

Como configurar um IPsec do LAN para LAN entre um roteador e um PIX usando Certificados digitais

Índice

[Introdução](#)

[Antes de Começar](#)

[Convenções](#)

[Pré-requisitos](#)

[Componentes Utilizados](#)

[Material de Suporte](#)

[Diagrama de Rede](#)

[Configurando o roteador e o firewall de PIX](#)

[Configurações](#)

[Obter certificados](#)

[Obter certificados no roteador](#)

[Obter certificados no PIX](#)

[Verificar](#)

[Exemplo de saída dos comandos show do roteador](#)

[Saída de exemplo de comandos show de PIX](#)

[Troubleshooting](#)

[Comandos para Troubleshooting](#)

[Exemplo de depuração de certificado do roteador](#)

[Exemplo de depuração de certificado do PIX](#)

[Exemplo de depuração de IPsec do roteador](#)

[Exemplos de depurações de IPsec do PIX](#)

[Problemas potenciais](#)

[Excluir certificados e pares de chaves RSA](#)

[Informações Relacionadas](#)

Introdução

Este documento ilustra como configurar um roteador Cisco e um firewall Cisco Secure PIX para executar um IPsec de LAN para LAN usando certificados digitais. Para chegar a essa configuração, é preciso executar as seguintes tarefas:

1. Configurar o roteador e o PIX.
2. Obtenha Certificados digitais no roteador e no PIX.
3. Configurar o IKE e as políticas de IPsec no roteador e no PIX, e defina que tráfego (tráfego

interessante) será cifrado usando o IPsec através de uma lista de acessos.

Antes de Começar

Convenções

Para obter mais informações sobre convenções de documento, consulte as [Convenções de dicas técnicas Cisco](#).

Pré-requisitos

Não existem requisitos específicos para este documento.

Componentes Utilizados

As informações neste documento são baseadas nas versões de software e hardware abaixo.

- Cisco 1700 Router
- Versão de software de Cisco IOS® 12.2(6)
- Cisco PIX Firewall 520
- Versão 6.0.1 do PIX Firewall.

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se você estiver trabalhando em uma rede ativa, certifique-se de que entende o impacto potencial de qualquer comando antes de utilizá-lo.

Material de Suporte

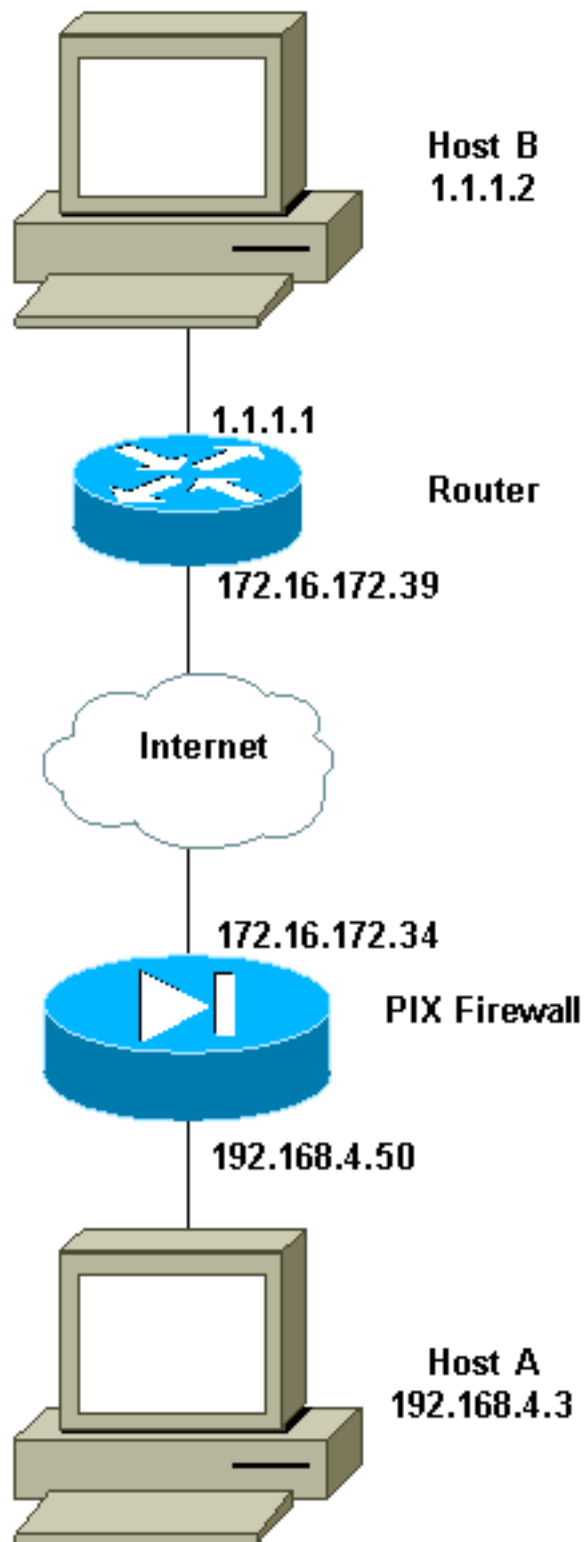
Em nosso exemplo, nós definimos o endereço de rede do host A (endereço de origem) e o endereço de rede do Host B (endereço de destino) como o tráfego que o IPsec cifrará no PIX. A lista de acesso no roteador é a imagem espelhada da lista de acesso no PIX.

