 123

R3:
!
! Configure trustpoint
!
crypto ca trustpoint IE1
  enrollment url http://10.0.0.100:80/certsrv/mscep/mscep.dll
  crl optional
!
! Generate keypair and enroll
!
ip domain name INE.com
crypto key generate rsa general-keys modulus 512
ntp server 10.0.0.100
crypto ca authenticate IE1
crypto ca enroll IE1

!
! L2L VPN:
! Configure ISAKMP policy
!
crypto isakmp policy 10
  encryption 3des
  auth rsa-sig
  hash md5
  group 2

!
! Create transform-set
!
crypto ipsec transform-set 3DES_MD5 esp-3des esp-md5-hmac
!
! Create access-list to classify traffic for encryption
!
ip access-list extended VLAN23_TO_VLAN121
  permit ip 136.1.23.0 0.0.0.255 136.1.121.0 0.0.0.255

!
! Create & apply crypto map
!
crypto map VPN 10 ipsec-isakmp

```

```
match address VLAN23_TO_VLAN121
set transform 3DES_MD5
set peer 136.1.123.12
!
interface FastEthernet 0/0
crypto map VPN
```

Verification

Note

Here are sample steps to enroll the ASA with the CA. Start by authenticating the CA. You should receive the certificate fingerprint (a hash of the certificate contents) which you are supposed to validate with the CA administrator via some out-of-band channel (e.g. phone).

```
Rack1ASA1(config)# crypto ca authenticate IE1
```

```
INFO: Certificate has the following attributes:  
Fingerprint:      74f95e93 4f8c8af3 5fd15364 8efbb479  
Do you accept this certificate? [yes/no]: yes  
Trustpoint CA certificate accepted.
```

Note

Now generate the RSA keys for the appliance (the hostname should have been already configured) and start the enrollment procedure.

```
Rack1ASA1(config)# crypto key generate rsa general-keys modulus 512  
WARNING: You have a RSA keypair already defined named <Default-RSA-Key>.
```

```
Do you really want to replace them? [yes/no]: yes  
Keypair generation process begin. Please wait...
```

```
Rack1ASA1(config)# crypto ca enroll IE1
```

```
%  
% Start certificate enrollment ..  
% Create a challenge password. You will need to verbally provide this  
password to the CA Administrator in order to revoke your certificate.  
For security reasons your password will not be saved in the configuration.  
Please make a note of it.  
Password: cisco  
Re-enter password: cisco
```

```
% The fully-qualified domain name in the certificate will be: ASA1.INE.com
```

```
% Include the device serial number in the subject name? [yes/no]: no
```

```
Request certificate from CA? [yes/no]: yes  
% Certificate request sent to Certificate Authority
```

Note

The following text should appear notifying that the certificate has been granted.

```
Rack1ASA1(config)# The certificate has been granted by CA!
```

 **Note**

Check the certificates received from the CA. There should be two – one granted to the router and another for the CA itself.

```
Rack1ASA1(config)# show crypto ca certificates
```

```
Certificate
  Status: Available
  Certificate Serial Number: 2358cad2000100000026
  Certificate Usage: General Purpose
  Public Key Type: RSA (512 bits)
  Issuer Name:
    cn=IESERVER1
    o=Internetwork Expert
    l=Reno
    st=NV
    c=US
    ea=bmcgahan@INE.com
  Subject Name:
    hostname=ASA1.INE.com
```

 **Note**

The following in the list of CRL distribution points encoded in the certificate. If you didn't set CRL as optional under the trustpoint, then the router will attempt to use those URL to retrieve the CRL and check the peer certificate against them during ISAKMP negotiations.

```
CRL Distribution Points:
```

```
[1] http://ieserver1/CertEnroll/IESERVER1(1).crl
[2] file://\IESERVER1\CertEnroll\IESERVER1(1).crl
```

```
Validity Date:
```

```
  start date: 10:07:47 UTC Jan 12 2007
  end   date: 10:17:47 UTC Jan 12 2008
```

```
Associated Trustpoints: IE1
```

```
CA Certificate
```

```
Status: Available
Certificate Serial Number: 6a8b964c37f91bb245b01de2a6363745
Certificate Usage: Signature
Public Key Type: RSA (1024 bits)
Issuer Name:
  cn=IESERVER1
  o=Internetwork Expert
  l=Reno
  st=NV
  c=US
  ea=bmcgahan@INE.com
Subject Name:
  cn=IESERVER1
  o=Internetwork Expert
```

