

Configuring a LAN-to-LAN IPSec Between a Router and a PIX Using Digital Certificates

Document ID: 18014

Introduction

Before You Begin

- Conventions

- Prerequisites

- Components Used

- Background Theory

- Network Diagram

Configuring the Router and the PIX Firewall

- Configurations

Obtain Certificates

- Obtain Certificates on the Router

- Obtain Certificates on the PIX

Verify

- Sample Output from Router show Commands

- Sample Output from PIX show Commands

Troubleshoot

- Troubleshooting Commands

- Sample Certificate Debug from the Router

- Sample Certificate Debug from the PIX

- Sample IPSec Debug from the Router

- Sample IPSec Debugs from the PIX

- Potential Problems

- Delete Certificates and RSA Key Pairs

Related Information

Introduction

This document illustrates how to configure a Cisco router and a Cisco Secure PIX Firewall to implement a LAN-to-LAN IPSec using digital certificates. To achieve this configuration, you need to perform the following tasks:

1. Configure the router and the PIX.
2. Obtain digital certificates on the router and the PIX.
3. Configure the IKE and IPSec policies on the router and the PIX, and define what traffic (interesting traffic) will be encrypted using IPSec through an access list.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

There are no specific prerequisites for this document.

Components Used

The information in this document is based on the software and hardware versions below.

- Cisco 1700 Router
- Cisco IOS® Software version 12.2(6)
- Cisco PIX Firewall 520
- PIX Firewall version 6.0.1.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

Background Theory

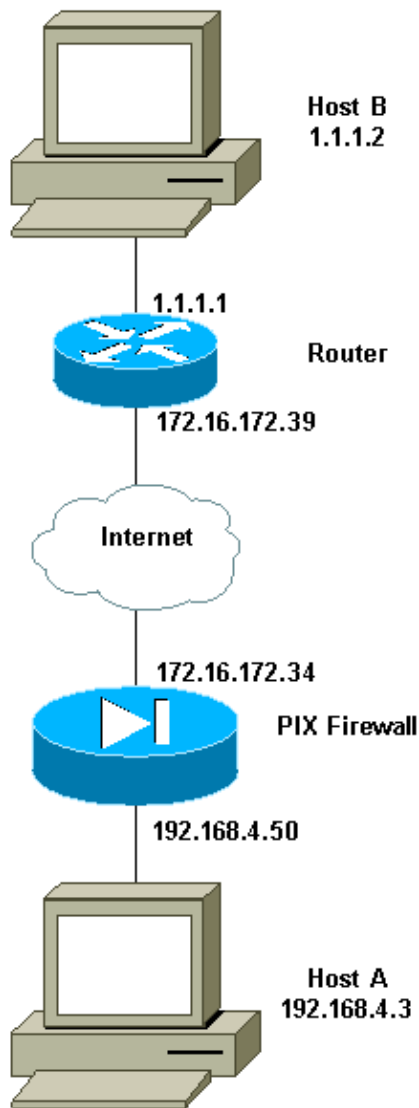
In our example, we have defined the network address of host A (source address) and the network address of host B (destination address) as the traffic that the IPsec will encrypt on the PIX. The access-list on the router is the mirror image of the access-list on the PIX.

We have configured the PIX and router so that the hosts residing on the inside LAN of the two devices use their private addresses while going through the IPsec tunnel. On the PIX, the **access-list** and **nat 0** commands work together. When host A on the 192.168.4.0 network goes to the 1.1.1.0 network, the access list permits the 192.168.4.0 network traffic to be encrypted without Network Address Translation (NAT). However, when those same users go anywhere else, they are translated to the 172.16.172.57 address through Port Address Translation (PAT). On the router, the **route-map** and **access-list** commands permit the 1.1.1.0 network traffic to be encrypted without NAT. However, when the same Host B goes anywhere else, they are translated to the 172.16.172.39 address through PAT.

To test the configuration, we pinged from host A behind the PIX Firewall to host B behind the router. When the IP packet arrived at the PIX Firewall, it matched the access-list and thus initiated the IPsec negotiation. Thus the PIX is the initiator and the router is the responder during the IPsec negotiation. For troubleshooting purposes, you need to examine both the PIX and router crypto debugs.

Network Diagram

This document uses the network setup shown in the diagram below.



Configuring the Router and the PIX Firewall

Configurations

In this section, you are presented with the information to configure the features described in this document.

- Router Sample Configuration
- PIX Sample Configuration

Router Sample Configuration

```
1720-1#show running-config
Building configuration...

Current configuration : 8694 bytes
!
! Last configuration change at 20:17:48 PST Thu Jan 10 2002
! NVRAM config last updated at 20:19:27 PST Thu Jan 10 2002
!
version 12.2
no parser cache
service timestamps debug uptime
service timestamps log uptime
```

```
no service password-encryption
!
hostname 1720-1
!
no logging buffered
enable secret 5 $1$6jAs$tNxIla/2DYFAtPLyCDXjo/
enable password ww
!
username cisco password 0 cisco
username all
memory-size iomem 15
clock timezone PST -8
ip subnet-zero
no ip domain-lookup
ip domain-name cisco.com
!
ip ssh time-out 120
ip ssh authentication-retries 3
!
!
!
crypto ca identity vpn
  enrollment retry count 20
  enrollment mode ra
  enrollment url http://171.69.89.16:80
  query url ldap://171.69.89.16
crypto ca certificate chain vpn
certificate 3B2FD652
308202C4 3082022D A0030201 0202043B 2FD65230 0D06092A 864886F7 0D010105
 0500302D 310B3009 06035504 06130275 73310E30 0C060355 040A1305 63697363
 6F310E30 0C060355 040B1305 736A7670 6E301E17 0D303230 31313130 33303631
 345A170D 30333031 31313033 33363134 5A304E31 0B300906 03550406 13027573
 310E300C 06035504 0A130563 6973636F 310E300C 06035504 0B130573 6A76706E
 311F301D 06092A86 4886F70D 01090216 10313732 302D312E 63697363 6F2E636F
 6D305C30 0D06092A 864886F7 0D010101 0500034B 00304802 4100A085 B4A756F8
 CEB91F2E 52E2A23F 847EC95F 44F65AF2 EBC1F816 081CC61F AB077482 F1FAD124
 2444B9F6 6B9EC48E 1B1EB5B9 D0E802BA B9A57048 EBB8CD18 773F0203 010001A3
 82011230 82010E30 0B060355 1D0F0404 030205A0 301B0603 551D1104 14301282
 10313732 302D312E 63697363 6F2E636F 6D302B06 03551D10 04243022 800F3230
 30323031 31313033 30363134 5A810F32 30303230 39323331 35333631 345A304F
 0603551D 1F044830 463044A0 42A040A4 3E303C31 0B300906 03550406 13027573
 310E300C 06035504 0A130563 6973636F 310E300C 06035504 0B130573 6A76706E
 310D300B 06035504 03130443 524C3130 1F060355 1D230418 30168014 46C1609C
 DBEA53EE 80A48060 1A96583B 0DF80D2F 301D0603 551D0E04 160414B1 2707AB30
 F7CFDC79 C554D1AE 3208EF16 CF96ED30 09060355 1D130402 30003019 06092A86
 4886F67D 07410004 0C300A1B 0456352E 30030204 B0300D06 092A8648 86F70D01
 01050500 03818100 E82DE82B AE5C7F80 EB9CED1A 306F36E6 437DA791 81D53CF3
 0E561C8A 7A168EDE 6728F371 3EB90B21 CC40E1F3 CA4ED98F CDF6A6E1 A2C0AA38
 4AE137C7 281AA7EC AD26D550 4E4AAA0B E0C588F8 661C4031 ACF35F7B 28330B64
 667E00E3 832AED7F 08D5EA3D 33CCB2BE E73DC41A B40A9B64 4CD2D98C 6943AE84
 55605741 E136A6BD
quit
certificate ra-sign 3B2FD319
308202FF 30820268 A0030201 0202043B 2FD31930 0D06092A 864886F7 0D010105
 0500302D 310B3009 06035504 06130275 73310E30 0C060355 040A1305 63697363
 6F310E30 0C060355 040B1305 736A7670 6E301E17 0D303130 36313932 32303333
 315A170D 30343036 31393232 33333331 5A304531 0B300906 03550406 13027573
 310E300C 06035504 0A130563 6973636F 310E300C 06035504 0B130573 6A76706E
 31163014 06035504 03130D46 69727374 204F6666 69636572 30819F30 0D06092A
 864886F7 0D010101 05000381 8D003081 89028181 00E85434 395790E9 416ED13D
 72F1A411 333A0984 66B8F68A 0ECA7E2B CBC40C39 A21E2D8A 5F94772D 69846720
 73227891 E43D46B6 B2D1DDC5 385C5135 DB2075F1 4D252ACF AC80DA4C 2111946F
 26F7193B 8EA1CA66 8332D2A1 5310B2D7 07C985A8 0B44CE37 BC95EAFB C328D4C6
 73B3B35E 0F6D25F5 DCAC6AFA 2DAAD6D1 47BB3396 E1020301 0001A382 01123082
 010E300B 0603551D 0F040403 02078030 2B060355 1D100424 3022800F 32303031
 30363139 32323033 33315A81 0F323030 33303732 37303233 3333315A 301B0603
```

551D0904 14301230 1006092A 864886F6 7D07441D 31030201 00304F06 03551D1F
04483046 3044A042 A040A43E 303C310B 30090603 55040613 02757331 0E300C06
0355040A 13056369 73636F31 0E300C06 0355040B 1305736A 76706E31 0D300B06
03550403 13044352 4C31301F 0603551D 23041830 16801446 C1609CDB EA53EE80
A480601A 96583B0D F80D2F30 1D060355 1D0E0416 04147BD2 620C611F 3AC69FB3
155FD8F9 8A7CF353 3A583009 0603551D 13040230 00301906 092A8648 86F67D07
4100040C 300A1B04 56352E30 030204B0 300D0609 2A864886 F70D0101 05050003
8181003A A6431D7D 1979DDF9 CC99D8F8 CC987F67 DBF67280 2A9418E9 C6255B08
DECDE1C2 50FCB1A6 544F1D51 C214162E E2403DAB 2F1294C4 841240ED FD6F799C
130A0B24 AC74DD74 C60EB5CD EC648631 E0B88B3F 3D19A2E1 6492958E 9F64746E
45C080AE E5A6C245 7827D7B1 380A6FE8 A01D9022 7F52AD9C B596743A 853549C5 771DA2
quit

certificate ra-encrypt 3B2FD318

308202D0 30820239 A0030201 0202043B 2FD31830 0D06092A 864886F7 0D010105
0500302D 310B3009 06035504 06130275 73310E30 0C060355 040A1305 63697363
6F310E30 0C060355 040B1305 736A7670 6E301E17 0D303130 36313932 32303333

