Troubleshooting de PIX para Passagem de Tráfego de Dados em um Túnel de IPSec Estabelecido

Contents

Introduction

Este documento discute porque um túnel IPSec estabelecido com sucesso ligando um Cisco VPN Client a um PIX é incapaz de transmitir dados e oferece uma solução para esse problema.

A incapacidade de transmitir dados em um túnel IPsec estabelecido entre um VPN Client e um PIX é frequentemente encontrada quando você não pode fazer ping ou Telnet de um VPN Client para qualquer host na LAN atrás do PIX. Em outras palavras, o VPN Client e o PIX não podem passar dados criptografados entre eles. Isso ocorre porque o PIX tem um túnel IPsec LAN a LAN para um roteador e também um VPN Client. A incapacidade de transmitir dados é o resultado de uma configuração com a mesma lista de controle de acesso (ACL) para o nat 0 e o mapa de criptografia estático para o peer IPsec LAN a LAN.

Prerequisites

Requirements

Não existem requisitos específicos para este documento.

Componentes Utilizados

As informações neste documento são baseadas nestas versões de software e hardware:

- Cisco Secure PIX Firewall 6.0.1
- Roteador Cisco 1720 que executa o Software Cisco IOS® versão 12.2(6)

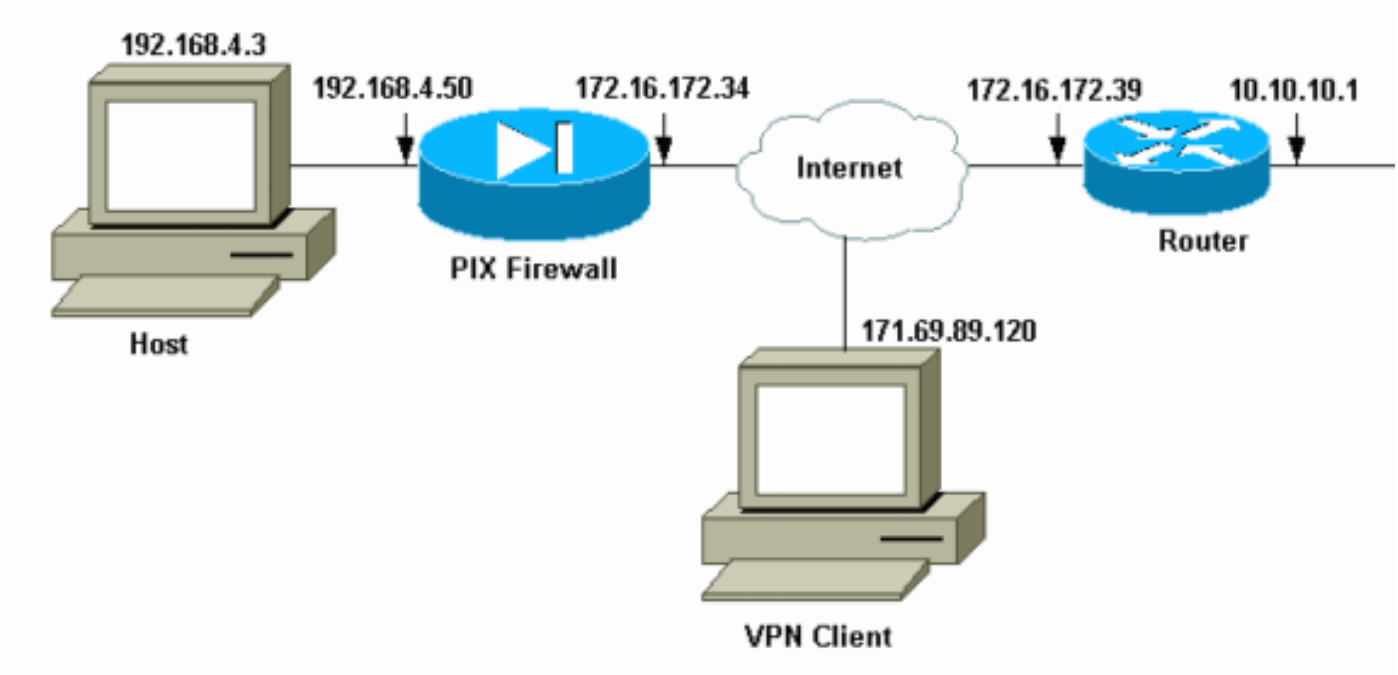
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Consulte as Convenções de Dicas Técnicas da Cisco para obter mais informações sobre convenções de documentos.