```
l=Reno
st=NV
c=US
ea=bmcgahan@INE.com
CRL Distribution Points:
 [1] http://ieserver1/CertEnroll/IESERVER1(1).crl
 [2] file://\IESERVER1\CertEnroll\IESERVER1(1).crl
Validity Date:
 start date: 09:01:58 UTC Jul 21 2006
 end date: 09:09:34 UTC Jul 21 2008
Associated Trustpoints: IE1
```

Note

Now repeat the same procedure with R3.

Rack1R3(config)#crypto ca authenticate IE1

Certificate has the following attributes:

Fingerprint: 74F95E93 4F8C8AF3 5FD15364 8EFBB479

% Do you accept this certificate? [yes/no]: yes

Trustpoint CA certificate accepted.

Rack1R3(config)#crypto ca enroll IE1

%

% Start certificate enrollment ..

% Create a challenge password. You will need to verbally provide this password to the CA Administrator in order to revoke your certificate.

For security reasons your password will not be saved in the configuration.

Please make a note of it.

Password: cisco

Re-enter password: cisco

% The fully-qualified domain name in the certificate will be:
Rack1R3.INE.com

% The subject name in the certificate will be: Rack1R3.INE.com

% Include the router serial number in the subject name? [yes/no]: no

% Include an IP address in the subject name? [no]: no

Request certificate from CA? [yes/no]: yes

% Certificate request sent to Certificate Authority

% The certificate request fingerprint will be displayed.

% The 'show crypto ca certificate' command will also show the fingerprint.

R3(config)#

Fingerprint: 4913AF72 E8D4DEC9 01526382 C936CF4D

Jan 12 11:07:54.762: %CRYPTO-6-CERTRET: Certificate received from Certificate Authority

Rack1R3#show crypto ca certificates

Certificate

Status: Available
Certificate Serial Number: 23869E24000100000027
Certificate Usage: General Purpose
Issuer:
 CN = IESERVER1
 O = Internetwork Expert
 L = Reno
 ST = NV
 C = US
 EA = bmcgahan@INE.com
Subject:
 Name: Rack1R3.INE.com
 OID.1.2.840.113549.1.9.2 = Rack1R3.INE.com
CRL Distribution Point:
 http://ieserver1/CertEnroll/IESERVER1(1).crl
Validity Date:
 start date: 10:57:52 UTC Jan 12 2007
 end date: 11:07:52 UTC Jan 12 2008
 renew date: 00:00:00 UTC Jan 1 1970
Associated Trustpoints: IE1

CA Certificate

Status: Available
Certificate Serial Number: 6A8B964C37F91BB245B01DE2A6363745
Certificate Usage: Signature
Issuer:
 CN = IESERVER1
 O = Internetwork Expert
 L = Reno
 ST = NV
 C = US
 EA = bmcgahan@INE.com
Subject:
 CN = IESERVER1
 O = Internetwork Expert
 L = Reno
 ST = NV
 C = US
 EA = bmcgahan@INE.com
CRL Distribution Point:
 http://ieserver1/CertEnroll/IESERVER1(1).crl
Validity Date:
 start date: 09:01:58 UTC Jul 21 2006
 end date: 09:09:34 UTC Jul 21 2008
Associated Trustpoints: IE1

Note

Now generate some traffic and verify that the IPsec tunnel establishes.

Rack1R2#ping 136.1.121.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 136.1.121.1, timeout is 2 seconds:
...!!
Success rate is 40 percent (2/5), round-trip min/avg/max = 12/30/48 ms

Note

Notice that authentication method for ISAKMP is RSA-Sig.

Rack1R3#show crypto isakmp sa detail

Codes: C - IKE configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal
X - IKE Extended Authentication
psk - Preshared key, rsig - RSA signature
renc - RSA encryption

C-id	Local	Remote	I-VRF	Encr	Hash	Auth	DH
	Lifetime	Cap.					
1	136.1.123.3	136.1.123.12		3des	md5	rsig	2
	23:58:57						

Note

Check the IPsec SAs and ensure packets are encrypted and decrypted.

Rack1R3#show crypto ipsec sa