315A170D 30343036 31393232 33333331 5A304531 0B300906 03550406 13027573
310E300C 06035504 0A130563 6973636F 310E300C 06035504 0B130573 6A76706E
31163014 06035504 03130D46 69727374 204F6666 69636572 30819F30 0D06092A
864886F7 0D010101 05000381 8D003081 89028181 00BFC427 727E15E9 30CB1BCB
C0EFFB2F 3E4916D4 EC365F57 C13D1356 6388E66D 7BCCBCB9 04DA2E7C C9639F31
AF15E7B1 E698A33C 0EB447E4 B3B72EC8 766EADCF 9883E612 AD782E39 B0603A90
0322CE78 D6735E07 BDC022F1 1164EC9E 31FC5309 9AA9DC1D 69ECC316 8727A6CB
ADCFB488 FF904D6D 9D9E5778 05B24D4B BB5B4F5F 4D020301 0001A381 E43081E1
300B0603 551D0F04 04030205 20301B06 03551D09 04143012 30100609 2A864886
F67D0744 1D310302 0100304F 0603551D 1F044830 463044A0 42A040A4 3E303C31
0B300906 03550406 13027573 310E300C 06035504 0A130563 6973636F 310E300C
06035504 0B130573 6A76706E 310D300B 06035504 03130443 524C3130 1F060355
1D230418 30168014 46C1609C DBEA53EE 80A48060 1A96583B 0DF80D2F 301D0603
551D0E04 16041400 A7C3DD9F 9FAB0A25 E1485FC7 DB88A63F 78CE4830 09060355
1D130402 30003019 06092A86 4886F67D 07410004 0C300A1B 0456352E 30030204
B0300D06 092A8648 86F70D01 01050500 03818100 69105382 0BE0BA59 B0CD2652
9C6A4585 940C7882 DCEB1D1E 610B8525 0C032A76 2C8758C2 F5CA1EF4 B946848A
C49047D5 6D1EF218 FA082A00 16CCD9FC 42DF3B05 A8EF2AAD 151637DE 67885BB2
BA0BB6A1 308F63FF 21C3CB00 9272257A 3C292645 FD62D486 C247F067 301C2FEE
5CF6D12B 6CFA1DAA E74E8B8E 5B017A2E 5BB6C5F9
quit

certificate ca 3B2FD307

308202E4 3082024D A0030201 0202043B 2FD30730 0D06092A 864886F7 0D010105
0500302D 310B3009 06035504 06130275 73310E30 0C060355 040A1305 63697363
6F310E30 0C060355 040B1305 736A7670 6E301E17 0D303130 36313932 32303234
305A170D 32313036 31393232 33323430 5A302D31 0B300906 03550406 13027573
310E300C 06035504 0A130563 6973636F 310E300C 06035504 0B130573 6A76706E
30819F30 0D06092A 864886F7 0D010101 05000381 8D003081 89028181 00E8C25B
EDF4A6EE A352B142 C16578F4 FBDAF45E 4F2F7733 8D2B8879 96138C63 1DB713BF
753BF845 2D7E600F AAF4D75B 9E959513 BB13FF13 36696F48 86C464F2 CF854A66
4F8E83F8 025F216B A44D4BB2 39ADD1A5 1BCCF812 09A19BDC 468EEAE1 B6C2A378
69C81348 1A9CD61C 551216F2 8B168FBB 94CBEF37 E1D9A8F7 80BBC17F D1020301
0001A382 010F3082 010B3011 06096086 480186F8 42010104 04030200 07304F06
03551D1F 04483046 3044A042 A040A43E 303C310B 30090603 55040613 02757331
0E300C06 0355040A 13056369 73636F31 0E300C06 0355040B 1305736A 76706E31
0D300B06 03550403 13044352 4C31302B 0603551D 10042430 22800F32 30303130
36313932 32303234 305A810F 32303231 30363139 32323332 34305A30 0B060355
1D0F0404 03020106 301F0603 551D2304 18301680 1446C160 9CDBEA53 EE80A480
601A9658 3B0DF80D 2F301D06 03551D0E 04160414 46C1609C DBEA53EE 80A48060
1A96583B 0DF80D2F 300C0603 551D1304 05300301 01FF301D 06092A86 4886F67D
07410004 10300E1B 0856352E 303A342E 30030204 90300D06 092A8648 86F70D01
01050500 03818100 7E3DBAC4 8CAE7D5A B19C0625 8780D222 F965A1A2 C2C25B84
CBC5A203 BF50FAC4 9656699A 5D28CB46 40776237 87163118 8F3C0F47 D0CAA36B
6AB34F99 AB71269E 78C0AC10 DA0B9EC5 AE448B46 701254CF 3EBC64C1 5DBB2EE5
56C0140B B0C83497 D79FB148 80018F51 3A4B6174 590B85AA 9CE3B391 629406AA
7CE9CC0D 01593E6B
quit

!

crypto isakmp policy 10

```

hash md5
crypto isakmp identity hostname
!
!
crypto ipsec transform-set myset esp-des esp-md5-hmac
!
!

crypto map vpn 10 ipsec-isakmp
set peer 172.16.172.34
set transform-set myset
match address 130
!
!
!
!
!
interface Loopback0
ip address 10.10.10.1 255.255.255.0
!
interface Loopback1
ip address 121.1.1.1 255.255.255.0
!
interface Loopback88
ip address 88.88.88.88 255.255.255.255
!
interface FastEthernet0
ip address 172.16.172.39 255.255.255.240
ip nat outside
speed auto
crypto map vpn
!
interface Serial0
ip nat inside
ip address 1.1.1.1 255.255.255.252
!
ip nat inside source route-map nonat interface FastEthernet0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 172.16.172.33
no ip http server
ip pim bidir-enable
!
access-list 120 deny ip 1.1.1.0 0.0.0.255 192.168.4.0 0.0.0.255
access-list 120 permit ip 1.1.1.0 0.0.0.255 any
access-list 130 permit ip 1.1.1.0 0.0.0.255 192.168.4.0 0.0.0.255
route-map nonat permit 10
match ip address 120
!
line con 0
line aux 0
line vty 0 4
exec-timeout 0 0
password cisco
no login
line vty 5 15
login
!
no scheduler allocate
end

```

PIX Sample Configuration

```

pix520-1# write terminal
Building configuration...
: Saved

```

```
:
PIX Version 6.0(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 2KFQnbNIdI.2KYOU encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pix520-1
domain-name vpn.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
access-list 130 permit ip 192.168.4.0 255.255.255.0 1.1.1.0 255.255.255.0
access-list 140 permit ip 192.168.4.0 255.255.255.0 1.1.1.0 255.255.255.0
no pager
logging on
logging monitor debugging
logging buffered debugging
logging trap debugging
logging history debugging
logging host outside 192.168.2.6
interface ethernet0 auto
interface ethernet1 auto
mtu outside 1500
mtu inside 1500
ip address outside 172.16.172.34 255.255.255.240
ip address inside 192.168.4.50 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
pdm history enable
arp timeout 14400
global (outside) 1 172.16.172.57 netmask 255.255.255.255
nat (inside) 0 access-list 140
nat (inside) 1 0.0.0.0 0.0.0.0 0 0
route outside 0.0.0.0 0.0.0.0 172.16.172.33 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00
h323 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server mytest protocol tacacs+
aaa-server nasir protocol radius
snmp-server host outside 192.168.2.6
no snmp-server location
no snmp-server contact
snmp-server community public
snmp-server enable traps
floodguard enable
sysopt connection permit-ipsec
no sysopt route dnat
crypto ipsec transform-set myset esp-des esp-md5-hmac
crypto map mymap 5 ipsec-isakmp
crypto map mymap 5 match address 130
crypto map mymap 5 set peer 172.16.172.39
crypto map mymap 5 set transform-set myset
```

```

crypto map mymap interface outside
isakmp enable outside
isakmp policy 10 authentication rsa-sig
isakmp policy 10 encryption des
isakmp policy 10 hash md5
isakmp policy 10 group 1
isakmp policy 10 lifetime 86400
ca identity cisco 171.69.89.16:/cgi-bin 171.69.89.16
ca configure cisco ra 20 5
telnet 192.168.4.0 255.255.255.0 inside
telnet 171.69.89.82 255.255.255.255 inside
telnet 192.168.4.3 255.255.255.255 inside
telnet timeout 5
ssh 172.0.0.0 255.0.0.0 outside
ssh 171.0.0.0 255.255.255.0 outside
ssh 171.0.0.0 255.0.0.0 outside
ssh 171.0.0.0 255.0.0.0 inside
ssh timeout 60
terminal width 80
Cryptochecksum:c2d5976fc87875678356cf83b135bb8c
: end
[OK]
pix520-1#

```

Obtain Certificates

Obtain Certificates on the Router

This section describes how to obtain digital certificates on the router.

1. Configure router's host name and IP domain name if this has not already been done.

```

1720-1# hostname 1720-1
1720-1# ip domain-name cisco.com

```

Note: The host name and domain name are required because the router assigns a fully qualified domain name (FQDN) to the keys and certificates used by IPSec, based on the host name and IP domain name you assign to the router. For example, a certificate is named "router.cisco.com" based on a router host name of "router" and a router IP domain name of "cisco.com."

2. Generate the RSA key pair for the router, which is used to sign and encrypt IKE key management messages. You need to generate the key pair in order to obtain a certificate for the router.

```

1720-1(config)#crypto key generate rsa
The name for the keys will be: 1720-1.cisco.com
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes

How many bits in the modulus [512]:
Generating RSA keys ...
[OK]

1720-1(config)#

```

Use the **show crypto key mypubkey rsa** command to see the router's RSA key pair.

```

1720-1#sh cr key mypubkey rsa
% Key pair was generated at: 19:26:22 PST Jan 10 2002
Key name: 1720-1.cisco.com
Usage: General Purpose Key
Key Data:

```

```
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00A085B4 756F8CE
B91F2E52 E2A23F84 7EC95F44 F65AF2EB C1F81608 1CC61FAB 077482F1 FAD12424
44B9F66B 9EC48E1B 1EB5B9D0 E802BAB9 A57048EB B8CD1877 3F020301 0001
```

% Key pair was generated at: 19:26:24 PST Jan 10 2002

Key name: 1720-1.cisco.com.server

Usage: Encryption Key

Key Data:

```
307C300D 06092A86 4886F70D 01010105 00036B00 30680261 00C653F7 2AE7E397
0041E273 BFCC0E35 E7AF9874 A73B77E8 B15EF54A CA2417AD AB75BAD9 BA1540F4
3DB849BD B70DF4D8 EBBBE7ED AB93BE4B 5C1E9E6A 560A9C8A 12D7CBE3 060DBE7E
8C1667AE 93993049 DA362602 4E4D9EF8 2F8C4777 30F9F958 7F020301 0001
```

1720-1#

3. Declare the certification authority (CA) server in order to configure the communication parameters between the router and the CA. If we are using a registration authority, then we also specify the registration authority (RA) mode. Use the **cr1 optional** command if you want the other peers' certificates to be accepted by the router even if the appropriate certificate revocation list (CRL) is not accessible to the router.

```
1720-1(config)# crypto ca identity vpn
1720-1(ca-identity)#enrollment url http://171.69.89.16:80
1720-1(ca-identity)# query url ldap://171.69.89.16
1720-1(ca-identity)# enrollment retry count 20
1720-1(ca-identity)# enrollment retry period 5
1720-1(ca-identity)# enrollment mode ra
1720-1(ca-identity)#exit
```

4. The router needs to authenticate the CA by obtaining the CA's self-signed certificate which contains the CA's public key. Because the CA signs its own certificate, the CA's public key should be manually authenticated by contacting the CA administrator to compare the CA certificate's fingerprint.

