Resolución de problemas de PIX para pasar el tráfico de datos en un túnel IPSec establecido

Contenido

Introducción

Este documento aborda y proporciona una solución al problema de por qué un túnel de IPSec establecido correctamente desde un Cisco VPN Client a un PIX no es capaz de pasar datos.

La incapacidad de pasar datos en un túnel IPsec establecido entre un Cliente VPN y un PIX se encuentra frecuentemente cuando no puede hacer ping o Telnet desde un Cliente VPN a cualquier host en la LAN detrás del PIX. En otras palabras, VPN Client y PIX no pueden pasar datos cifrados entre ellos. Esto ocurre porque el PIX tiene un túnel IPsec de LAN a LAN a un router y también un cliente VPN. La incapacidad para pasar datos es el resultado de una configuración con la misma lista de control de acceso (ACL) para nat 0 y el mapa criptográfico estático para el peer IPSec de LAN a LAN.

Prerequisites

Requirements

No hay requisitos específicos para este documento.

Componentes Utilizados

La información que contiene este documento se basa en las siguientes versiones de software y

hardware.

- Cisco Secure PIX Firewall 6.0.1
- Cisco 1720 Router que ejecuta Cisco IOS® Software Release 12.2(6)

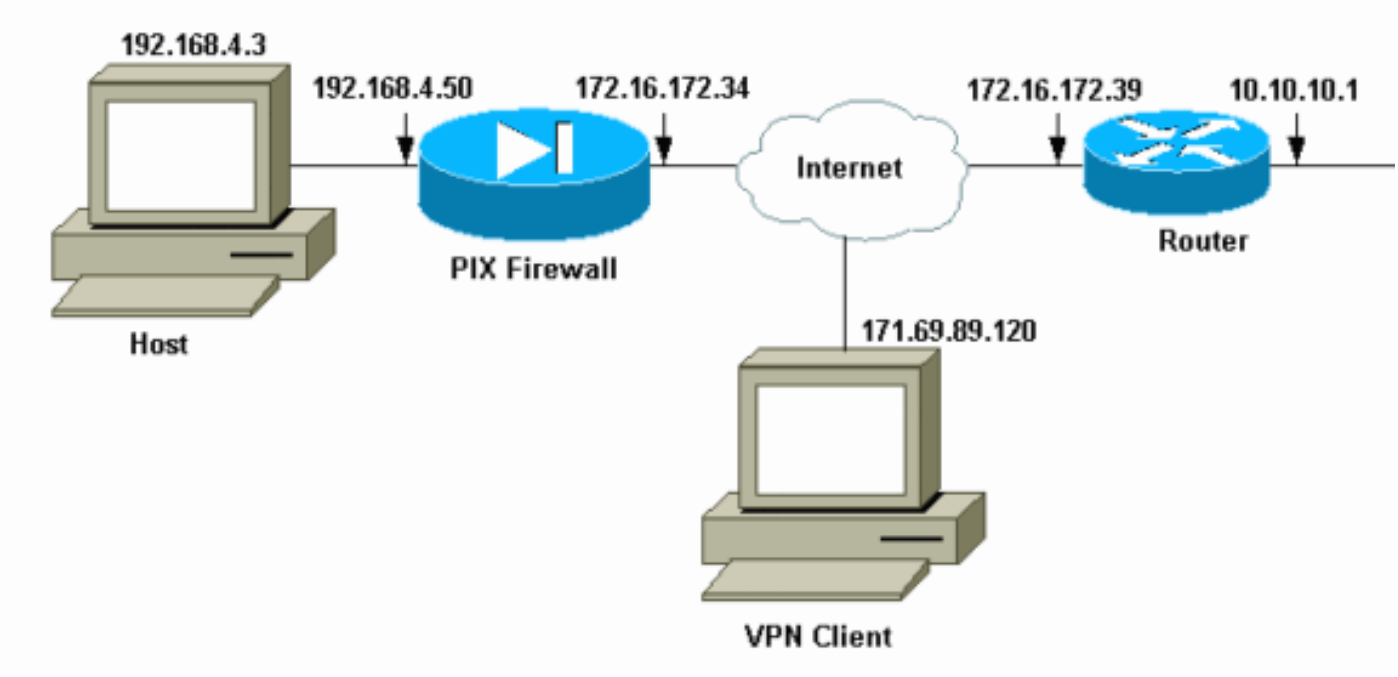
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Convenciones

Consulte Convenciones de Consejos Técnicos de Cisco para obtener más información sobre las convenciones sobre documentos.