```
interface: Ethernet0/0
  Crypto map tag: VPN, local addr. 136.1.123.3

  protected vrf:
    local ident (addr/mask/prot/port): (136.1.23.0/255.255.255.0/0/0)
    remote ident (addr/mask/prot/port): (136.1.121.0/255.255.255.0/0/0)
    current_peer: 136.1.123.12:500
      PERMIT, flags={origin_is_acl,}
      #pkts encaps: 2, #pkts encrypt: 2, #pkts digest 2
      #pkts decaps: 2, #pkts decrypt: 2, #pkts verify 2
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts compr. failed: 0
      #pkts not decompressed: 0, #pkts decompress failed: 0
      #send errors 2, #recv errors 0

    local crypto endpt.: 136.1.123.3, remote crypto endpt.:
136.1.123.12
      path mtu 1500, media mtu 1500
      current outbound spi: 4A807DEA
      <snip>
```

 **Note**

Let's run some ISAKMP debugging and observe how the devices proceed with Phase 1 when using digital certificates:

```
Rack1R3#clear crypto isakmp
Rack1R3#clear crypto sa
```

```
Rack1R3#debug crypto isakmp
Crypto ISAKMP debugging is on
```

```
Rack1R3#ping 136.1.121.1 source fastEthernet 0/1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 136.1.121.1, timeout is 2 seconds:
Packet sent with a source address of 136.1.23.3
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 4/7/8 ms
```

 **Note**

The initial steps are the same. The initiator prepares an ISAKMP proposal. Note that even though there is a PSK found matching the remote peer, it is not used, as the negotiate police will use digital certificates.

```
ISAKMP:(0): SA request profile is (NULL)
ISAKMP: Created a peer struct for 136.1.123.12, peer port 500
ISAKMP: New peer created peer = 0x83D1E870 peer_handle = 0x80000006
ISAKMP: Locking peer struct 0x83D1E870, refcount 1 for isakmp_initiator
ISAKMP: local port 500, remote port 500
ISAKMP: set new node 0 to QM_IDLE
insert sa successfully sa = 82EFD620
ISAKMP:(0):Can not start Aggressive mode, trying Main mode.
ISAKMP:(0):found peer pre-shared key matching 136.1.123.12
ISAKMP:(0): constructed NAT-T vendor-rfc3947 ID
ISAKMP:(0): constructed NAT-T vendor-07 ID
ISAKMP:(0): constructed NAT-T vendor-03 ID
ISAKMP:(0): constructed NAT-T vendor-02 ID
ISAKMP:(0):Input = IKE_MSG_FROM_IPSEC, IKE_SA_REQ_MM
ISAKMP:(0):Old State = IKE_READY New State = IKE_I_MM1
```

 **Note**

Initial exchange results in the negotiated Phase 1 policy that uses RSA signatures for authentication:

```
ISAKMP:(0): beginning Main Mode exchange
ISAKMP:(0): sending packet to 136.1.123.12 my_port 500 peer_port 500
(I) MM_NO_STATE
ISAKMP:(0):Sending an IKE IPv4 Packet.
ISAKMP (0:0): received packet from 136.1.123.12 dport 500 sport 500
Global (I) MM_NO_STATE
ISAKMP:(0):Input = IKE_MSG_FROM_PEER, IKE_MM_EXCH
ISAKMP:(0):Old State = IKE_I_MM1 New State = IKE_I_MM2

ISAKMP:(0): processing SA payload. message ID = 0
ISAKMP:(0): processing vendor id payload
ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
ISAKMP:(0): vendor ID is NAT-T v2
ISAKMP:(0): processing vendor id payload
ISAKMP:(0): processing IKE frag vendor id payload
ISAKMP:(0): Support for IKE Fragmentation not enabled
ISAKMP : Scanning profiles for xauth ...
ISAKMP:(0):Checking ISAKMP transform 1 against priority 10 policy
ISAKMP:      encryption 3DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 2
ISAKMP:      auth RSA sig
ISAKMP:      life type in seconds
ISAKMP:      life duration (VPI) of 0x0 0x1 0x51 0x80
ISAKMP:(0):atts are acceptable. Next payload is 0
ISAKMP:(0):Acceptable atts:actual life: 0
ISAKMP:(0):Acceptable atts:life: 0
ISAKMP:(0):Fill atts in sa vpi_length:4
ISAKMP:(0):Fill atts in sa life_in_seconds:86400
ISAKMP:(0):Returning Actual lifetime: 86400
ISAKMP:(0)::Started lifetime timer: 86400.