In this example, we manually authenticate the public key by comparing the two fingerprints after we receive the CA's certificate, rather than entering it with a command statement.

```
1720-1(config)#cr ca authenticate vpn
Certificate has the following attributes:
Fingerprint: 1FCDF2C8 2DEDA6AC 4819D4C4 B4CFF2F5
% Do you accept this certificate? [yes/no]: y
1720-1(config)#
```

Use the **sh crypto ca cert** command to view the CA and RA certificates and verify that the authentication was successful.

```
1720-1#sh cr ca cert
RA Signature Certificate
Status: Available
```

!--- The authentication was successful.

```
Certificate Serial Number: 3B2FD319
Key Usage: Signature
Issuer:
  OU = sjvpn
  O = cisco
  C = us
Subject:
  CN = First Officer
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
```

```
start date: 14:03:31 PST Jun 19 2001
end date: 14:33:31 PST Jun 19 2004
Associated Identity: vpn
```

RA KeyEncipher Certificate
Status: Available

!--- The authentication was successful.

```
Certificate Serial Number: 3B2FD318
Key Usage: Encryption
Issuer:
  OU = sjvpn
  O = cisco
  C = us
Subject:
  CN = First Officer
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 14:03:31 PST Jun 19 2001
  end date: 14:33:31 PST Jun 19 2004
Associated Identity: vpn
```

CA Certificate
Status: Available

!--- The authentication was successful.

```
Certificate Serial Number: 3B2FD307
Key Usage: General Purpose
Issuer:
  OU = sjvpn
  O = cisco
  C = us
Subject:
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 14:02:40 PST Jun 19 2001
  end date: 14:32:40 PST Jun 19 2021
Associated Identity: vpn
```

5. Obtain a signed certificate from the CA for each of the router's RSA key pairs. If you generated general purpose RSA keys, the router has one RSA key pair and needs only one certificate. If you generated special usage RSA keys, then the router has two RSA key pairs and needs two certificates.

You have to contact the CA administrator to manually grant the router certificates if it is configured on the CA server. Also, if the CA server is configured so that you have to provide the password at the time of enrollment, then contact CA administrator for this password.

In this example, the CA server was set up so that we did not need to provide a password during enrollment.

```
1720-1(config)#cr ca enroll vpn
%
% 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:

Re-enter password:

```
% The subject name in the certificate will be: 1720-1.cisco.com
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [yes/no]: n
Request certificate from CA? [yes/no]: y
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.
```

```
1720-1(config)# Fingerprint: A1D6C28B 6575AD08 F0B656D4 7161F76F
```

```
3d09h: CRYPTO_PKI: status = 102: certificate request pending
```

After we execute the commands for enrollment, the router communicates with the CA server and attempts to get its certificate. During this time, if the CA server is configured to require a manual authentication of the certificates, then you will need to contact the CA administrator.

Use the **sh crypto ca cert** command to view the router certificate and verify that the enrollment has been successful.

In the following example, the certificates have not been approved.

```
1720-1#sh crypto ca cert
RA Signature Certificate
Status: Available
Certificate Serial Number: 3B2FD319
Key Usage: Signature
Issuer:
  OU = sjvpn
  O = cisco
  C = us
Subject:
  CN = First Officer
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 14:03:31 PST Jun 19 2001
  end date: 14:33:31 PST Jun 19 2004
Associated Identity: vpn
```

```
RA KeyEncipher Certificate
Status: Available
Certificate Serial Number: 3B2FD318
Key Usage: Encryption
Issuer:
  OU = sjvpn
  O = cisco
  C = us
Subject:
  CN = First Officer
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
```

```
start date: 14:03:31 PST Jun 19 2001
end date: 14:33:31 PST Jun 19 2004
Associated Identity: vpn
```

CA Certificate

```
Status: Available
Certificate Serial Number: 3B2FD307
Key Usage: General Purpose
Issuer:
  OU = sjvpn
  O = cisco
  C = us
Subject:
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 14:02:40 PST Jun 19 2001
  end date: 14:32:40 PST Jun 19 2021
Associated Identity: vpn
```

Certificate

```
Subject Name Contains:
  Name: 1720-1.cisco.com
Status: Pending
```

!--- The certificate is still pending.

```
Key Usage: General Purpose
Fingerprint: A1D6C28B 6575AD08 F0B656D4 7161F76F
Associated Identity: vpn
```

The following example output shows that the certificate has been received from the CA.

```
3d09h: %CRYPTO-6-CERTRET: Certificate received from Certificate Authority
```

```
1720-1#sh crypto ca cert
Certificate
  Status: Available
```

!--- This status indicates that the certificates were successfully received.

```
Certificate Serial Number: 3B2FD652
Key Usage: General Purpose
Issuer:
  OU = sjvpn
  O = cisco
  C = us
Subject Name Contains:
  Name: 1720-1.cisco.com
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 19:06:14 PST Jan 10 2002
  end date: 19:36:14 PST Jan 10 2003
Associated Identity: vpn
```

RA Signature Certificate

```
Status: Available
Certificate Serial Number: 3B2FD319
Key Usage: Signature
Issuer:
  OU = sjvpn
```

```
O = cisco
C = us
Subject:
  CN = First Officer
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 14:03:31 PST Jun 19 2001
  end date: 14:33:31 PST Jun 19 2004
Associated Identity: vpn
```

```
RA KeyEncipher Certificate
Status: Available
Certificate Serial Number: 3B2FD318
Key Usage: Encryption
Issuer:
  OU = sjvpn
  O = cisco
  C = us
Subject:
  CN = First Officer
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 14:03:31 PST Jun 19 2001
  end date: 14:33:31 PST Jun 19 2004
Associated Identity: vpn
```

```
CA Certificate
Status: Available
Certificate Serial Number: 3B2FD307
Key Usage: General Purpose
Issuer:
  OU = sjvpn
O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 14:02:40 PST Jun 19 2001
  end date: 14:32:40 PST Jun 19 2021
Associated Identity: vpn
```

6. You can manually request the CA for the CRL. To update the CRL on the router, use the following command:

```
1720-1(config)#crypto ca crl request vpn
1720-1(config)#exit
```

Use the **show crypto ca crls** command to view the CRL.

```
1720-1#sh crypto ca crls
CRL Issuer Name:
  OU = sjvpn, O = cisco, C = us
LastUpdate: 16:17:34 PST Jan 10 2002
NextUpdate: 17:17:34 PST Jan 11 2002
Retrieved from CRL Distribution Point:
  LDAP: CN = CRL1, OU = sjvpn, O = cisco, C = us

1720-1#
```

7. Issue a **write mem** command to save the configuration.

```
1720-1# wr m
Building configuration?
[OK]
1720-1#
```

Obtain Certificates on the PIX

To get the certificates on a PIX firewall, you will follow the same steps as on the router. However, the PIX command syntax is different.

1. Set host name and IP domain name.

```
hostname pix520-1
domain-name vpn.com
```

2. Generate the RSA key pair.

```
pix520-1(config)# ca generate rsa key 512
```

Use the **show ca mypubkey rsa** command to display the RSA key pair.

```
pix520-1(config)# sh ca mypubkey rsa

% Key pair was generated at: 04:54:34 Jan 11 2002

Key name: pix520-1.vpn.com
Usage: General Purpose Key
Key Data:
 305c300d 06092a86 4886f70d 01010105 00034b00 30480241 009d95d5 e1147546
 1f9ef873 81a36256 4b81388b 188fbc66 40fc4c56 c1801311 ff450cca e8d715c3
 ffb8fa28 d347120f ae8a9972 3a88321c a71c1c7f ef29b810 2f020301 0001
pix520-1(config)#
```

3. Declare the CA server.

```
pix520-1(config)# ca identity cisco 171.69.89.16 171.69.89.16
pix520-1(config)# ca configure cisco ra 20 5
```

4. Authenticate the CA.

```
pix520-1(config)# ca authenticate cisco

Certificate has the following attributes:

Fingerprint: 1fcd2c8 2deda6ac 4819d4c4 b4cff2f5
pix520-1(config)#
```

Use the **show ca cert** command to view the CA certificate on the PIX.

```
pix520-1(config)# sh ca cert
CA Certificate
  Status: Available

!--- The authentication was successful.

Certificate Serial Number: 3b2fd307
Key Usage: General Purpose
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
```

```
Validity Date:
  start date: 22:02:40 Jun 19 2001

  end   date: 22:32:40 Jun 19 2021
```

```
RA Signature Certificate
  Status: Available
```

```
!--- The authentication was successful.
```

```
Certificate Serial Number: 3b2fd319
Key Usage: Signature
  CN = First Officer
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 22:03:31 Jun 19 2001

  end   date: 22:33:31 Jun 19 2004
```

```
RA KeyEncipher Certificate
  Status: Available
```

```
!--- The authentication was successful.
```

```
Certificate Serial Number: 3b2fd318
Key Usage: Encryption
  CN = First Officer
  OU = sjvpn
  O = cisco
  C = us
CRL Distribution Point:
  CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
  start date: 22:03:31 Jun 19 2001

  end   date: 22:33:31 Jun 19 2004
```

5. Request the CA for the CRL.

```
pix520-1(config)# ca enroll cisco 171.69.89.16
%
% Start certificate enrollment ..

% The subject name in the certificate will be: pix520-1.vpn.com

% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.

pix520-1(config)#      Fingerprint:  6961df68 d3b5e667 8903a66b 969eee64

CRYPTO_PKI: status = 102: certificate request pending
CRYPTO_PKI: status = 102: certificate request pending
```

The certificate has been granted by CA!

```
pix520-1(config)#
pix520-1(config)# show ca cert
Certificate
  Status: Available
```

!--- The enrollment was successful.

Certificate Serial Number: 3b2fd653
Key Usage: General Purpose
Subject Name
 Name: pix520-1.vpn.com
CRL Distribution Point:
 CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
 start date: 04:13:45 Jan 11 2002
 end date: 04:43:45 Jan 11 2003

RA Signature Certificate
Status: Available

!--- The enrollment was successful.

Certificate Serial Number: 3b2fd319
Key Usage: Signature
 CN = First Officer
 OU = sjvpn
 O = cisco
 C = us
CRL Distribution Point:
 CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
 start date: 22:03:31 Jun 19 2001

 end date: 22:33:31 Jun 19 2004

CA Certificate
Status: Available

!--- The enrollment was successful.

Certificate Serial Number: 3b2fd307
Key Usage: General Purpose
 OU = sjvpn
 O = cisco
 C = us
CRL Distribution Point:
 CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
 start date: 22:02:40 Jun 19 2001

 end date: 22:32:40 Jun 19 2021

RA KeyEncipher Certificate
Status: Available

!--- The enrollment was successful.

Certificate Serial Number: 3b2fd318
Key Usage: Encryption
 CN = First Officer
 OU = sjvpn
 O = cisco
 C = us
CRL Distribution Point:
 CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
 start date: 22:03:31 Jun 19 2001

```
end    date: 22:33:31 Jun 19 2004
```

```
pix520-1(config)#
```

```
pix520-1(config)# ca crl request cisco
```

6. Use the **sh ca crl** command to view the CRL.