Nós configuramos o PIX e o roteador de modo que os anfitriões que residem no LAN interno dos dois dispositivos usassem seus endereços privados ao atravessar o túnel de IPsec. No PIX, os comandos `access-list` e `nat 0` trabalham juntos. Quando o host A na rede de 192.168.4.0 vai à rede de 1.1.1.0, a lista de acessos permite o tráfego de rede de 192.168.4.0 ser cifrada sem Network Address Translation (NAT). Contudo, quando aqueles mesmos usuários vão em qualquer outro lugar, são traduzidos ao endereço de 172.16.172.57 através da tradução de endereço de porta (PAT). No roteador, os **comandos `route-map` and `access-list`** permitem o tráfego de rede de 1.1.1.0 ser cifrados sem NAT. Contudo, quando o mesmo Host B vai em qualquer outro lugar, são traduzidos ao endereço de 172.16.172.39 através da PANCADINHA.

Para testar a configuração, nós sibilamos do host A atrás do PIX Firewall ao Host B atrás do roteador. Quando o pacote IP chegou no PIX Firewall, combinou a lista de acesso e iniciou assim a negociação de IPsec. Assim o PIX é o iniciador e o roteador é o que responde durante a negociação de IPsec. Para propósitos de Troubleshooting, você precisa de examinar o PIX e a criptografia de roteador debuga.

Diagrama de Rede

Este documento utiliza a instalação de rede mostrada no diagrama abaixo.



[Configurando o roteador e o firewall de PIX](#)

[Configurações](#)

Nesta seção, você encontrará informações para configurar os recursos descritos neste documento.

- [Configuração de exemplo de roteador](#)

- [Configuração de exemplo PIX](#)

Configuração de exemplo de roteador

```
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$6jAs$tNxI1a/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
CDFA6E15 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
 C0EFFF2F 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 C0C25B84
CBC5A203 BF50FAC4 9656699A 52D8CB46 40776237 87163118
8F3C0F47 D2CAA36B
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

```

Configuração de exemplo PIX

```

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#
```

[Obter certificados](#)

[Obter certificados no roteador](#)

Esta seção descreve como obter Certificados digitais no roteador.

1. Configurar o nome de host do roteador e o Domain Name IP se isto não tem sido feito

```
já.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# **Nota:** O nome de host e o Domain Name são exigidos porque o roteador atribui um nome de domínio totalmente qualificado (FQDN) às chaves e aos Certificados usados pelo IPsec, com base no nome de host e no Domain Name que IP você atribui ao roteador. Por exemplo, um certificado é nomeado “router.cisco.com” baseado em um nome de host do roteador do “roteador” e em um Domain Name IP do roteador do “cisco.com.”

2. Gerencia o par de chaves RSA para o roteador, que é usado para assinar e cifrar

mensagens do gerenciamento chave IKE. Você precisa de gerar o par de chaves a fim obter um certificado para o roteador.