Solución de problemas de PIX

Diagrama de la red



Configuración de ejemplo problemático

PIX 520
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 "140" defines interesting traffic to
bypass NAT for VPN !--- and defines VPN interesting
traffic. This is incorrect. access-list 140 permit ip
192.168.4.0 255.255.255.0 10.10.10.0 255.255.255.0
access-list 140 permit ip 192.168.4.0 255.255.255.0
10.1.2.0 255.255.255.0
no pager
logging on
logging console debugging
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 addresses on the outside and inside interfaces.
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
ip local pool ippool 10.1.2.1-10.1.2.254
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
!--- The nat 0 command bypasses NAT for the packets
destined over the IPsec tunnel.

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
!--- The sysopt command bypasses conduits or ACLs that
check to be applied !--- on the inbound VPN packets
```

```
after decryption.

sysopt connection permit-ipsec
no sysopt route dnat
!--- The crypto ipsec command defines IPsec encryption
and authen algo.

crypto ipsec transform-set myset esp-des esp-md5-hmac
crypto dynamic-map dynmap 10 set transform-set myset
!--- The crypto map commands define the IPsec !---
Security Assocation (SA) (Phase II SA) parameters.

crypto map mymap 5 ipsec-isakmp
crypto map mymap 5 match address 140
crypto map mymap 5 set peer 172.16.172.39
crypto map mymap 5 set transform-set myset
crypto map mymap 10 ipsec-isakmp dynamic dynmap
crypto map mymap interface outside
isakmp enable outside
!--- The isakmp key command defines the pre-shared key
for the peer address.

isakmp key ******** address 172.16.172.39 netmask
255.255.255.255 no-xauth
no-config-mode
isakmp identity address
!--- The isakmp policy defines the Phase 1 SA
parameters.

isakmp policy 10 authentication pre-share
isakmp policy 10 encryption des
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption Des
isakmp policy 20 hash sha
isakmp policy 20 group 1
isakmp policy 20 lifetime 86400
vpngroup vpn3000 address-pool ippool
vpngroup vpn3000 idle-time 1800
vpngroup vpn3000 password ********
telnet 192.168.4.0 255.255.255.0 inside
telnet 171.69.89.82 255.255.255.255 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 timeout 60
terminal width 80
Cryptochecksum:55948dc706cc700e9c10e1d24a8b125c
```

En la configuración problemática el tráfico interesante, o el tráfico que se va a cifrar para el túnel de LAN a LAN, se define por la ACL 140. La configuración utiliza la misma ACL que la ACL nat 0.

Comprender la secuencia general de eventos

Cuando un paquete IP llega a la interfaz interna del PIX, se verifica la traducción de direcciones de red (NAT). Después de eso, se comprueban las ACL para los mapas criptográficos.

- **Cómo se usa nat 0.**La ACL nat 0 define lo que no se debe incluir en NAT. La ACL en el

comando **nat 0** define la dirección de origen y destino para la cual se inhabilitan las reglas NAT en el PIX. Por lo tanto, un paquete IP que tiene una dirección de origen y de destino que coincide con la ACL definida en el comando **nat 0** omite todas las reglas NAT en el PIX.Para implementar túneles de LAN a LAN entre un PIX y otro dispositivo VPN con la ayuda de las direcciones privadas, utilice el comando **nat 0** para saltar la NAT. Las reglas en el firewall PIX impiden que las direcciones privadas se incluyan en NAT mientras estas reglas van a la LAN remota a través del túnel IPsec.

- **Cómo se utiliza la ACL crypto.**Después de las inspecciones NAT, el PIX verifica el origen y el destino de cada paquete IP que llega a su interfaz interna para que coincida con las ACL definidas en los mapas criptográficos estáticos y dinámicos. Si el PIX encuentra una coincidencia con la ACL, el PIX realiza cualquiera de estos pasos:Si no hay ninguna asociación de seguridad IPSec (SA) actual ya construida con el dispositivo IPSec de par para el tráfico, el PIX inicia las negociaciones IPsec. Una vez que se construyen las SA, cifra el paquete y lo envía a través del túnel IPsec al par IPsec.Si ya hay una SA IPsec construida con el par, el PIX cifra el paquete IP y envía el paquete cifrado al dispositivo IPsec del par.
- **ACL dinámica.**Una vez que un Cliente VPN se conecta con el PIX con la ayuda de IPsec, el PIX crea una ACL dinámica que especifica la dirección de origen y destino que se debe utilizar para definir el tráfico interesante para esta conexión IPSec.

Comprender la serie problemática de eventos en el PIX

Un error común de configuración es utilizar la misma ACL para nat 0 y los mapas criptográficos estáticos. En estas secciones se explica por qué esto provoca un error y cómo corregir el problema.

La configuración de PIX muestra que la ACL 140 de nat 0 omite NAT cuando los paquetes IP van de la red 192.168.4.0/24 a las redes 10.10.10.0/24 y 10.1.2.0/24 (dirección de red definida en el pool local de IP). Además, ACL 140 define el tráfico interesante para el mapa crypto estático para el peer 172.16.172.39.

Cuando un paquete IP llega a la interfaz interior de PIX, la verificación NAT se completa y luego el PIX verifica las ACL en los mapas criptográficos. El PIX comienza con el mapa crypto con el número de instancia más bajo. Esto se debe a que el mapa criptográfico estático del ejemplo anterior tiene el número de instancia más bajo, la ACL 140 se verifica. A continuación, se verifica la ACL dinámica para el mapa criptográfico dinámico. En esta configuración, la ACL 140 se define para cifrar el tráfico que va desde la red 192.168.4.0 /24 a las redes 10.10.10.0/24 0 y 10.1.2.0 /24. Sin embargo, para el túnel de LAN a LAN, sólo desea cifrar el tráfico entre las redes 192.168.4.0 /24 y 10.10.10.0 /24. Así es como el router de peer IPSec define su ACL crypto.

Comprender la serie problemática de eventos en el PIX

Cuando un cliente establece una conexión IPSec con el PIX, se le asigna una dirección IP del conjunto local IP. En este caso, al cliente se le asigna 10.1.2.1. El PIX también genera una ACL dinámica, como muestra este resultado del comando **show crypto map**:

```
Crypto Map "mymap" 20 ipsec-isakmp
Peer = 171.69.89.120
access-list dynacl2 permit ip host 172.16.172.34 host 10.1.2.1 (hitcnt=0)
dynamic (created from dynamic map dynmap/10)
Current peer: 171.69.89.120
```

```
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
Crypto Map "mymap" 30 ipsec-isakmp
Peer = 171.69.89.120
access-list dynacl3 permit ip any host 10.1.2.1 (hitcnt=0)
dynamic (created from dynamic map dynmap/10)
Current peer: 171.69.89.120
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
pix520-1(config)#
```

El comando **show crypto map** también muestra el mapa crypto estático:

```
Crypto Map: "mymap" interfaces: { outside }
Crypto Map "mymap" 5 ipsec-isakmp
Peer = 172.16.172.39
access-list 140 permit ip 192.168.4.0 255.255.255.0 10.10.10.0255.255.255.0
   (hitcnt=45)
access-list 140 permit ip 192.168.4.0 255.255.255.0 10.1.2.0 255.255.255.0
   (hitcnt=84)
Current peer: 172.16.172.39
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset,}
```

Una vez que se establece el túnel IPsec entre el cliente y el PIX, el cliente inicia un ping al host 192.168.4.3. Cuando recibe la solicitud de eco, el host 192.168.4.3 responde con una respuesta de eco como muestra este resultado del comando **debug icmp trace**.

```
27: Inbound ICMP echo request (len 32 id 2 seq 7680)
   10.1.2.1 > 192.168.4.3> 192.168.4.3
28: Outbound ICMP echo reply (Len 32 id 2 seq 7680)
   192.168.4.3 >192.168.4.3 > 10.1.2.1
29: Inbound ICMP echo request (Len 32 id 2 seq 7936)
   10.1.2.1 > 192.168.4.3> 192.168.4.3
30: Outbound ICMP echo reply (Len 32 id 2 seq 7936)
   192.168.4.3 >192.168.4.3 > 10.1.2.1
```

Sin embargo, la respuesta de eco no llega al VPN Client (host 10.1.2.1) y el ping falla. Puede ver esto con la ayuda del comando **show crypto ipsec sa** en el PIX. Esta salida muestra que el PIX descifra 120 paquetes que vienen del VPN Client, pero no cifra ningún paquete ni envía los paquetes cifrados al cliente. Por lo tanto, el número de paquetes encapsulados es cero.