```
pix520-1(config)# sh ca crl
```

```
CRL:
```

```
  CRL Issuer Name:
```

```
    OU = sjvpn, O = cisco, C = us
```

```
  LastUpdate: 00:17:34 Jan 11 2002
```

```
  NextUpdate: 01:17:34 Jan 12 2002
```

```
pix520-1(config)#
```

7. To save the certificates on the PIX, use the following command:

```
pix520-1(config)# ca save all
```

```
pix520-1(config)#
```

Verify

This section provides information you can use to confirm your configuration is working properly.

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only), which allows you to view an analysis of **show** command output.

Show commands can be run on the PIX and the router.

- **show crypto isakmp sa** – View all current IKE security associations (SAs) at a peer.
- **show crypto ipsec sa** – Shows the settings used by current IPsec security associations.
- **show crypto engine connections active** – (Router only) Shows current connections and information regarding encrypted and decrypted packets.
- **show crypto ca crls** – (Router only) Shows the current CRL on router.
- **show crypto ca certificates** – (Router only) Shows the router, CA server, and RA certificates on the router. It also shows the certificate distribution point (CDP).
- **show ca certificates** – (PIX only) Shows the PIX, CA, and RA certificates. Unlike the router, it does not show the CDP.
- **show ca crl** – (PIX only) Shows the CRL on the PIX.
- **show clock** – Shows the current time on the router/PIX (from enable mode).

Sample Output from Router show Commands

```
1720-1#sh cr isa sa
```

dst	src	state	conn-id	slot
172.16.172.39	172.16.172.34	QM_IDLE	110	0

```
1720-1#sh cr map
```

```
  Interfaces using crypto map mymap:
```

```
  Crypto Map "vpn" 10 ipsec-isakmp
```

```
    Peer = 172.16.172.34
```

```
    Extended IP access list 130
```

```
      access-list 130 permit ip 1.1.1.0 0.0.0.255 192.168.4.0 0.0.0.255
```

```
Current peer: 172.16.172.34
Security association lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform sets={ myset, }
Interfaces using crypto map vpn:
    FastEthernet0
Interfaces using crypto map certificate:
```

```
1720-1#sh cr isa policy
```

```
Protection suite of priority 10
```

```
    encryption algorithm:  DES - Data Encryption Standard
                          (56 bit keys).
```

```
    hash algorithm:       Message Digest 5
```

```
    authentication method: Rivest-Shamir-Adleman Signature
```

```
    Diffie-Hellman group: #1 (768 bit)
```

```
    lifetime:            86400 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
```

```
1720-1#
```

```
1720-1#sh cr ipsec sa
```

```
interface: FastEthernet0
```

```
    Crypto map tag: vpn, local addr. 172.16.172.39
```

```
local ident (addr/mask/prot/port):
```

```
(1.1.1.0/255.255.255.0/0/0)
```

```
remote ident (addr/mask/prot/port):
```

```
(192.168.4.0/255.255.255.0/0/0)
```

```
current_peer: 172.16.172.34
```

```
    PERMIT, flags={origin_is_acl,}
```

```
    #pkts encaps: 3, #pkts encrypt: 3, #pkts digest 3
```

```
    #pkts decaps: 3, #pkts decrypt: 3, #pkts verify 3
```

```
    #pkts compressed: 0, #pkts decompressed: 0
```

```
    #pkts not compressed: 0, #pkts compr. failed: 0,
```

```
        #pkts decompress failed: 0
```

```
    #send errors 0, #recv errors 0
```

```
local crypto endpt.: 172.16.172.39,
```

```
    remote crypto endpt.: 172.16.172.34
```

```
path mtu 1500, media mtu 1500
```

```
current outbound spi: 3803A0C1
```

```
inbound esp sas:
```

```
    spi: 0xD740971C(3611334428)
```

```
    transform: esp-des esp-md5-hmac ,
```

```
    in use settings = {Tunnel, }
```

```
    slot: 0, conn id: 200, flow_id: 1,
```

```
        crypto map: vpn
```

```
    sa timing: remaining key lifetime
```

```
        (k/sec): (4607999/3150)
```

```
    IV size: 8 bytes
```

```
    replay detection support: Y
```

```
inbound ah sas:
```

```
inbound pcp sas:
```

```
outbound esp sas:
```

```
    spi: 0x3803A0C1(939761857)
```

```
    transform: esp-des esp-md5-hmac ,
```

```
    in use settings = {Tunnel, }
```

```
slot: 0, conn id: 201, flow_id: 2,
crypto map: vpn
sa timing: remaining key lifetime
(k/sec): (4607999/3141)
IV size: 8 bytes
replay detection support: Y
```

outbound ah sas:

outbound pcp sas:

1720-1#

1720-1# **sh cr en conn ac**

ID	Interface	IP-Address	State	Algorithm	Encrypt	Decrypt
110	FastEthernet0	172.16.172.39	set	HMAC_MD5+DES_56_CB	0	0
114	FastEthernet0	172.16.172.39	alloc	NONE	0	0
115	FastEthernet0	172.16.172.39	alloc	NONE	0	0
116	FastEthernet0	172.16.172.39	alloc	NONE	0	0
117	FastEthernet0	172.16.172.39	alloc	NONE	0	0
200	FastEthernet0	172.16.172.39	set	HMAC_MD5+DES_56_CB	0	3
201	FastEthernet0	172.16.172.39	set	HMAC_MD5+DES_56_CB	3	0

1720-1#**sh clock**

01:06:41.786 PST Fri Jan 11 2002

Sample Output from PIX show Commands

pix520-1# **sh cr isa sa**

```
Total      : 1
Embryonic  : 0
  dst      src      state    pending  created
  172.16.172.39  172.16.172.34  QM_IDLE    0        1
pix520-1#
```

pix520-1# **sh cr map**

Crypto Map: "mymap" interfaces: { outside }

Crypto Map "mymap" 5 ipsec-isakmp

Peer = 172.16.172.39

access-list 130 permit ip

192.168.4.0 255.255.255.0 1.1.1.0 255.255.255.0 (hitcnt=91)

Current peer: 172.16.172.39

Security association lifetime:

4608000 kilobytes/28800 seconds

PFS (Y/N): N

Transform sets={ myset, }

pix520-1# **sh cr isa policy**

Protection suite of priority 10

encryption algorithm: DES - Data Encryption Standard (56 bit keys).

hash algorithm: Message Digest 5

authentication method: Rivest-Shamir-Adleman Signature

Diffie-Hellman group: #1 (768 bit)

lifetime: 86400 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
pix520-1#
pix520-1# sh cr ipsec sa

interface: outside
  Crypto map tag: mymap, local addr. 172.16.172.34

  local ident (addr/mask/prot/port):
  (192.168.4.0/255.255.255.0/0/0)
  remote ident (addr/mask/prot/port):
  (1.1.1.0/255.255.255.0/0/0)
  current_peer: 172.16.172.39
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 3, #pkts encrypt: 3, #pkts digest 3
    #pkts decaps: 3, #pkts decrypt: 3, #pkts verify 3
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0,
      #pkts decompress failed: 0
    #send errors 2, #recv errors 0

  local crypto endpt.: 172.16.172.34, remote
    crypto endpt.: 172.16.172.39
  path mtu 1500, ipsec overhead 56, media mtu 1500
  current outbound spi: d740971c

  inbound esp sas:
    spi: 0x3803a0c1(939761857)
      transform: esp-des esp-md5-hmac ,
      in use settings = {Tunnel, }
      slot: 0, conn id: 4, crypto map: mymap
      sa timing: remaining key lifetime
        (k/sec): (4607999/2971)
      IV size: 8 bytes
      replay detection support: Y

  inbound ah sas:

  inbound pcp sas:

  outbound esp sas:
    spi: 0xd740971c(3611334428)
      transform: esp-des esp-md5-hmac ,
      in use settings = {Tunnel, }
      slot: 0, conn id: 3, crypto map: mymap
      sa timing: remaining key lifetime
        (k/sec): (4607999/2971)
      IV size: 8 bytes
      replay detection support: Y

  outbound ah sas:

  outbound pcp sas:

pix520-1# pix520-1# sh cr en
Crypto Engine Connection Map:
  size = 8, free = 6, used = 2, active = 2
pix520-1#
```

```
pix520-1# sh clock
09:27:54 Jan 11 2002
pix520-1#
```

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

Troubleshooting Commands

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only) , which allows you to view an analysis of **show** command output.

Note: Before issuing **debug** commands, please see Important Information on Debug Commands.

The following debugs must be running on both IPSec peers:

- **debug crypto isakmp** – (Router & PIX) Displays errors during Phase 1.
- **debug crypto ipsec** – (Router & PIX) Displays errors during Phase 2.
- **debug crypto engine** – (Router only) Displays information from the crypto engine.
- **debug crypto pki transactions** – (Router only) Shows information regarding the router public key infrastructure (PKI) transactions.
- **debug crypto pki messages** – (Router only) Shows information regarding PKI input/out messages.
- **debug crypto ca** – (PIX only) Displays information regarding PKI transactions and input/output messages.

Clearing security associations must be done on both peers. The PIX commands are performed in enable mode; the router commands are performed in non-enable mode.

- **clear crypto isakmp sa** – (PIX) Clears the Phase 1 security associations.
- **clear crypto ipsec sa** – (PIX) Clears the Phase 2 security associations.
- **clear crypto isakmp** – (Router) Clears the Phase 1 security associations.
- **clear crypto sa** – (Router) Clears the Phase 2 security associations.

Sample Certificate Debug from the Router

This sections shows the debugs from the router when we run the following PKI debug commands while obtaining certificates from a CA server. These debugs were obtained during a successful session.

```
1720-1#debug cr pki transactions
Crypto PKI Trans debugging is on
1720-1#debug cr pki messages
Crypto PKI Msg debugging is on
```

```
1720-1(config)#cr ca authenticate vpn
Certificate has the following attributes:
Fingerprint: 1FCDF2C8 2DEDA6AC 4819D4C4 B4CFF2F5
% Do you accept this certificate? [yes/no]:
08:48:10: CRYPTO_PKI: Sending CA Certificate Request:
GET /cgi-bin/pkiclient.exe?operation=GetCACert&message =vpn HTTP/1.0
```

```
08:48:10: CRYPTO_PKI: can not resolve server name/IP address
08:48:10: CRYPTO_PKI: Using unresolved IP Address 171.69.89.16
08:48:10: CRYPTO_PKI: http connection opened
08:48:11: CRYPTO_PKI: HTTP response header:
```

```
HTTP/1.1 200 OK
Date: Fri, 11 Jan 2002 19:10:53 Pacific Standard Time
Server: Entrust/VPNConnector v5.0
Connection: close
Content-Type: application/x-x509-ra-ca-certs
```

Content-Type indicates we have received CA and RA certificates.