ISAKMP:(0): processing vendor id payload
ISAKMP:(0): vendor ID seems Unity/DPD but major 123 mismatch
ISAKMP:(0): vendor ID is NAT-T v2
ISAKMP:(0): processing vendor id payload
ISAKMP:(0): processing IKE frag vendor id payload
ISAKMP:(0): Support for IKE Fragmentation not enabled
ISAKMP:(0):Input = IKE_MSG_INTERNAL, IKE_PROCESS_MAIN_MODE
ISAKMP:(0):Old State = IKE_I_MM2 New State = IKE_I_MM2
```

Note

Now the local endpoint constructs an IKE message with KE payload, just as usual. However, in addition to that, it includes special CERT_REQ payload in the same packet. This payload specifies the DN of the CA that the responder must choose to answer. The remote endpoint must find a configured trustpoint matching this DN. Additionally, the CERT_REQ payload contains the type (format) of the certificate that the peer should present.

```
ISAKMP (0:0): constructing CERT_REQ for issuer
cn=IESERVER1,o=Internetwork Expert\,
Inc.,l=Reno,st=Nevada,c=US,e=bmcgahan@INE.com
ISAKMP:(0): sending packet to 136.1.123.12 my_port 500 peer_port 500
(I) MM_SA_SETUP
ISAKMP:(0):Sending an IKE IPv4 Packet.
ISAKMP:(0):Input = IKE_MSG_INTERNAL, IKE_PROCESS_COMPLETE
ISAKMP:(0):Old State = IKE_I_MM2 New State = IKE_I_MM3
```

Note

The responder sends us a new message completing the DH key exchange. Additionally, the KE response contains CERT_REQ payload from the peer, instructing us to find the matching certificate as well.

```
ISAKMP (0:0): received packet from 136.1.123.12 dport 500 sport 500
Global (I) MM_SA_SETUP
ISAKMP:(0):Input = IKE_MSG_FROM_PEER, IKE_MM_EXCH
ISAKMP:(0):Old State = IKE_I_MM3 New State = IKE_I_MM4

ISAKMP:(0): processing KE payload. message ID = 0
ISAKMP:(0): processing NONCE payload. message ID = 0
ISAKMP:(1005): processing CERT_REQ payload. message ID = 0
ISAKMP:(1005): peer wants a CT_X509_SIGNATURE cert
ISAKMP:(1005): peer wants cert issued by cn=IESERVER1,o=Internetwork
Expert\, Inc.,l=Reno,st=Nevada,c=US,e=bmcgahan@INE.com
  Choosing trustpoint IE1 as issuer
ISAKMP:(1005): processing vendor id payload
ISAKMP:(1005): vendor ID is Unity
ISAKMP:(1005): processing vendor id payload
ISAKMP:(1005): vendor ID seems Unity/DPD but major 175 mismatch
ISAKMP:(1005): vendor ID is XAUTH
ISAKMP:(1005): processing vendor id payload
ISAKMP:(1005): speaking to another IOS box!
ISAKMP:(1005): processing vendor id payload
ISAKMP:(1005):vendor ID seems Unity/DPD but hash mismatch
ISAKMP:received payload type 20
ISAKMP:received payload type 20
ISAKMP:(1005):Input = IKE_MSG_INTERNAL, IKE_PROCESS_MAIN_MODE
ISAKMP:(1005):Old State = IKE_I_MM4 New State = IKE_I_MM4
```

 **Note**

If a matching CA is found, the initiator may now generate new IKE message to present and authenticate itself. This message contains the identity of the local endpoint (ID payload), the certificate (CERT payload) and some signed material to prove our identity. Notice that the IKE identity must match the subject name in the certificate. Thus, even though the local system is configured to use the IP address as its IKE ID, it changes to the hostname, as it is presented in the certificate.