Solucionar problemas do PIX

Diagrama de Rede



Exemplo de configuração com problemas

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
```

Na configuração problemática, o tráfego interessante, ou o tráfego a ser criptografado para o túnel de LAN para LAN, é definido pela ACL 140. A configuração usa a mesma ACL que a ACL nat 0.

Entender a sequência geral de eventos

Quando um pacote IP chega à interface interna do PIX, a Network Address Translation (NAT) é verificada. Depois disso, as ACLs para os mapas de criptografia são verificadas.

- **Como se utiliza o nat 0.**A ACL nat 0 define o que não deve ser incluído na NAT. A ACL no

comando **nat 0** define o endereço de origem e de destino para os quais as regras NAT no PIX estão desativadas. Portanto, um pacote IP que tem um endereço de origem e de destino que corresponde à ACL definida no comando **nat 0** ignora todas as regras NAT no PIX.Para implementar túneis LAN a LAN entre um PIX e outro dispositivo VPN com a ajuda dos endereços privados, use o comando **nat 0** para ignorar o NAT. As regras no firewall PIX impedem que os endereços privados sejam incluídos no NAT enquanto essas regras vão para a LAN remota pelo túnel IPsec.

- **Como a ACL de criptografia é usada.**Após as inspeções de NAT, o PIX verifica a origem e o destino de cada pacote IP que chega em sua interface interna para corresponder às ACLs definidas nos mapas de criptografia estáticos e dinâmicos. Se o PIX encontrar uma correspondência com a ACL, o PIX executa qualquer uma destas etapas:Se não houver uma associação de segurança (SA) IPsec atual criada com o dispositivo IPsec peer para o tráfego, o PIX iniciará as negociações de IPsec. Depois que as SAs são criadas, ele criptografa o pacote e o envia pelo túnel IPsec para o peer IPsec.Se já houver uma SA IPsec construída com o peer, o PIX criptografa o pacote IP e envia o pacote criptografado ao dispositivo IPsec do peer.
- **ACL dinâmica.**Quando um VPN Client se conecta ao PIX com a ajuda do IPsec, o PIX cria uma ACL dinâmica que especifica o endereço de origem e de destino a serem usados para definir o tráfego interessante para essa conexão IPsec.

Entender a série problemática de eventos no PIX

Um erro de configuração comum é usar a mesma ACL para nat 0 e os mapas de criptografia estáticos. Essas seções discutem por que isso leva a um erro e como corrigir o problema.

A configuração do PIX mostra que a ACL nat 0 140 ignora o NAT quando os pacotes IP vão da rede 192.168.4.0/24 para as redes 10.10.10.0/24 e 10.1.2.0/24 (endereço de rede definido no pool local IP). Além disso, a ACL 140 define o tráfego interessante para o mapa de criptografia estático para o peer 172.16.172.39.

Quando um pacote IP chega à interface interna do PIX, a verificação do NAT é concluída e o PIX verifica as ACLs nos mapas de criptografia. O PIX começa com o mapa de criptografia com o menor número de instância. Isso porque o mapa de criptografia estático no exemplo anterior tem o menor número de instância, a ACL 140 é verificada. Em seguida, a ACL dinâmica para o mapa de criptografia dinâmico é verificada. Nesta configuração, a ACL 140 é definida para criptografar o tráfego que vai da rede 192.168.4.0 /24 para as redes 10.10.10.0/24 0 e 10.1.2.0 /24. No entanto, para o túnel LAN a LAN, você só deseja criptografar o tráfego entre as redes 192.168.4.0 /24 e 10.10.10.0 /24. É assim que o roteador peer IPsec define sua ACL de criptografia.

Entender a série problemática de eventos no PIX

Quando um cliente estabelece uma conexão IPsec com o PIX, é atribuído um endereço IP do pool local IP. Nesse caso, o cliente recebe 10.1.2.1. O PIX também gera uma ACL dinâmica, como esta saída do comando **show crypto map** mostra:

```
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)#
```

O comando **show crypto map** também mostra o mapa de criptografia 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,}
```

Quando o túnel IPsec é estabelecido entre o cliente e o PIX, o cliente inicia um ping para o host 192.168.4.3. Quando ele recebe a solicitação de eco, o host 192.168.4.3 responde com uma resposta de eco como esta saída do comando **debug icmp trace** mostra.

```
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
```

No entanto, a resposta de eco não alcança o VPN Client (host 10.1.2.1) e o ping falha. Você pode ver isso com a ajuda do comando **show crypto ipsec sa** no PIX. Esta saída mostra que o PIX descriptografa 120 pacotes que vêm do VPN Client, mas não criptografa nenhum pacote ou envia os pacotes criptografados ao cliente. Portanto, o número de pacotes encapsulados é zero.

```
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:
```

Observação: quando o host 192.168.4.3 responde à solicitação de eco, o pacote IP vem para a interface interna do PIX.

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

Quando o pacote IP chega à interface interna, o PIX verifica a ACL 140 nat 0 e determina se os endereços origem e destino do pacote IP correspondem à ACL. Portanto, esse pacote IP ignora todas as regras de NAT no PIX. Em seguida, as ACLs de criptografia são verificadas. Como o mapa de criptografia estático tem o menor número de instância, sua ACL é verificada primeiro. Como este exemplo usa a ACL 140 para o mapa de criptografia estático, o PIX verifica essa ACL. Agora, o pacote IP tem um endereço de origem 192.168.4.3 e um destino 10.1.2.1. Como isso corresponde à ACL 140, o PIX pensa que esse pacote IP é destinado ao túnel IPsec LAN a LAN com o peer 172.16.172.39 (ao contrário de nossos objetivos). Portanto, ele verifica o banco de dados SA para ver se já existe um SA atual com o peer 172.16.72.39 para esse tráfego. Como a saída do comando **show crypto ipsec sa** mostra, não existe SA para esse tráfego. O PIX não criptografa ou envia o pacote ao VPN Client. Em vez disso, ele inicia outra negociação de IPsec com o peer 172.16.172.39 como esta saída mostra:

```
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)
```

A negociação de IPsec falha por estes motivos:

- O peer 172.16.172.39 define somente as redes 10.10.10.0/24 e 192.168.4.0/24 como o tráfego interessante em sua ACL para o crypto map peer 172.16.172.34.
- As identidades de proxy não correspondem durante a negociação de IPsec entre os dois pares.
- Se o peer iniciar a negociação e a configuração local especificar PFS (Forwarding Signal, segredo de encaminhamento perfeito), o peer deverá executar uma troca PFS ou a negociação falhará. Se a configuração local não especificar um grupo, um padrão de group1 será assumido e uma oferta de group1 ou group2 será aceita. Se a configuração local especificar group2, esse grupo deverá fazer parte da oferta do peer ou a negociação falhará. Se a configuração local não especificar PFS, ela aceitará qualquer oferta de PFS do peer. O grupo Diffie-Hellman prime modulus de 1024 bits, group2, fornece mais segurança do que group1, mas requer mais tempo de processamento do que group1.**Observação:** o comando **crypto map set pfs** define o IPsec para solicitar o PFS quando solicita novas SAs para essa entrada de mapa de criptografia. Use o comando **no crypto map set pfs** para especificar que o IPsec não solicita PFS. Esse comando está disponível somente para entradas de mapa de criptografia IPsec-ISAKMP e entradas de mapa de criptografia dinâmico. Por padrão, o PFS não é solicitado. Com o PFS, toda vez que um SA novo é negociado, ocorre uma nova troca Diffie-Hellman. Isso requer tempo de processamento adicional. O PFS adiciona outro nível de segurança porque se uma chave for quebrada por um invasor, somente os dados enviados com essa chave serão comprometidos. Durante a negociação, esse comando faz com que o