08:48:11: CRYPTO_PKI:CA and RA certs:

```
08:48:11:      30 82 08 EA 06 09 2A 86 48 86 F7 0D 01 07 02 A0
08:48:11:      82 08 DB 30 82 08 D7 02 01 01 31 00 30 0B 06 09
08:48:11:      2A 86 48 86 F7 0D 01 07 01 A0 82 08 BF 30 82 02
```

!--- Hex data omitted.

```
08:48:11:      14 06 03 55 04 03 13 0D 46 69 72 73 74 20 4F 66
08:48:11:      66 69 63 65 72 30 81 9F 30 0D 06 09 2A 86 48 86
08:48:11:      80 01 8F 51 3A 4B 61 74 59 0B 85 AA 9C E3 B3 91
08:48:11:      62 94 06 AA 7C E9 CC 0D 01 59 3E 6B 31 00
```

08:48:11:

08:48:11: CRYPTO_PKI: Error: Certificate, private key or CRL was not found while selecting certificate chain

08:48:11: CRYPTO_PKI: WARNING: A certificate chain could not be constructed while selecting certificate status

08:48:11: CRYPTO_PKI: Error: Certificate, private key or CRL was not found while selecting certificate chain

08:48:11: CRYPTO_PKI: WARNING: A certificate chain could not be constructed while selecting certificate status

```
08:48:11: CRYPTO_PKI: crypto_process_ra_certs() For:vpn
08:48:11: CRYPTO_PKI: crypto_set_ra_pubkey() (using global_auth_context)
08:48:11: CRYPTO_PKI: crypto_set_ra_pubkey() (using global_auth_context)
08:48:11: CRYPTO_PKI: transaction GetCACert completed
08:48:11: CRYPTO_PKI: CA certificate received.
08:48:11: CRYPTO_PKI: CA certificate received.
```

% Please answer 'yes' or 'no'.

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

1720-1(config)#

08:49:08: CRYPTO_PKI: crypto_process_ra_certs() For:vpn

1720-1(config)#**cr ca enroll vpn**

%

% 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:

Re-enter password:

% The subject name in the certificate will be: 1720-1.cisco.com

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

% Include an IP address in the subject name? [yes/no]: **n**

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

% Certificate request sent to Certificate Authority

% The certificate request fingerprint will be displayed.

% The 'show crypto ca certificate' command will also show

% the fingerprint.

1720-1(config)# Fingerprint: CB9730B0 5EAAEBCB CC04C77B 2B7F253D

08:51:09: CRYPTO_PKI: transaction PKCSReq completed
08:51:09: CRYPTO_PKI: status:
08:51:10: CRYPTO_PKI:Write out pkcs#10 content:272
08:51:10: 30 82 01 0C 30 81 B7 02 01 00 30 21 31 1F 30 1D
08:51:10: 06 09 2A 86 48 86 F7 0D 01 09 02 16 10 31 37 32

!--- Hex data omitted.

08:51:10: 8F 87 32 4A 25 27 2A 9B 17 F1 1F C5 67 1E 2A D2
08:51:10:
08:51:10: CRYPTO_PKI:Enveloped Data ...

08:51:10: 30 80 06 09 2A 86 48 86 F7 0D 01 07 03 A0 80 30

!--- Hex data omitted.

08:51:10: 2F C8 94 16 FE 2F 1B 00 00 00 00 00 00 00 00
08:51:10: 00
08:51:10:
08:51:10: CRYPTO_PKI:Signed Data 1311 bytes
08:51:10: 30 80 06 09 2A 86 48 86 F7 0D 01 07 02 A0 80 30
08:51:10: 80 02 01 01 31 0E 30 0C 06 08 2A 86 48 86 F7 0D

!--- Hex data omitted.

08:51:10: D0 56 7D 24 59 9C DE 00 00 00 00 00 00 00 00
08:51:10:
08:51:10: CRYPTO_PKI: can not resolve server name/IP address
08:51:10: CRYPTO_PKI: Using unresolved IP Address 171.69.89.16
08:51:10: CRYPTO_PKI: http connection opened
08:51:13: CRYPTO_PKI: received msg of 656 bytes
08:51:13: CRYPTO_PKI: HTTP response header:
HTTP/1.1 200 OK
Date: Fri, 11 Jan 2002 19:13:55 Pacific Standard Time
Server: Entrust/VPNConnector v5.0
Connection: close
Content-Type: application/x-pki-message

08:51:13: CRYPTO_PKI:Received pki message: 487 types
08:51:13: 30 82 01 E3 06 09 2A 86 48 86 F7 0D 01 07 02 A0

!--- Hex data omitted.

08:51:13: E6 E3 CC 8B 6C 5E 74 9E 6A 0B 7D E1 B7 31 A0 EF
08:51:13: 02 1B C6 F3 C2 B9 86
08:51:13:
08:51:13: CRYPTO_PKI: signed attr: pki-message-type: 13 01 33
08:51:13:
08:51:13: CRYPTO_PKI: signed attr: pki-status: 13 01 33
08:51:13:
08:51:13: CRYPTO_PKI: signed attr: pki-recipient-nonce:
08:51:13: 04 20 32 46 37 30 36 35 37 45 39 44 43 31 36 31
08:51:13: 39 31 34 39 30 32 33 34 46 35 42 44 30 46 41 31
08:51:13: 46 34
08:51:13:
08:51:13: CRYPTO_PKI: signed attr: pki-transaction-id:
08:51:13: 13 20 35 33 43 46 43 31 35 30 37 36 42 33 35 42
08:51:13: 37 30 42 43 42 39 39 36 44 36 42 46 39 32 38 30
08:51:13: 37 35
08:51:13:
08:51:13: CRYPTO_PKI: status = 102: certificate request pending
08:51:13: CRYPTO_PKI:Write out getcert initial content:84
08:51:13: 30 52 30 2D 31 0B 30 09 06 03 55 04 06 13 02 75
08:51:13: 73 31 0E 30 0C 06 03 55 04 0A 13 05 63 69 73 63

08:51:13: 6F 31 0E 30 0C 06 03 55 04 0B 13 05 73 6A 76 70
08:51:13: 6E 30 21 31 1F 30 1D 06 09 2A 86 48 86 F7 0D 01
08:51:13: 09 02 16 10 31 37 32 30 2D 31 2E 63 69 73 63 6F
08:51:13: 2E 63 6F 6D
08:51:13: CRYPTO_PKI:Enveloped Data ...
08:51:13: 30 80 06 09 2A 86 48 86 F7 0D 01 07 03 A0 80 30

!--- Hex data omitted.

08:51:13:
08:51:13: CRYPTO_PKI:Signed Data 1738 bytes
08:51:13: 30 80 06 09 2A 86 48 86 F7 0D 01 07 02 A0 80 30

!--- Hex data omitted.

08:51:14: 59 DA 00 00 00 00 00 00 00 00
08:51:14:
08:51:14: CRYPTO_PKI: can not resolve server name/IP address
08:51:14: CRYPTO_PKI: Using unresolved IP Address 171.69.89.16
08:51:14: CRYPTO_PKI: http connection opened
08:51:36: CRYPTO_PKI: received msg of 656 bytes
08:51:36: CRYPTO_PKI: HTTP response header:
HTTP/1.1 200 OK
Date: Fri, 11 Jan 2002 19:13:58 Pacific Standard Time
Server: Entrust/VPNConnector v5.0
Connection: close
Content-Type: application/x-pki-message

08:51:36: CRYPTO_PKI:Received pki message: 487 types
08:51:36: 30 82 01 E3 06 09 2A 86 48 86 F7 0D 01 07 02 A0
08:51:36: 82 01 D4 30 82 01 D0 02 01 01 31 0E 30 0C 06 08

!--- Hex data omitted.

08:51:36: E6 E3 CC 8B 6C 5E 74 9E 6A 0B 7D E1 B7 31 A0 EF
08:51:36: 02 1B C6 F3 C2 B9 86
08:51:36:
08:51:36: CRYPTO_PKI: signed attr: pki-message-type: 13 01 33
08:51:36:
08:51:36: CRYPTO_PKI: signed attr: pki-status: 13 01 33
08:51:36:
08:51:36: CRYPTO_PKI: signed attr: pki-recipient-nonce:
08:51:36: 04 20 32 46 37 30 36 35 37 45 39 44 43 31 36 31
08:51:36: 39 31 34 39 30 32 33 34 46 35 42 44 30 46 41 31
08:51:36: 46 34
08:51:36:
08:51:36: CRYPTO_PKI: signed attr: pki-transaction-id:
08:51:36: 13 20 35 33 43 46 43 31 35 30 37 36 42 33 35 42
08:51:36: 37 30 42 43 42 39 39 36 44 36 42 46 39 32 38 30
08:51:36: 37 35
08:51:36:
08:51:36: CRYPTO_PKI: status = 102: certificate request pending
08:51:46: CRYPTO_PKI: All sockets are closed.
08:51:56: CRYPTO_PKI: All sockets are closed.
08:52:36: CRYPTO_PKI: resend GetCertInitial, 1
08:52:36: CRYPTO_PKI: resend GetCertInitial for session: 0
08:52:36: CRYPTO_PKI: can not resolve server name/IP address
08:52:36: CRYPTO_PKI: Using unresolved IP Address 171.69.89.16
08:52:36: CRYPTO_PKI: http connection opened
08:52:38: CRYPTO_PKI: received msg of 1647 bytes
08:52:38: CRYPTO_PKI: HTTP response header:

```
HTTP/1.1 200 OK
Date: Fri, 11 Jan 2002 19:15:20 Pacific Standard Time
Server: Entrust/VPNConnector v5.0
Connection: close
Content-Type: application/x-pki-message
```

```
08:52:38: CRYPTO_PKI:Received pki message: 1478 types
08:52:38:      30 82 05 C2 06 09 2A 86 48 86 F7 0D 01 07 02 A0
```

!--- Hex data omitted.

```
08:52:38:      B4 0D EC 6D 61 9B
08:52:38:
08:52:38: CRYPTO_PKI: signed attr: pki-message-type: 13 01 33
08:52:38:
08:52:38: CRYPTO_PKI: signed attr: pki-status: 13 01 30
08:52:38:
08:52:38: CRYPTO_PKI: signed attr: pki-recipient-nonce:
08:52:38:      04 20 32 41 35 44 31 31 42 34 43 39 46 31 34 32
08:52:38:      30 30 38 34 32 43 35 45 38 36 44 44 43 41 45 44
08:52:38:      33 34
08:52:38:
08:52:38: CRYPTO_PKI: signed attr: pki-transaction-id:
08:52:38:      13 20 35 33 43 46 43 31 35 30 37 36 42 33 35 42
08:52:38:      37 30 42 43 42 39 39 36 44 36 42 46 39 32 38 30
08:52:38:      37 35
08:52:38:
08:52:38: CRYPTO_PKI: status = 100: certificate is granted
```