```
ISAKMP:(1005):Send initial contact
ISAKMP:(1005):My ID configured as IPv4 Addr, but Addr not in Cert!
ISAKMP:(1005):Using FQDN as My ID
ISAKMP:(1005):SA is doing RSA signature authentication using id type
ID_FQDN
ISAKMP (0:1005): ID payload
  next-payload : 6
  type         : 2
  FQDN name    : Rack1R3.INE.com
  protocol     : 17
  port         : 500
  length       : 23
ISAKMP:(1005):Total payload length: 23
ISAKMP (0:1005): constructing CERT payload for hostname=Rack1R3.INE.com
ISAKMP: growing send buffer from 1024 to 3072
ISAKMP:(1005): using the IE1 trustpoint's keypair to sign
ISAKMP:(1005): sending packet to 136.1.123.12 my_port 500 peer_port 500
(I) MM_KEY_EXCH
ISAKMP:(1005):Sending an IKE IPv4 Packet.
ISAKMP:(1005):Input = IKE_MESG_INTERNAL, IKE_PROCESS_COMPLETE
ISAKMP:(1005):Old State = IKE_I_MM4  New State = IKE_I_MM5
```

 **Note**

If the peer was able to validate our identity, it responds with its own ID, CERT and signature. The local end processes this information. First, the certificate presented must be verifiable using the locally configured CA. Next, the IKE ID presented, must match the subject's name in the certificate. And finally, the signature presented should match the peer's certificate.

```

ISAKMP (0:1005): received packet from 136.1.123.12 dport 500 sport 500
Global (I) MM_KEY_EXCH
ISAKMP:(1005): processing ID payload. message ID = 0
ISAKMP (0:1005): ID payload
    next-payload : 6
    type          : 2
    FQDN name     : Rack1ASA1.INE.com
    protocol      : 0
    port          : 0
    length        : 25
ISAKMP:(0):: peer matches *none* of the profiles
ISAKMP:(1005): processing CERT payload. message ID = 0
ISAKMP:(1005): processing a CT_X509_SIGNATURE cert
ISAKMP:(1005): peer's pubkey isn't cached
ISAKMP:(1005): Unable to get DN from certificate!
ISAKMP:(1005): Cert presented by peer contains no OU field.
ISAKMP:(0):: peer matches *none* of the profiles
ISAKMP:(1005): processing SIG payload. message ID = 0
ISAKMP:received payload type 17
ISAKMP:(1005): processing vendor id payload
ISAKMP:(1005): vendor ID is DPD
ISAKMP:(1005):SA authentication status:
    authenticated
ISAKMP:(1005):SA has been authenticated with 136.1.123.12
ISAKMP: Trying to insert a peer 136.1.123.3/136.1.123.12/500/, and
inserted successfully 83D1E870.

```

Note

After this, IKE Phase 1 has completed and Phase 2 may start. Now look at the same debugging output in the ASA firewall:

Rack1ASA1#debug crypto isakmp

```

%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE RECEIVED Message (msgid=0)
with payloads : HDR + SA (1) + VENDOR (13) + VENDOR (13) + VENDOR (13)
+ VENDOR (13) + NONE (0) total length : 200
%ASA-7-715047: IP = 136.1.123.3, processing SA payload
%ASA-7-713906: IP = 136.1.123.3, Oakley proposal is acceptable
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received NAT-Traversal RFC VID
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received NAT-Traversal ver 03 VID
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received NAT-Traversal ver 02 VID
%ASA-7-715047: IP = 136.1.123.3, processing IKE SA payload
%ASA-7-715028: IP = 136.1.123.3, IKE SA Proposal # 1, Transform # 1
acceptable Matches global IKE entry # 3
%ASA-7-715046: IP = 136.1.123.3, constructing ISAKMP SA payload
%ASA-7-715046: IP = 136.1.123.3, constructing NAT-Traversal VID ver 02
payload

```