```
pix520-1(config)#show crypto ipsec sa
interface: outside
Crypto map tag: mymap, local addr. 172.16.172.34
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (10.1.2.1/255.255.255.255/0/0)
current_peer: 171.69.89.120
dynamic allocated peer ip: 10.1.2.1
PERMIT, flags={}
#pkts encaps: 0, #pkts encrypt: 0, #pkts digest 0
!--- No packets encrypted and sent to client. #pkts decaps: 120, #pkts decrypt: 120, #pkts
verify 120
!--- 120 packets received from client. #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.: 172.16.172.34, remote crypto endpt.: 171.69.89.120
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: 33a45029
inbound esp sas:
spi: 0x279fc5e9(664782313)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 5, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607985/27809)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound pcp sas:
outbound ESP sas:
spi: 0x33a45029(866406441)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 6, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4608000/27809)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
local ident (addr/mask/prot/port): (192.168.4.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (10.10.10.0/255.255.255.0/0/0)
current_peer: 172.16.172.39
PERMIT, flags={origin_is_acl,}
#pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
#pkts decaps: 23, #pkts decrypt: 23, #pkts verify 23
#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.: 172.16.172.34, remote crypto endpt.: 172.16.172.39
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: f264e92c
inbound ESP sas:
spi: 0x2772b869(661829737)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 1, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607997/2420)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound PCP sas:
outbound ESP sas:
spi: 0xf264e92c(4066699564)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607999/2420)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
```

Nota: Cuando el host 192.168.4.3 responde a la solicitud de eco, el paquete IP llega a la interfaz interna del PIX.

```
38: Outbound ICMP echo reply (Len 32 id 2 seq 8960)
   192.168.4.3 >192.168.4.3 > 10.1.2.1
```

Una vez que el paquete IP llega a la interfaz interna, el PIX verifica la ACL nat 0 140 y determina que las direcciones de origen y destino del paquete IP coinciden con la ACL. Por lo tanto, este paquete IP omite todas las reglas NAT en el PIX. A continuación, se comprueban las ACL criptográficas. Dado que el mapa criptográfico estático tiene el número de instancia más bajo, su ACL se verifica primero. Dado que este ejemplo utiliza ACL 140 para el mapa crypto estático, el PIX verifica esta ACL. Ahora, el paquete IP tiene una dirección de origen de 192.168.4.3 y un destino de 10.1.2.1. Dado que esto coincide con la ACL 140, el PIX piensa que este paquete IP está destinado al túnel IPsec de LAN a LAN con el peer 172.16.172.39 (contrario a nuestros objetivos). Por lo tanto, verifica la base de datos SA para ver si ya hay una SA actual con el peer 172.16.72.39 para este tráfico. Como se muestra en la salida del comando **show crypto ipsec sa**, no existe ninguna SA para este tráfico. El PIX no cifra ni envía el paquete al VPN Client. En su lugar, inicia otra negociación IPSec con el peer 172.16.172.39 como muestra este resultado:

```
crypto_isakmp_process_block: src 172.16.172.39, dest 172.16.172.34
return status is IKMP_NO_ERR_NO_TRANS02303: sa_request, (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= 10.1.2.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform=
ESP-Des esp-md5-hmac , lifedur= 28800s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
702303: sa_request, (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= 10.1.2.0/255.255.255.0/0/0 (type=4), protocol= ESP, transform=
ESP-Des esp-md5-hmac , lifedur= 28800s and 4608000kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4004
ISAKMP (0): sending NOTIFY message 36137 protocol 1
return status is IKMP_NO_ERR_NO_TRANSIPSEC(key_engine): request timer
fired: count = 2,
(identity) local= 172.16.172.34, remote= 172.16.172.39,
local_proxy= 192.168.4.0/255.255.255.0/0/0 (type=4),
remote_proxy= 10.1.2.0/255.255.255.0/0/0 (type=4)
```

La negociación IPSec falla por estas razones:

- El peer 172.16.172.39 define solamente las redes 10.10.10.0/24 y 192.168.4.0/24 como el tráfico interesante en su ACL para el crypto map peer 172.16.172.34.
- Las identidades proxy no coinciden durante la negociación IPSec entre los dos peers.
- Si el par inicia la negociación y la configuración local especifica el secreto de reenvío perfecto (PFS), el par debe realizar un intercambio PFS o la negociación falla. Si la configuración local no especifica un grupo, se asume un valor predeterminado de group1 y se acepta una oferta de group1 o group2. Si la configuración local especifica group2, ese grupo debe ser parte de la oferta del par o la negociación falla. Si la configuración local no especifica PFS, acepta cualquier oferta de PFS del par. El grupo de módulos primos Diffie-Hellman de 1024 bits, group2, proporciona más seguridad que group1, pero requiere más tiempo de procesamiento que group1.**Nota:** El comando **crypto map set pfs** configura IPsec para solicitar PFS cuando solicita nuevas SA para esta entrada de crypto map. Utilice el comando **no crypto map set pfs** para especificar que IPsec no solicite PFS. Este comando sólo está disponible para entradas de mapa crypto IPsec-ISAKMP y entradas de mapa crypto dinámico. De forma predeterminada, PFS no se solicita. Con PFS, cada vez que se negocia una nueva SA, se produce un nuevo intercambio Diffie-Hellman. Esto requiere tiempo de procesamiento adicional. PFS agrega otro nivel de seguridad porque si un atacante alguna vez descifró una clave, sólo los datos enviados con esa clave se verán comprometidos. Durante la negociación, este comando hace que IPsec solicite PFS cuando solicita nuevas SA para la entrada de mapa crypto. El valor predeterminado (group1) se envía si la instrucción **set pfs** no

especifica un grupo.**Nota:** Las negociaciones IKE con un peer remoto pueden bloquearse cuando un firewall PIX tiene numerosos túneles que se originan desde el firewall PIX y terminan en un solo peer remoto. Este problema ocurre cuando PFS no está habilitado, y el peer local solicita muchas solicitudes de clave de nuevo simultáneas. Si este problema ocurre, la SA IKE no se recupera hasta que se agota o hasta que se borra manualmente con el comando **clear [crypto] isakmp sa**. Las unidades de firewall PIX configuradas con muchos túneles a muchos peers o a muchos clientes que comparten el mismo túnel no se ven afectadas por este problema. Si su configuración se ve afectada, habilite PFS con el comando **crypto map map map map seqnum set pfs**.

Los paquetes IP en el PIX se descartan en última instancia.

Comprender la solución

El método correcto para rectificar este error es definir dos ACL independientes para nat 0 y los mapas criptográficos estáticos. Para hacer esto, el ejemplo define ACL 190 para el comando **nat 0** y utiliza la ACL 140 modificada para el mapa crypto estático, como muestra este resultado.

PIX 520-1

```
pix520-1(config)#
pix520-1(config)#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 140 defines interesting traffic in
order to bypass NAT for VPN. access-list 140 permit ip
192.168.4.0 255.255.255.0 10.10.10.0255.255.255.0
!--- Defines VPN interesting traffic. access-list 190
permit ip 192.168.4.0 255.255.255.0
10.10.10.0255.255.255.0
access-list 190 permit ip 192.168.4.0 255.255.255.0
10.1.2.0 255.255.255.0
no pager
logging on
logging console debugging
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
ip local pool ippool 10.1.2.1-10.1.2.254
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
!--- The nat 0 command bypasses NAT for the packets
destined over the IPsec tunnel..

Nat (inside) 0 access-list 190
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 dynamic-map dynmap 10 set transform-set myset
!--- The crypto map commands define the IPsec SA (Phase
II SA) parameters.

crypto map mymap 5 ipsec-isakmp
crypto map mymap 5 match address 140
crypto map mymap 5 set peer 172.16.172.39
crypto map mymap 5 set transform-set myset
crypto map mymap 10 ipsec-isakmp dynamic dynmap
crypto map mymap interface outside
isakmp enable outside
isakmp key ******** address 172.16.172.39 netmask
255.255.255.255 no-xauth
no-config-mode
isakmp identity address
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption Des
isakmp policy 10 hash sha
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
isakmp policy 20 authentication pre-share
isakmp policy 20 encryption Des
isakmp policy 20 hash sha
```