!--- Certificate is granted by the CA.

```
08:52:38: CRYPTO_PKI:Verified signed data 985 bytes:
08:52:38:      30 82 03 D5 06 09 2A 86 48 86 F7 0D 01 07 03 A0
```

!--- Hex data omitted.

```
08:52:38:      39 DE 0A 10 3B D1 17 30 79 83 E0 54 D9 59 47 13
08:52:38:      86 9A E5 5D F8 45 3D 61 63
08:52:38:
08:52:38: CRYPTO_PKI:Decrypted enveloped content:
08:52:38:      30 82 02 F3 06 09 2A 86 48 86 F7 0D 01 07 02 A0
08:52:38:      82 02 E4 30 82 02 E0 02 01 01 31 00 30 0B 06 09
```

!--- Hex data omitted.

```
08:52:39:      CE 33 54 B3 4A 62 23 65 6E B1 83 D9 7C 24 87 A5
08:52:39:      E8 FF D8 50 6F 31 00
08:52:39:
08:52:39: CRYPTO_PKI: All enrollment requests completed.
08:52:39: %CRYPTO-6-CERTRET: Certificate received from Certificate Authority
08:52:49: CRYPTO_PKI: All enrollment requests completed.
```

Sample Certificate Debug from the PIX

This sections shows the debugs from the PIX when we run the following PKI debug commands while obtaining certificates from a CA server. These debugs were obtained during a successful session.

```
pix520-1(config)#
pix520-1(config)# debug cr ca
pix520-1(config)#
```

```
pix520-1(config)# ca configure cisco ra 20 5
```

```
pix520-1(config)# ca authenticate cisco
```

```
CI thread sleeps!
```

```
Crypto CA thread wakes up!
```

```
CRYPTO_PKI: http connection opened
```

```
Certificate has the following attributes:
```

```
Fingerprint: 1fcd2c8 2deda6ac 4819d4c4 b4cff2f5
```

```
PKI: key process suspended and continued
```

```
CRYPTO_PKI: WARNING: A certificate chain could not  
be constructed while selecting certificate status
```

```
CRYPTO_PKI: WARNING: A certificate chain could not  
be constructed while selecting certificate status
```

```
CRYPTO_PKI: Name: CN = First Officer, OU = sjvpn, O = cisco, C = us
```

```
CRYPTO_PKI: Name: CN = First Officer, OU = sjvpn, O = cisco, C = us
```

```
CRYPTO_PKI: transaction GetCACert completed
```

```
CRYPTO_PKI: Name: CN = First Officer, OU = sjvpn, O = cisco, C = us
```

```
CRYPTO_PKI: Name: CN = First Officer, OU = sjvpn, O = cisco, C = us
```

```
Crypto CA thread sleeps!
```

```
pix520-1(config)# !
```

```
pix520-1(config)# sh ca cert
```

```
CA
```

```
CRYPTO_PKI: Name: OU = sjvpn, O = cisco, C = us
```

```
CRYPTO_PKI: Name: CN = CRL1, OU = sjvpn, O = cisco, C = us
```

```
CRYPTO_PKI: Name: CN = First Officer, OU = sjvpn, O = cisco, C = us
```

```
CRYPTO_PKI: Name: CN = CRL1, OU = sjvpn, O = cisco, C = us
```

```
CRYPTO_PKI: Name: CN = First Officer, OU = sjvpn, O = cisco, C = us
```

```
CRYPTO_PKI: Name: CN = CRL1, OU = sjvpn, O = cisco, C = us Certificate
```

```
Status: Available
```

```
Certificate Serial Number: 3b2fd307
```

```
Key Usage: General Purpose
```

```
OU = sjvpn
```

```
O = cisco
```

```
C = us
```

```
CRL Distribution Point:
```

```
CN = CRL1, OU = sjvpn, O = cisco, C = us
```

```
Validity Date:
```

```
start date: 22:02:40 Jun 19 2001
```

```
end date: 22:32:40 Jun 19 2021
```

```
RA Signature Certificate
```

```
Certificate Serial Number: 3b2fd319
```

```
Key Usage: Signature
```

```
CN = First Officer
```

```
OU = sjvpn
```

```
O = cisco
```

```
C = us
```

```
CRL Distribution Point:
```

```
CN = CRL1, OU = sjvpn, O = cisco, C = us
```

```
Validity Date:
```

```
start date: 22:03:31 Jun 19 2001
```

```
end date: 22:33:31 Jun 19 2004
```

```
RA KeyEncipher Certificate
```

```
Status: Available
```

```
Certificate Serial Number: 3b2fd318
```

```
Key Usage: Encryption
```

```
CN = First Officer
OU = sjvpn
O = cisco
C = us
CRL Distribution Point:
CN = CRL1, OU = sjvpn, O = cisco, C = us
Validity Date:
start date: 22:03:31 Jun 19 2001

end date: 22:33:31 Jun 19 2004
```

```
pix520-1(config)#
Status: Available
```

```
pix520-1(config)# ca enroll cisco 171.69.89.16
```

```
CI thread sleeps!
% Crypto CA thread wakes up!
% Start certificate enrollment ..
```

```
% The subject name in the certificate will be: pix520-1.vpn.com
```

```
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
```

```
pix520-1(config)# Fingerprint: bc923bc0 ee66b336 08a513b1 a226c5c8
```

```
CRYPTO_PKI: transaction PKCSReq completed
CRYPTO_PKI: status:
Crypto CA thread sleeps!
PKI: key process suspended and continued
CRYPTO_PKI: http connection opened
CRYPTO_PKI: received msg of 656 bytes
CRYPTO_PKI: WARNING: Certificate, private key or CRL was
not found while selecting CRL
```

```
CRYPTO_PKI: signed attr: pki-message-type:
13 01 33
CRYPTO_PKI: signed attr: pki-status:
13 01 33
CRYPTO_PKI: signed attr: pki-recipient-nonce:
04 20 30 36 38 33 34 44 35 46 30 44 31 37 42 39 42 30 30 44
37 37 42 33 44 37 39 42 45 43 43 43 41 41
CRYPTO_PKI: signed attr: pki-transaction-id:
13 20 64 38 32 36 37 37 34 33 31 39 62 65 65 31 62 65 34 36
65 33 63 32 38 37 66 61 65 31 31 36 64 32
CRYPTO_PKI: status = 102: certificate request pending
CRYPTO_PKI: All sockets are closed.
CRYPTO_PKI: All sockets are closed.
CRYPTO_PKI: resend GetCertInitial for session: 0
CRYPTO_PKI: http connection opened
```

```
!--- The certificate has been granted by CA!
```

```
CRYPTO_PKI: received msg of 1720 bytes
CRYPTO_PKI: WARNING: Certificate, private key or
CRL was not found while selecting CRL
```

```
PKI: key process suspended and continued
```

```

CRYPTO_PKI: signed attr: pki-message-type:
13 01 33
CRYPTO_PKI: signed attr: pki-status:
13 01 30
CRYPTO_PKI: signed attr: pki-recipient-nonce:
04 20 34 42 41 36 31 31 31 42 42 35 42 38 42 43 44 31 36 31
34 30 34 44 45 34 45 33 33 41 34 41 46 36
CRYPTO_PKI: signed attr: pki-transaction-id:
13 20 64 38 32 36 37 37 34 33 31 39 62 65 65 31 62 65 34 36
65 33 63 32 38 37 66 61 65 31 31 36 64 32
CRYPTO_PKI: status = 100: certificate is granted
CRYPTO_PKI: WARNING: Certificate, private key or CRL was not found
while selecting CRL

CRYPTO_PKI: All enrollment requests completed.
CRYPTO_PKI: All enrollment requests completed.
CRYPTO_PKI: WARNING: Certificate, private key or CRL was not found
while selecting CRL

```

Sample IPSec Debug from the Router

This section shows the IPSec debugs on the router during the time both the IPSec peers are negotiating the IPSec tunnel.

```

1720-1#debug crypto ipsec
1720-1#debug crypto isakmp
1720-1#debug crypto engine
1720-1#sh debug
Cryptographic Subsystem:
  Crypto ISAKMP debugging is on
  Crypto Engine debugging is on
  Crypto IPSEC debugging is on
1720-1#

3d11h: ISAKMP (0:0): received packet from 172.16.172.34 (N) NEW SA
3d11h: ISAKMP: local port 500, remote port 500
3d11h: ISAKMP (0:110): processing SA payload. message ID = 0
3d11h: ISAKMP (0:110): Checking ISAKMP transform 1 against
priority 10 policy
3d11h: ISAKMP:      encryption DES-CBC
3d11h: ISAKMP:      hash MD5
3d11h: ISAKMP:      default group 1
3d11h: ISAKMP:      auth RSA sig

!--- IKE phase one is accepting certificates as the authentication method.

3d11h: ISAKMP (0:110): atts are acceptable. Next payload is 3
3d11h: CryptoEngine0: generate alg parameter
3d11h: CryptoEngine0: CRYPTO_ISA_DH_CREATE(hw)(ipsec)
3d11h: CRYPTO_ENGINE: Dh phase 1 status: 0
3d11h: ISAKMP (0:110): SA is doing RSA signature authentication using
id type ID_FQDN
3d11h: ISAKMP (0:110): sending packet to 172.16.172.34 (R) MM_SA_SETUP
3d11h: ISAKMP (0:110): received packet from 172.16.172.34 (R) MM_SA_SETUP
3d11h: ISAKMP (0:110): processing KE payload. message ID = 0
3d11h: CryptoEngine0: generate alg parameter
3d11h: CryptoEngine0: CRYPTO_ISA_DH_SHARE_SECRET(hw)(ipsec)
3d11h: ISAKMP (0:110): processing NONCE payload. message ID = 0
3d11h: CryptoEngine0: calculate pkey hmac for conn id 110
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_HMAC(hw)(ipsec)
3d11h: CryptoEngine0: create ISAKMP SKEYID for conn id 110
3d11h: CryptoEngine0: CRYPTO_ISA_SA_CREATE(hw)(ipsec)
3d11h: ISAKMP (0:110): SKEYID state generated
3d11h: ISAKMP (0:110): processing CERT_REQ payload. message ID = 0
3d11h: ISAKMP (0:110): peer wants a CT_X509_SIGNATURE cert