```
%ASA-7-715046: IP = 136.1.123.3, constructing Fragmentation VID +
extended capabilities payload
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE SENDING Message (msgid=0)
with payloads : HDR + SA (1) + VENDOR (13) + VENDOR (13) + NONE (0)
total length : 128
```

Note

The first two messages negotiate the ISAKMP SA parameters. Now the next two messages perform DH KE and include CERT_REQ payloads. This instructs both peers to prepare the digital certificates. First we receive the KE message from the initiator.

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE RECEIVED Message (msgid=0)
with payloads : HDR + KE (4) + NONCE (10) + CERT_REQ (7) + VENDOR (13)
+ VENDOR (13) + VENDOR (13) + VENDOR (13) + NAT-D (130) + NAT-D (130) +
NONE (0) total length : 455
%ASA-7-715047: IP = 136.1.123.3, processing ke payload
%ASA-7-715047: IP = 136.1.123.3, processing ISA_KE payload
%ASA-7-715047: IP = 136.1.123.3, processing nonce payload
%ASA-7-715047: IP = 136.1.123.3, processing cert request payload
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received Cisco Unity client VID
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received DPD VID
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715038: IP = 136.1.123.3, Processing IOS/PIX Vendor ID payload
(version: 1.0.0, capabilities: 00000f7f)
%ASA-7-715047: IP = 136.1.123.3, processing VID payload
%ASA-7-715049: IP = 136.1.123.3, Received xauth V6 VID
%ASA-7-715047: IP = 136.1.123.3, processing NAT-Discovery payload
%ASA-7-713906: IP = 136.1.123.3, computing NAT Discovery hash
%ASA-7-715047: IP = 136.1.123.3, processing NAT-Discovery payload
%ASA-7-713906: IP = 136.1.123.3, computing NAT Discovery hash
%ASA-7-715046: IP = 136.1.123.3, constructing ke payload
%ASA-7-715046: IP = 136.1.123.3, constructing nonce payload
%ASA-7-715046: IP = 136.1.123.3, constructing certreq payload
%ASA-7-715046: IP = 136.1.123.3, constructing Cisco Unity VID payload
%ASA-7-715046: IP = 136.1.123.3, constructing xauth V6 VID payload
%ASA-7-715048: IP = 136.1.123.3, Send IOS VID
%ASA-7-715038: IP = 136.1.123.3, Constructing ASA spoofing IOS Vendor
ID payload (version: 1.0.0, capabilities: 20000001)
%ASA-7-715046: IP = 136.1.123.3, constructing VID payload
%ASA-7-715048: IP = 136.1.123.3, Send Altiga/Cisco VPN3000/Cisco ASA GW
VID
%ASA-7-715046: IP = 136.1.123.3, constructing NAT-Discovery payload
%ASA-7-713906: IP = 136.1.123.3, computing NAT Discovery hash
%ASA-7-715046: IP = 136.1.123.3, constructing NAT-Discovery payload
%ASA-7-713906: IP = 136.1.123.3, computing NAT Discovery hash
%ASA-7-713906: IP = 136.1.123.3, Generating keys for Responder...
```

 **Note**

This endpoint responds and requests a certificate from the initiator as well:

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE SENDING Message (msgid=0)
with payloads : HDR + KE (4) + NONCE (10) + CERT_REQ (7) + VENDOR (13)
+ VENDOR (13) + VENDOR (13) + VENDOR (13) + NAT-D (130) + NAT-D (130) +
NONE (0) total length : 455
```

 **Note**

Last two messages authenticate both peers. Notice the IKE ID payload, the CERT payload and the SIG payload (carries the digital signature):

```
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE RECEIVED Message (msgid=0)
with payloads : HDR + ID (5) + CERT (6) + SIG (9) + NOTIFY (11) + NONE
(0) total length : 1161
```

```
%ASA-7-715047: IP = 136.1.123.3, processing ID payload
```

```
%ASA-7-713906: IP = 136.1.123.3, ID_FQDN ID received, len 15
0000: 5261636B 3152332E 494E452E 636F6D          Rack1R3.INE.com
```

```
%ASA-7-715047: IP = 136.1.123.3, processing cert payload
```

```
%ASA-7-715001: IP = 136.1.123.3, processing RSA signature
```

```
%ASA-7-715076: IP = 136.1.123.3, Computing hash for ISAKMP
```

```
%ASA-7-713906: Dump of received Signature, len 64:
```

```
0000: 78D66814 DB7BCC43 70D0E33F BCACD32C      x.h..{.Cp..?...,
0010: CF2B9215 38A5164A DEA99128 821EE93A      .+..8..J...(...:
0020: E873D8C6 95B4983C 11A929B3 B51A071A      .s.....<..).....
0030: C72C496C 84F1A505 638D488C 029E8815      .,Il....c.H.....
```

```
%ASA-7-715047: IP = 136.1.123.3, processing notify payload
```

```
%ASA-6-713172: IP = 136.1.123.3, Automatic NAT Detection Status:
```

```
Remote end is NOT behind a NAT device      This end is NOT behind a
NAT device
```

 **Note**

Before generating a response, the ASA tries to match the remote endpoint to a local tunnel-group. It first uses the OU field from the subject's name (OU is not present though). It then tries to use the IKE ID (the hostname) and does not find a matching group. Finally, it falls back to the initiator's IP address and finds a matching tunnel-group.