```
isakmp policy 20 group 1
isakmp policy 20 lifetime 86400
vpngroup vpn3000 address-pool ippool
vpngroup vpn3000 idle-time 1800
vpngroup vpn3000 password ********
telnet 192.168.4.0 255.255.255.0 inside
telnet 171.69.89.82 255.255.255.255 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 timeout 60
terminal width 80
Cryptochecksum:e2cb98b30d3899597b3af484fae4f9ae
: end
[OK]
pix520-1(config)# pix520-1(config)#show crypto map
```

Después de que se realicen los cambios y el cliente establezca un túnel IPsec con el PIX, ejecute el comando **show crypto map**. Este comando muestra que para el mapa crypto estático, el tráfico interesante definido por la ACL 140 es sólo 192.168.4.0/24 y 10.10.10.0/24, que fue el objetivo original. Además, la lista de acceso dinámico muestra el tráfico interesante definido como el cliente (10.1.2.1) y el PIX (172.16.172.34).

```
pix520-1(config)#show crypto map
Crypto Map: "mymap" interfaces: { outside }
Crypto Map "mymap" 5 ipsec-isakmp
Peer = 172.16.172.39
access-list 140 permit ip 192.168.4.0 255.255.255.0 10.10.10.0255.255.255.0
   (hitcnt=57)
Current peer: 172.16.172.39
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
Crypto Map "mymap" 10 ipsec-isakmp
Dynamic map template tag: dynmap
Crypto Map "mymap" 20 ipsec-isakmp
Peer = 171.69.89.120
access-list dynacl4 permit ip host 172.16.172.34 host 10.1.2.1 (hitcnt=0)
dynamic (created from dynamic map dynmap/10)
Current peer: 171.69.89.120
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
Crypto Map "mymap" 30 ipsec-isakmp
Peer = 171.69.89.120
access-list dynacl5 permit ip any host 10.1.2.1 (hitcnt=13)
dynamic (created from dynamic map dynmap/10)
Current peer: 171.69.89.120
Security association lifetime: 4608000 kilobytes/28800 seconds
PFS (Y/N): N
Transform sets={ myset, }
```

Cuando VPN Client 10.1.2.1 envía un ping al host 192.168.4.3, la respuesta de eco llega a la interfaz interna del PIX. El PIX verifica el nat 0 ACL 190 y determina que el paquete IP coincide con la ACL. Por lo tanto, el paquete omite las reglas NAT en el PIX. Luego, el PIX verifica la ACL 140 de mapa crypto estático para encontrar una coincidencia. Esta vez, el origen y el destino del paquete IP no coinciden con la ACL 140. Por lo tanto, el PIX verifica la ACL dinámica y encuentra una coincidencia. El PIX entonces verifica su base de datos SA para ver si una SA IPSec ya está establecida con el cliente. Dado que el cliente ya ha establecido una conexión IPSec con el PIX,

existe una SA IPSec. El PIX luego cifra los paquetes y los envía al cliente VPN. Utilice el resultado del comando **show crypto ipsec sa** del PIX para ver que los paquetes están cifrados y descifrados. En este caso, el PIX cifró dieciséis paquetes y los envió al cliente. El PIX también recibió paquetes cifrados del VPN Client y descifró dieciséis paquetes.

```
pix520-1(config)#show crypto ipsec sa
interface: outside
Crypto map tag: mymap, local addr. 172.16.172.34
local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (10.1.2.1/255.255.255.255/0/0)
current_peer: 171.69.89.120
dynamic allocated peer ip: 10.1.2.1
PERMIT, flags={}
#pkts encaps: 16, #pkts encrypt: 16,#pkts digest 16
#pkts decaps: 16, #pkts decrypt: 16, #pkts verify 16
#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.: 172.16.172.34, remote crypto endpt.: 171.69.89.120
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: 613d083d
inbound ESP sas:
spi: 0x6adf97df(1793038303)
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): (4607998/27420)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound PCP sas:
outbound ESP sas:
spi: 0x613d083d(1631389757)
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): (4607999/27420)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
local ident (addr/mask/prot/port): (192.168.4.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (10.10.10.0/255.255.255.0/0/0)
current_peer: 172.16.172.39
PERMIT, flags={origin_is_acl,}
#pkts encaps: 9, #pkts encrypt: 9, #pkts digest 9
#pkts decaps: 9, #pkts decrypt: 9, #pkts verify 9
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. Failed: 0, #pkts decompress failed: 0
#send errors 1, #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: 58009c01
inbound ESP sas:
spi: 0x2d408709(759203593)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607998/3319)
IV size: 8 bytes
replay detection support: Y
```