```
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]
```

Use o comando `show crypto key mypubkey rsa` ver o par de chaves RSA do roteador.

```
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 EBBE7ED AB93BE4B 5C1E9E6A 560A9C8A 12D7CBE3 060DBE7E
 8C1667AE 93993049 DA362602 4E4D9EF8 2F8C4777 30F9F958 7F020301 0001
```

1720-1#

3. Declare o server do Certification Authority (CA) a fim configurar os parâmetros de comunicação entre o roteador e o CA. Se nós estamos usando uma autoridade de registro, a seguir nós igualmente especificamos o modo do registration authority (RA autoridade de registro). Use o comando `crl optional` se você quer os Certificados dos outros pares ser aceitado pelo roteador mesmo se o Certificate Revocation List (CRL) apropriado não é

```
acessível ao roteador.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. O roteador precisa de autenticar CA obtendo o certificado auto-assinado de CA que contém a chave pública de CA. Porque CA assina seu próprio certificado, a chave pública de CA deve manualmente ser autenticada contactando o administrador de CA para comparar a impressão digital do certificado de CA. Neste exemplo, nós autenticamos manualmente a chave pública comparando as duas impressões digitais depois que nós recebemos o certificado de CA, um pouco do que entrando n com uma instrução de comando.

```
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
```

Use o comando `sh crypto ca cert` ver os Certificados de CA e RA e verificar que a autenticação era bem sucedida.

```
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. Obtenha um certificado assinado de CA para cada um dos pares de chaves RSA do roteador. Se você gerou chaves de uso geral RSA, o roteador tem um par de chaves RSA e precisa somente um certificado. Se você gerou chaves especiais do uso RSA, a seguir o roteador tem dois pares de chaves RSA e precisa dois Certificados. Você tem que contactar o administrador de CA para conceder manualmente os certificados de roteador se é configurado no server de CA. Também, se o server de CA é configurado de modo que você tenha que fornecer a senha na altura do registro, a seguir administrador de CA do contato para esta senha. Neste exemplo, o server de CA estabeleceu-se de modo que nós não precisássemos de fornecer uma senha durante o registro.

```

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* Depois que nós executamos os comandos para o registro, o roteador comunica-se com o server de CA e tenta-se obter seu certificado. Durante este tempo, se o server de CA é configurado para exigir uma autenticação manual dos Certificados, a seguir sua necessidade da vontade de contactar o administrador de CA. Use o comando `sh crypto ca cert` ver o certificado de roteador e verificar que o registro foi bem sucedido. No exemplo seguinte, os Certificados não foram aprovados.

```

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 **A saída de exemplo seguinte mostra
que o certificado esteve recebido de 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. Você pode manualmente pedir CA para o CRL. Para atualizar o CRL no roteador, use o comando seguinte:

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

```
1720-1(config)#exit
```

Use o comando **show crypto ca crls** ver o 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. Emita um comando **write mem** salvar a configuração. `1720-1# wr m`

```
Building configuration?
```

```
[OK]
```

```
1720-1#
```

Obter certificados no PIX

Para obter os Certificados em um PIX Firewall, você seguirá as mesmas etapas que no roteador. Contudo, a sintaxe de comando pix é diferente.

1. Ajuste o nome de host e o Domain Name IP.

```
hostname pix520-1
```

```
domain-name vpn.com
```

2. Gerencia o par de chaves RSA. `pix520-1(config)# ca generate rsa key 512`

Use o comando **show ca mypubkey rsa** indicar o par de chaves RSA. `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 o server de CA. `pix520-1(config)# ca identity cisco 171.69.89.16 171.69.89.16`

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

4. Autentique CA. `pix520-1(config)# ca authenticate cisco`

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

```
Fingerprint: 1fcdcf2c8 2deda6ac 4819d4c4 b4cff2f5
```

```

pix520-1(config)# Use o comando show ca cert ver o certificado de CA no 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. Peça CA para o 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 O certificado foi concedido por

```

```

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 o comando sh ca crl ver o 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. Para salvar os Certificados no PIX, use o comando seguinte:pix520-1(config)# ca save all

```

pix520-1(config)#

```


Esta seção fornece informações que você pode usar para confirmar se sua configuração está funcionando adequadamente.