```

```

3d11h: ISAKMP (0:110): peer want cert issued by OU = sjvpn,
O = cisco, C = us
3d11h: ISAKMP (0:110): processing vendor id payload
3d11h: ISAKMP (0:110): processing vendor id payload
3d11h: ISAKMP (0:110): processing vendor id payload
3d11h: ISAKMP (0:110): speaking to another IOS box!
3d11h: ISAKMP (0:110): sending packet to 172.16.172.34 (R) MM_KEY_EXCH
3d11h: ISAKMP (0:110): received packet from 172.16.172.34 (R) MM_KEY_EXCH
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_DECRYPT(hw)(ipsec)
3d11h: ISAKMP (0:110): processing ID payload. message ID = 0
3d11h: ISAKMP (0:110): processing CERT payload. message ID = 0
3d11h: ISAKMP (0:110): processing a CT_X509_SIGNATURE cert
3d11h: ISAKMP (0:110): processing SIG payload. message ID = 0
3d11h: ISAKMP (110): sa->peer.name = , sa->peer_id.id.id_fqdn.fqdn =
pix520-1.vpn.com
3d11h: Crypto engine 0: RSA decrypt with public key
3d11h: CryptoEngine0: CRYPTO_RSA_PUB_DECRYPT
3d11h: CryptoEngine0: generate hmac context for conn id 110
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_HMAC(hw)(ipsec)
3d11h: ISAKMP (0:110): SA has been authenticated with 172.16.172.34
3d11h: ISAKMP (110): ID payload
      next-payload : 6
      type          : 2
      protocol      : 17
      port          : 500
      length        : 20
3d11h: ISAKMP (110): Total payload length: 24
3d11h: CryptoEngine0: generate hmac context for conn id 110
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_HMAC(hw)(ipsec)
3d11h: Crypto engine 0: RSA encrypt with private key
3d11h: CryptoEngine0: CRYPTO_RSA_PRIV_ENCRYPT
3d11h: CRYPTO_ENGINE: key process suspended and continued
3d11h: CryptoEngine0: clear dh number for conn id 1
3d11h: CryptoEngine0: CRYPTO_ISA_DH_DELETE(hw)(ipsec)
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_ENCRYPT(hw)(ipsec)
3d11h: ISAKMP (0:110): sending packet to 172.16.172.34 (R) QM_IDLE
3d11h: ISAKMP (0:110): received packet from 172.16.172.34 (R) QM_IDLE
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_DECRYPT(hw)(ipsec)
3d11h: CryptoEngine0: generate hmac context for conn id 110
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_HMAC(hw)(ipsec)
3d11h: ISAKMP (0:110): processing HASH payload. message ID = -140325145
3d11h: ISAKMP (0:110): processing SA payload. message ID = -140325145
3d11h: ISAKMP (0:110): Checking IPsec proposal 1
3d11h: ISAKMP: transform 1, ESP_DES
3d11h: ISAKMP:   attributes in transform:
3d11h: ISAKMP:     encaps is 1
3d11h: ISAKMP:     SA life type in seconds
3d11h: ISAKMP:     SA life duration (basic) of 28800
3d11h: ISAKMP:     SA life type in kilobytes
3d11h: ISAKMP:     SA life duration (VPI) of 0x0 0x46 0x50 0x0
3d11h: ISAKMP:     authenticator is HMAC-MD5
3d11h: validate proposal 0
3d11h: ISAKMP (0:110): atts are acceptable.
3d11h: IPSEC(validate_proposal_request): proposal part #1,
      (key eng. msg.) INBOUND local= 172.16.172.39, remote= 172.16.172.34,
      local_proxy= 1.1.1.0/255.255.255.0/0/0 (type=4),
      remote_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
      protocol= ESP, transform= esp-des esp-md5-hmac ,
      lifedur= 0s and 0kb,
      spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
3d11h: validate proposal request 0
3d11h: ISAKMP (0:110): processing NONCE payload. message ID = -140325145
3d11h: ISAKMP (0:110): processing ID payload. message ID = -140325145
3d11h: ISAKMP (0:110): processing ID payload. message ID = -140325145
3d11h: ISAKMP (0:110): asking for 1 spis from ipsec
3d11h: IPSEC(key_engine): got a queue event...

```

```
3d11h: IPSEC(spi_response): getting spi 3611334428 for SA
      from 172.16.172.39 to 172.16.172.34 for prot 3
3d11h: ISAKMP: received ke message (2/1)
3d11h: CryptoEngine0: generate hmac context for conn id 110
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_HMAC(hw)(ipsec)
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_ENCRYPT(hw)(ipsec)
3d11h: ISAKMP (0:110): sending packet to 172.16.172.34 (R) QM_IDLE
3d11h: ISAKMP (0:110): received packet from 172.16.172.34 (R) QM_IDLE
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_DECRYPT(hw)(ipsec)
3d11h: CryptoEngine0: generate hmac context for conn id 110
3d11h: CryptoEngine0: CRYPTO_ISA_IKE_HMAC(hw)(ipsec)
3d11h: ipsec allocate flow 0
3d11h: ipsec allocate flow 0
3d11h: CryptoEngine0: CRYPTO_ISA_IPSEC_KEY_CREATE(hw)(ipsec)
3d11h: CryptoEngine0: CRYPTO_ISA_IPSEC_KEY_CREATE(hw)(ipsec)
3d11h: ISAKMP (0:110): Creating IPsec SAs
3d11h: inbound SA from 172.16.172.34 to 172.16.172.39
      (proxy 192.168.4.0 to 1.1.1.0)
3d11h: has spi 0xD740971C and conn_id 200 and flags 4
3d11h: lifetime of 28800 seconds
3d11h: lifetime of 4608000 kilobytes
3d11h: outbound SA from 172.16.172.39 to 172.16.172.34
      (proxy 1.1.1.0 to 192.168.4.0 )
3d11h: has spi 939761857 and conn_id 201 and flags C
3d11h: lifetime of 28800 seconds
3d11h: lifetime of 4608000 kilobytes
3d11h: ISAKMP (0:110): deleting node -140325145 error FALSE
reason "quick mode done (await())"
3d11h: IPSEC(key_engine): got a queue event...
3d11h: IPSEC(initialize_sas): ,
      (key eng. msg.) INBOUND local= 172.16.172.39,
      remote= 172.16.172.34,
      local_proxy= 1.1.1.0/255.255.255.0/0/0 (type=4),
      remote_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
      protocol= ESP, transform= esp-des esp-md5-hmac ,
      lifedur= 28800s and 4608000kb,
      spi= 0xD740971C(3611334428), conn_id= 200,
      keysize= 0, flags= 0x4
3d11h: IPSEC(initialize_sas): ,
      (key eng. msg.) OUTBOUND local= 172.16.172.39,
      remote= 172.16.172.34,
      local_proxy= 1.1.1.0/255.255.255.0/0/0 (type=4),
      remote_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
      protocol= ESP, transform= esp-des esp-md5-hmac ,
      lifedur= 28800s and 4608000kb,
      spi= 0x3803A0C1(939761857), conn_id= 201, keysize= 0,
      flags= 0xC
3d11h: IPSEC(create_sa): sa created,
      (sa) sa_dest= 172.16.172.39, sa_prot= 50,
      sa_spi= 0xD740971C(3611334428),
      sa_trans= esp-des esp-md5-hmac , sa_conn_id= 200
3d11h: IPSEC(create_sa): sa created,
      (sa) sa_dest= 172.16.172.34, sa_prot= 50,
      sa_spi= 0x3803A0C1(939761857),
      sa_trans= esp-des esp-md5-hmac , sa_conn_id= 201
3d11h: ISAKMP (0:108): purging SA., sa=811A823C,
delme=811A823C
3d11h: CryptoEngine0: delete connection 108
3d11h: CryptoEngine0: CRYPTO_ISA_SA_DELETE(hw)(ipsec)
3d11h: ISAKMP (0:107): purging SA., sa=811FE440,
delme=811FE440
3d11h: CryptoEngine0: delete connection 107
3d11h: CryptoEngine0: CRYPTO_ISA_SA_DELETE(hw)(ipsec)
1720-1#
```

Sample IPsec Debugs from the PIX

This section shows the IPsec debugs on the PIX during the time both IPsec peers are negotiating the IPsec tunnel.

```
pix520-1# debug crypto ipsec
pix520-1# debug crypto isakmp
pix520-1# sh debug
debug crypto ipsec 1
debug crypto isakmp 1
debug fover status
    tx      Off
    rx      Off
    open    Off
    cable   Off
    txdmp   Off
    rxdmp   Off
    ifc     Off
    rxip    Off
    txip    Off
    get     Off
    put     Off
    verify  Off
    switch  Off
    fail    Off
    fmsg    Off

ISAKMP (0): beginning Main Mode exchange

crypto_isakmp_process_block: src 172.16.172.39,
dest 172.16.172.34
OAK_MM exchange
ISAKMP (0): processing SA payload. message ID = 0

ISAKMP (0): Checking ISAKMP transform 1 against
priority 10 policy
ISAKMP:      encryption DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 1
ISAKMP:      auth RSA sig
ISAKMP (0): atts are acceptable. Next payload is 0
ISAKMP (0): SA is doing RSA signature authentication
using id type ID_FQDN
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 172.16.172.39,
dest 172.16.172.34
OAK_MM exchange
ISAKMP (0): processing KE payload. message ID = 0

ISAKMP (0): processing NONCE payload. message ID = 0

ISAKMP (0): processing CERT_REQ payload. message ID = 0
ISAKMP (0): peer wants a CT_X509_SIGNATURE cert
ISAKMP (0): processing vendor id payload

ISAKMP (0): speaking to another IOS box!

ISAKMP (0): ID payload
    next-payload : 6
    type         : 2
    protocol     : 17
    port         : 500
    length       : 20
ISAKMP (0): Total payload length: 24
return status is IKMP_NO_ERROR
```

```
crypto_isakmp_process_block: src 172.16.172.39,
dest 172.16.172.34
OAK_MM exchange
ISAKMP (0): processing ID payload. message ID = 0
ISAKMP (0): processing CERT payload. message ID = 0
ISAKMP (0): processing a CT_X509_SIGNATURE cert
ISAKMP (0): processing SIG payload. message ID = 0
ISAKMP (0): sa->peer.name = , sa->peer_id.id.id_fqdn.fqdn =
1720-1.cisco.com
ISAKMP (0): SA has been authenticated

ISAKMP (0): beginning Quick Mode exchange,
M-ID of -140325145:f7a2cee7IPSEC(key_engine):
got a queue event...
IPSEC(spi_response): getting spi 0x3803a0c1(939761857)
  for SA from 172.16.172.39 to 172.16.172.34 for prot 3

return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 172.16.172.39,
dest 172.16.172.34
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_IDLE
ISAKMP (0): processing SA payload.
message ID = 4154642151
ISAKMP : Checking IPsec proposal 1

ISAKMP: transform 1, ESP_DES
ISAKMP:   attributes in transform:
ISAKMP:     encaps is 1
ISAKMP:     SA life type in seconds
ISAKMP:     SA life duration (basic) of 28800
ISAKMP:     SA life type in kilobytes
ISAKMP:     SA life duration (VPI)
of 0x0 0x46 0x50 0x0
ISAKMP:     authenticator is HMAC-MD5
ISAKMP (0): atts are acceptable.
IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) dest= 172.16.172.39,
  src= 172.16.172.34,
  dest_proxy= 1.1.1.0/255.255.255.0/0/0 (type=4),
  src_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
  protocol= ESP, transform= esp-des esp-md5-hmac ,
  lifedur= 0s and 0kb,
  spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4

ISAKMP (0): processing NONCE payload.
message ID = 4154642151

ISAKMP (0): processing ID payload.
message ID = 4154642151
ISAKMP (0): processing ID payload.
message ID = 4154642151
ISAKMP (0): processing NOTIFY payload 24576
  protocol 3 spi 3611334428,
  message ID = 4154642151
ISAKMP (0): processing responder lifetime
ISAKMP (0): responder lifetime of 3600s
ISAKMP (0): Creating IPsec SAs
  inbound SA from 172.16.172.39 to
    172.16.172.34 (proxy 1.1.1.0 to 192.168.4.0)
  has spi 939761857 and conn_id 4 and flags 4
  lifetime of 3600 seconds
  lifetime of 4608000 kilobytes
  outbound SA from 172.16.172.34 to
    172.16.172.39 (proxy 192.168.4.0 to 1.1.1.0)
```

```

        has spi 3611334428 and conn_id 3 and flags 4
        lifetime of 3600 seconds
        lifetime of 4608000 kilobytes
IPSEC(key_engine): got a queue event...
IPSEC(initialize_sas): ,
  (key eng. msg.) dest= 172.16.172.34, src= 172.16.172.39,
  dest_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
  src_proxy= 1.1.1.0/255.255.255.0/0/0 (type=4),
  protocol= ESP, transform= esp-des esp-md5-hmac ,
  lifedur= 3600s and 4608000kb,
  spi= 0x3803a0c1(939761857), conn_id= 4, keysize= 0,
  flags= 0x4
IPSEC(initialize_sas): ,
  (key eng. msg.) src= 172.16.172.34, dest= 172.16.172.39,
  src_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
  dest_proxy= 1.1.1.0/255.255.255.0/0/0 (type=4),
  protocol= ESP, transform= esp-des esp-md5-hmac ,
  lifedur= 3600s and 4608000kb,
  spi= 0xd740971c(3611334428), conn_id= 3, keysize= 0,
  flags= 0x4

return status is IKMP_NO_ERROR

pix520-1(config)#