```
%ASA-7-713906: IP = 136.1.123.3, Trying to find group via OU...
%ASA-3-713020: IP = 136.1.123.3, No Group found by matching OU(s) from
ID payload:      Unknown
%ASA-7-713906: IP = 136.1.123.3, Trying to find group via IKE ID...
%ASA-7-713906: IP = 136.1.123.3, Trying to find group via IP ADDR...
%ASA-7-713906: IP = 136.1.123.3, Connection landed on tunnel_group
136.1.123.3
```

Note

Now it attempts to validate the certificate presented by the peer. It finds the local trustpoint and validates the trust chain. Notice that CRL check is not performed, as CRL is configured to be optional for the trustpoint.

```
%ASA-7-717025: Validating certificate chain containing 1
certificate(s).
%ASA-7-717029: Identified client certificate within certificate chain.
serial number: 3DDB0F2700000000000E, subject name:
hostname=Rack1R3.INE.com.
%ASA-7-717030: Found a suitable trustpoint IE1 to validate certificate.
%ASA-7-717024: Checking CRL from trustpoint: IE1 for an optional CRL
check. CRL will be cached for the next CRL check.
%ASA-6-717022: Certificate was successfully validated. serial number:
3DDB0F2700000000000E, subject name: hostname=Rack1R3.INE.com.
%ASA-6-717028: Certificate chain was successfully validated with
warning, revocation status was not checked.
%ASA-7-713906: Group = 136.1.123.3, IP = 136.1.123.3, peer ID type 2
received (FQDN)
```

Note

Now the local endpoint constructs an authentication reply with its own IKE ID, CERT and SIG payloads.

```
%ASA-7-715046: Group = 136.1.123.3, IP = 136.1.123.3, constructing ID
payload
%ASA-7-715046: Group = 136.1.123.3, IP = 136.1.123.3, constructing cert
payload
%ASA-7-715001: Group = 136.1.123.3, IP = 136.1.123.3, constructing RSA
signature
%ASA-7-715076: Group = 136.1.123.3, IP = 136.1.123.3, Computing hash
for ISAKMP
%ASA-4-717026: Name lookup failed for hostname sc09-aaa during PKI
operation.
```

```
%ASA-3-717010: CRL polling failed for trustpoint IE1.  
%ASA-7-713906: Constructed Signature Len: 64  
%ASA-7-713906: Constructed Signature:  
0000: 64982959 4272E3FD 17A9D36B E911BC9C      d.)YBr.....k....  
0010: 7D96F327 17A13173 59A83A20 4264A63C      }..'...1sY.: Bd.<  
0020: C2335FA8 EF901972 66984FB8 8EAB72C7      .3_....rf.O...r.  
0030: 0A5FFE10 5F1A1BC7 831945DC D51EBCDD      ._._.E.....
```

```
%ASA-7-715034: IP = 136.1.123.3, Constructing IOS keep alive payload:  
proposal=32767/32767 sec.  
%ASA-7-715046: Group = 136.1.123.3, IP = 136.1.123.3, constructing dpd  
vid payload  
%ASA-7-713236: IP = 136.1.123.3, IKE_DECODE SENDING Message (msgid=0)  
with payloads : HDR + ID (5) + CERT (6) + SIG (9) + IOS KEEPALIVE (128)  
+ VENDOR (13) + NONE (0) total length : 1168  
%ASA-6-113009: AAA retrieved default group policy (DfltGrpPolicy) for  
user = 136.1.123.3
```

 **Note**

After this, IPsec Phase 1 has completed and both parties may proceed to Phase 2.

```
%ASA-5-713119: Group = 136.1.123.3, IP = 136.1.123.3, PHASE 1 COMPLETED  
%ASA-7-713121: IP = 136.1.123.3, Keep-alive type for this connection:  
DPD  
%ASA-7-715080: Group = 136.1.123.3, IP = 136.1.123.3, Starting P1 rekey  
timer: 82080 seconds.
```