```
inbound ah sas:
inbound PCP sas: outbound ESP sas:
spi: 0x58009c01(1476434945)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 1, crypto map: mymap
sa timing: remaining key lifetime (k/sec): (4607999/3319)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
pix520-1(config)# sh cr isa sa
Total : 2
Embryonic : 0
dst src state pending created
172.16.172.39 172.16.172.34 QM_IDLE 0 1
172.16.172.34 171.69.89.120 QM_IDLE 0 2
pix520-1(config)# sh cr ipsec sa
```

Configuración del router y salida del comando show

Cisco 1720-1

```
1720-1#show run
Building configuration...
Current configuration : 1592 bytes
!
! Last configuration change at 21:08:49 PST Mon Jan 7
2002
! NVRAM config last updated at 18:18:17 PST Mon Jan 7
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
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
!
!
!--- The crypto isakmp policy command defines the Phase
1 SA parameters.

crypto isakmp policy 15
authentication pre-share
crypto isakmp key cisco123 address 172.16.172.34
!
```

```
!
!--- The crypto ipsec transform-set command defines
IPsec encryption !--- and authentication algorithims.

crypto ipsec transform-set myset ESP-Des esp-md5-hmac
!
!
!--- The crypto map command defines the IPsec SA (Phase
II SA) parameters..

crypto map vpn 10 ipsec-isakmp
set peer 172.16.172.34
set transform-set myset
match address 150
!
!
!
!
!
interface FastEthernet0
ip address 172.16.172.39 255.255.255.240
speed auto
!--- The crypto map applied to the outbound interface.
crypto map vpn
interface Ethernet0
ip address 10.10.10.1 255.255.255.240
speed auto
no ip route-cache
no ip mroute-cache
!
!
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 defines interesting VPN traffic.
access-list 150 permit ip 10.10.10.0 0.0.0.255
192.168.4.0 0.0.0.255
!
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
1720-1#
```

```
1720-1#show crypto isa sa
DST src state conn-id slot
172.16.172.39 172.16.172.34 QM_IDLE 132 0
1720-1#show crypto ipsec sa
interface: FastEthernet0
Crypto map tag: vpn, local addr. 172.16.172.39
local ident (addr/mask/prot/port): (10.10.10.0/255.255.255.0/0/0)
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: 9 #pkts encrypt: 9 #pkts digest 9
#pkts decaps: 9, #pkts decrypt: 9, #pkts verify 9
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. Failed: 0, #pkts decompress failed: 0
#send errors 7, #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: 2D408709
inbound ESP sas:
spi: 0x58009C01(1476434945)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
!--- IPsec SA 200 as seen in the show crypto engine connection active command.

slot: 0, conn id: 200, flow_id: 1, crypto map: vpn
sa timing: remaining key lifetime (k/sec): (4607998/3144)
IV size: 8 bytes
replay detection support: Y
inbound ah sas:
inbound PCP sas:
outbound ESP sas:
spi: 0x2D408709(759203593)
transform: ESP-Des esp-md5-hmac ,
in use settings ={Tunnel, }
!--- IPsec SA 201 as seen in the show crypto engine connection active command.

slot: 0, conn id: 201, flow_id: 2, crypto map: vpn
sa timing: remaining key lifetime (k/sec): (4607998/3144)
IV size: 8 bytes
replay detection support: Y
outbound ah sas:
outbound PCP sas:
1720-1#

1720-1#show crypto map
Interfaces using crypto map mymap:
Crypto Map "vpn" 10 ipsec-isakmp
Peer = 172.16.172.34
Extended IP access list 150
access-list 150 permit ip 10.10.10.0 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
```

Información Relacionada

- Cisco PIX Firewall Software
- Referencias de Comandos de Cisco Secure PIX Firewall
- Avisos de campos de productos de seguridad (incluido PIX)
- Solicitudes de Comentarios (RFC)
- Negociación IPSec/Protocolos IKE
- Soporte Técnico y Documentación - Cisco Systems