A [Output Interpreter Tool \(somente clientes registrados\)](#) oferece suporte a determinados comandos show, o que permite exibir uma análise da saída do comando show.

Os comandos show podem ser executados no PIX e no roteador.

- **show crypto isakmp sa** - Ver todas as associações de segurança (SAs) IKE atuais no correspondente.
- **mostre IPsec cripto sa** - Mostra os ajustes usados por associações de segurança do IPsec atual.
- **show crypto engine connections active** - (Apenas roteador) Exibe as conexões atuais e as informações sobre pacotes codificados e decodificados.
- **mostre crls criptos Ca** - (roteador somente) mostra o CRL atual no roteador.
- **mostre Certificados Ca criptos** - (roteador somente) mostra o roteador, o server de CA, e os Certificados RA no roteador. Igualmente mostra o ponto de distribuição certificado (CDP).
- **show ca certificates** - (PIX somente) mostra os Certificados PIX, de CA, e RA. Ao contrário do roteador, não mostra o CDP.
- **show ca crl** - (PIX somente) mostra o CRL no PIX.
- **pulso de disparo da mostra** - Mostra as horas atual no router/PIX (do modo enable).

[Exemplo de saída dos comandos show do roteador](#)

```
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

Saída de exemplo de comandos show de PIX

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#
```

[Troubleshooting](#)

Esta seção fornece informações que podem ser usadas para o troubleshooting da sua configuração.

[Comandos para Troubleshooting](#)

A [Output Interpreter Tool](#) ([somente clientes registrados](#)) oferece suporte a determinados comandos show, o que permite exibir uma análise da saída do comando show.

Nota: Antes de emitir **comandos debug**, consulte [Informações importantes sobre comandos debug](#).

O seguinte debuga deve ser executado em ambos os ipsec peer:

- **isakmp do debug crypto** - (roteador & PIX) indica erros durante a fase 1.
- **IPsec do debug crypto** - (roteador & PIX) indica erros durante a fase 2.
- **debug crypto engine** - (somente roteador) Exibe informações sobre o mecanismo de criptografia
- **debug crypto pki transactions** - (roteador somente) mostra a informação em relação às transações do Public Key Infrastructure (PKI) do roteador.
- **debug crypto pki messages** - (roteador somente) mostra a informação em relação ao PKI entrado/para fora mensagens.
- **debug crypto ca** - (PIX somente) indica a informação em relação às transações PKI e às mensagens do entrada/saída.

A limpeza de associações de segurança deve ser feita em ambos os correspondentes. Os comandos pix são executados no modo enable; os comandos do roteador são executados no modo não habilitado.

- **clear crypto isakmp sa** - (PIX) Limpa as associações de segurança da Fase 1.
- **clear crypto ipsec sa** - (PIX) Limpa as associações de segurança da Fase 2.
- **clear crypto isakmp** - (Roteador) Limpa as associações de segurança da Fase 1.
- **clear crypto sa** - (Roteador) Limpa as associações de segurança da Fase 2.

[Exemplo de depuração de certificado do roteador](#)

Isto secciona as mostras debuga do roteador quando nós executamos os seguintes comandos debug PKI ao obter Certificados de um server de CA. Estes debugam foram obtidos durante uma sessão bem-sucedida.

```
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 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:
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.
```

Exemplo de depuração de certificado do PIX

Isto secciona as mostras debuga do PIX quando nós executamos os seguintes comandos debug PKI ao obter Certificados de um server de CA. Estes debugam foram obtidos durante uma sessão bem-sucedida.

```
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: 1fcdf2c8 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
```

Exemplo de depuração de IPsec do roteador

Esta seção mostra que o IPsec debuga no roteador durante o tempo onde ambos os ipsec peer estão negociando o túnel de IPsec.

```
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 = sjvnp, 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#

```

[Exemplos de depurações de IPsec do PIX](#)

Esta seção mostra que o IPsec debuga no PIX durante o tempo onde ambos os ipsec peer estão negociando o túnel de IPsec.

```

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)#
```

Problemas potenciais

Esta seção discute os sintomas, as causas, e as definições dos erros comuns que são feitos ao obter Certificados no roteador e no PIX.