```

Potential Problems

This section discusses the symptoms, causes, and resolutions of common mistakes that are made while obtaining certificates on both the router and PIX.

ISAKMP Identity Mismatch

The router and PIX assign a FQDN to the keys and certificates used by IPsec. During IKE or phase 1 negotiation the router/IOS checks the FQDN in the certificate. Therefore we have to use ISAKMP identity as hostname, instead of address on both the PIX and router. In the following example, the router/IOS is checking for the FQDN in the certificate.

```

ISAKMP (0): SA is doing RSA signature authentication using
  id type ID_FQDN return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 172.16.172.39, d
  est 172.16.172.34

```

Router Debugs:

```

3d15h: CryptoEngine0: CRYPTO_ISA_DH_CREATE(hw)(ipsec)
3d15h: CRYPTO_ENGINE: Dh phase 1 status: 0
3d15h: ISAKMP (152): My ID configured as IPv4 Addr,
  but Addr not in Cert!
3d15h: ISAKMP (152): Using FQDN as My ID
3d15h: ISAKMP (0:152): SA is doing RSA signature
  authentication using id type ID_FQDN
3d15h: ISAKMP (0:152): sending packet to 172.16.172.34 (R)
  MM_SA_SETUP
3d15h: ISAKMP (0:152): received packet from 172.16.172.34 (R)
  MM_SA_SETUP

3d15h: ISAKMP (0:162): processing a CT_X509_SIGNATURE cert
3d15h: %CRYPTO-6-IKMP_NO_ID_CERT_ADDR_MATCH: ID of
  172.16.172.34 (type 1) an
  certificate addr with 172.16.172.34
3d15h: ISAKMP (0:162): processing SIG payload.

```

```
message ID = 0
3d15h: Crypto engine 0: RSA decrypt with public key
```

PIX debugs:

```
ISAKMP (0): beginning Main Mode exchange

crypto_isakmp_process_block: src 172.16.172.39, dest 172.16.172.34
OAK_MM exchange
ISAKMP (0): processing SA payload. message ID = 0

ISAKMP (0): Checking ISAKMP transform 1 against priority 10 policy
ISAKMP:      encryption DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 1
ISAKMP:      auth RSA sig
ISAKMP (0): atts are acceptable. Next payload is 0
ISAKMP (0): SA is doing RSA signature authentication using id type ID_IPV4_ADDR
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 172.16.172.39, dest 172.16.172.34
OAK_MM exchange
ISAKMP (0): processing KE payload. message ID = 0

ISAKMP (0): processing NONCE payload. message ID = 0

ISAKMP (0): processing vendor id payload

ISAKMP (0): speaking to another IOS box!

ISAKMP (0): ID payload
      next-payload : 9
      type          : 1
      protocol      : 17
      port          : 500
      length        : 8
ISAKMP (0): Total payload length: 12
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 172.16.172.39, dest 172.16.172.34
OAK_MM exchange
ISAKMP (0): processing ID payload. message ID = 0
ISAKMP (0): processing CERT payload. message ID = 0
ISAKMP (0): processing a CT_X509_SIGNATURE cert
return status is IKMP_ERR_RETRANS
```

Time and Date Mismatch

The certificates on the PIX and router are valid for a certain time interval, as demonstrated in the following example.

```
Certificate
  Status: Available
  Certificate Serial Number: 3b2fd653
  Key Usage: General Purpose
  Subject Name
    Name: pix520-1.vpn.com
  CRL Distribution Point:
    CN = CRL1, OU = sjvpn, O = cisco, C = us

  Validity Date:
```

!--- The certificates are valid between the start and end date.

```
start date: 04:13:45 Jan 11 2002
end   date: 04:43:45 Jan 11 2003
```

The following **show** command output also demonstrates the time interval.

```
1720-1#sh crypto ca crls
CRL Issuer Name:
  OU = sjvpn, O = cisco, C = us
  LastUpdate: 16:17:34 PST Jan 10 2002
  NextUpdate: 17:17:34 PST Jan 11 2002
  Retrieved from CRL Distribution Point:
    LDAP: CN = CRL1, OU = sjvpn, O = cisco, C = us
```

If the date and time of the clock on the router or PIX does not fall between the start and end dates on the certificates and the next/last update of the CRL, then you will get the following error during phase 1 negotiation:

Router debug:

```
CRYPTO_PKI: New CRL Not Yet Valid
(router time not synched to CA?)
CRL published: 16:17:34 PST Jan 10 2002
  Router time: 16:07:02 PST Feb 28 1993
    packet to
      172.16.172.34 (R) MM_KEY_EXCH
00:07:01: ISAKMP (0:10): received packet from
  172.16.172.34 (R) MM_KEY_EXCH
```

In this example, the router time was set to 16:07:02 february 28 1993, which does not fall between the valid times required by the CA. To solve the problem, set the appropriate time on the router.

```
1720-1#clock set 01:05:01 january 11 2002
1720-1#sh clock
01:05:04.903 PST Fri Jan 11 2002
1720-1#
```

HTTP/TCP port 80 Blocked

The router and the PIX use TCP port 80 during authentication and enrollment with the CA server. If you have enrollment or authentication problems, verify that HTTP/TCP port 80 is not being blocked between the router/PIX and the CA server.

PIX/Router Does not have the CRL

Since we did not specify the **crl optional** command on the PIX/router, both of these devices will check for the CRL during phase one negotiation. If the CRL is not present, you will see the following errors.

PIX debug:

```
ISAKMP (0): processing CERT payload.
  message ID = 0
ISAKMP (0): processing a CT_X509_SIGNATURE cert
CRYPTO_PKI: status = 0: poll CRL
CI thread sleeps!
Crypto CA thread wakes up!
CRYPTO_PKI: Name: CN = CRL1, OU = sjvpn,
  O = cisco, C = us
CRYPTO_PKI: ldap_bind() succeeded.
Fail to verify and insert CRL

CRYPTO_PKI: the current router time:
  02:58:08 Jan 12 2002

CRYPTO_PKI: the last CRL update time:
```

00:17:34 Jan 11 2002

CRYPTO_PKI: the next CRL update time:
01:17:34 Jan 12 2002

CRYPTO_PKI: server timer behind router:
nextUpdate: 3c3f8eae, now: 3c3fa640
CRYPTO_PKI: status = 275: failed to insert CRL
CRYPTO_PKI: transaction GetCRL completed
CRYPTO_PKI: blocking callback
received status: 105
Crypto CA thread sleeps!
CI thread wakes up!
ISAKMP (0): Unknown error in cert
validation, 65535
return status is IKMP_ERR_RETRANS

To resolve this problem, get the certificates from the CA server by issuing a **ca crl request** *ca nickname* command; we used **cr ca crl request Cisco**.

Delete Certificates and RSA Key Pairs

You may need to delete digital certificates or RSA key pairs from the router or the PIX.

Delete Router Certificates and RSA Key Pairs

Commands:

- **no crypto ca identity** *ca nickname* – Delete the router certificates.
- **crypto key zeroize rsa** – Delete the RSA key pair.

To delete the certificates, follow the example below:

```
1720-1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
1720-1(config)#no crypto ca identity vpn
% Removing an identity will destroy all certificates received from
the related Certificate Authority.

Are you sure you want to do this? [yes/no]: y
% Be sure to ask the CA administrator to revoke your certificates.

No enrollment sessions are currently active.

1720-1(config)#
1720-1#sh cr ca cert
1720-1#

!--- The certificates are no longer available.
```

To delete the RSA key pair on the router, follow the example below:

```
1720-1(config)#crypto key zeroize rsa
% Keys to be removed are named 1720-1.cisco.com.
Do you really want to remove these keys? [yes/no]: y
1720-1(config)#.

1720-1#sh crypto key mypubkey rsa
1720-1#

!--- The RSA key pairs are no longer available.
```

Delete PIX Certificates and RSA Key Pairs

Commands:

- **no ca identity *ca nickname*** – Delete the certificates from the PIX.
- **ca zeroize rsa** – Delete the RSA keypair from the PIX.

To delete the certificates on the PIX, follow the example below:

```
pix520-1(config)# no ca identity cisco
% Removing the identity will destroy all certificates.
% Be sure to ask the CA administrator to revoke your certificates.

pix520-1(config)# sh cr ca cert
pix520-1(config)#

!--- The certificates are no longer available.
```

To delete the RSA key pair on the PIX, follow the example below:

```
pix520-1(config)# ca zeroize rsa

pix520-1(config)# sh ca mypubkey rsa

!--- The RSA key pairs are no longer available.
```

Related Information

- [IPSec Support Page](#)
- [PIX Support Page](#)
- [Documentation for PIX Firewall](#)
- [PIX Command Reference](#)
- [Requests for Comments \(RFCs\)](#)
- [Technical Support – Cisco Systems](#)

All contents are Copyright © 2006–2007 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.

Updated: Feb 02, 2006

Document ID: 18014