Má combinação da identidade de ISAKMP

O roteador e o PIX atribuem um FQDN às chaves e aos Certificados usados pelo IPsec. Durante o IKE ou a negociação da fase 1 o router/IOS verifica o FQDN no certificado. Conseqüentemente nós temos que usar a identidade de ISAKMP como o hostname, em vez do endereço no PIX e no roteador. No exemplo seguinte, o router/IOS está verificando para ver se há o FQDN no certificado.

```
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)#
```

```
O roteador debuga:
```

```
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
rx dmp 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)#
```

```
O PIX debuga:
```

```
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)#
```

[Má combinação das horas e data](#)

Os Certificados no PIX e no roteador são válidos para um determinado intervalo de tempo, como demonstrado no exemplo seguinte.

```
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)#
```

O seguinte **show command output** (resultado do comando show) igualmente demonstra o intervalo de tempo.

```
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
```

Se a data e hora do pulso de disparo no roteador ou no PIX não cai entre as datas de início e de término nos Certificados e atualização seguinte/última do CRL, a seguir você obterá o seguinte erro durante a negociação da fase 1:

O roteador debuga:

```
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
```

Neste exemplo, o tempo do roteador foi ajustado 16:07:02 ao 28 de fevereiro 1993, que não cai entre o tempo válido exigido por CA. Para resolver o problema, ajuste o período apropriado no roteador.

```
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#
```

[Porta 80 HTTP/TCP obstruída](#)

O roteador e o PIX usam a porta TCP 80 durante a autenticação e o registro com o server de CA. Se você tem o registro ou os problemas de autenticação, verifique que a porta 80 HTTP/TCP não está sendo obstruída entre o router/PIX e o server de CA.

[PIX/Router não tem o CRL](#)

Desde que nós não especificamos o **comando crl optional** no PIX/router, both of these dispositivos verificarão para ver se há o CRL durante a negociação fase um. Se o CRL não está atual, você verá os seguintes erros.

PIX debug:

```
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#
```

Para resolver este problema, obtenha os Certificados do server de CA emitindo um comando do apelido *ca do pedido do crl Ca*; nós usamos o **cr ca crl request Cisco**.

[Excluir certificados e pares de chaves RSA](#)

Você pode precisar de suprimir de Certificados digitais ou de pares de chaves RSA do roteador

ou do PIX.

Certificados de roteador da supressão e pares de chaves RSA

Comandos:

- **nenhum apelido ca cripto do ca identity** - Suprima dos certificados de roteador.
- **rsa do zeroize da chave de criptografia** - Suprima do par de chaves RSA.

Para suprimir dos Certificados, siga o exemplo abaixo:

```
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.
```

Para suprimir do par de chaves RSA no roteador, siga o exemplo abaixo:

```
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.
```

Suprima de Certificados PIX e de pares de chaves RSA

Comandos:

- **nenhum apelido ca do ca identity** - Suprima dos Certificados do PIX.
- **rsa do zeroize Ca** - Suprima do par de chave RSA do PIX.

Para suprimir dos Certificados no PIX, siga o exemplo abaixo:

```
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.
```

Para suprimir do par de chaves RSA no PIX, siga o exemplo abaixo:

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

pix520-1(config)# sh ca mypubkey rsa
!--- The RSA key pairs are no longer available.
```

Informações Relacionadas

- [Página de suporte do IPSec](#)
- [Página de suporte do PIX](#)
- [Solicitações de Comentários \(RFCs\)](#)
- [Suporte Técnico - Cisco Systems](#)