IPsec solicite o PFS quando solicita novos SAs para a entrada do mapa de criptografia. O padrão (group1) será enviado se a instrução **set pfs** não especificar um grupo.**Observação:** as negociações de IKE com um peer remoto podem travar quando um firewall PIX tem vários túneis que se originam do firewall PIX e terminam em um único peer remoto. Esse problema ocorre quando o PFS não está habilitado e o peer local solicita muitas solicitações de rechaveamento simultâneas. Se esse problema ocorrer, o SA IKE não se recuperará até que o tempo limite seja excedido ou até que você o limpe manualmente com o comando **clear [crypto] isakmp sa**. As unidades de firewall PIX configuradas com muitos túneis para muitos pares ou muitos clientes que compartilham o mesmo túnel não são afetados por esse problema. Se sua configuração for afetada, ative o PFS com o comando **crypto map mapname seqnum set pfs**.

Os pacotes IP no PIX são finalmente descartados.

Entender a solução

O método correto para corrigir esse erro é definir duas ACLs separadas para nat 0 e os mapas de criptografia estáticos. Para fazer isso, o exemplo define a ACL 190 para o comando **nat 0** e usa a ACL 140 modificada para o mapa de criptografia estático, como mostra essa saída.

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
```

Depois que as alterações forem feitas e o cliente estabelecer um túnel IPsec com o PIX, emita o comando show crypto map. Esse comando mostra que para o mapa de criptografia estático, o tráfego interessante definido pela ACL 140 é apenas 192.168.4.0/24 e 10.10.10.0/24, que era o objetivo original. Além disso, a lista de acesso dinâmico mostra o tráfego interessante definido como o cliente (10.1.2.1) e o 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, }
```

Quando o VPN Client 10.1.2.1 envia um ping ao host 192.168.4.3, a resposta de eco vem para a interface interna do PIX. O PIX verifica a ACL 190 nat 0 e determina se o pacote IP corresponde à ACL. Portanto, o pacote ignora as regras de NAT no PIX. Em seguida, o PIX verifica a ACL 140 do mapa de criptografia estático para encontrar uma correspondência. Desta vez, a origem e o destino do pacote IP não correspondem à ACL 140. Portanto, o PIX verifica a ACL dinâmica e encontra uma correspondência. Em seguida, o PIX verifica seu banco de dados SA para ver se

um SA IPsec já está estabelecido com o cliente. Como o cliente já estabeleceu uma conexão IPsec com o PIX, existe uma SA IPsec. Em seguida, o PIX criptografa os pacotes e o envia ao VPN Client. Use a saída do comando **show crypto ipsec sa** do PIX para ver se os pacotes são criptografados e descriptografados. Nesse caso, o PIX criptografou dezesseis pacotes e os enviou ao cliente. O PIX também recebeu pacotes criptografados do VPN Client e descriptografou dezesseis pacotes.

```
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
```

Configuração do roteador e saída do 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
```

Informações Relacionadas

- Cisco PIX Firewall Software
- Referências do comando Cisco Secure PIX Firewall
- Avisos de campo de produto de segurança (incluindo PIX)
- Solicitações de Comentários (RFCs)
- Negociação IPsec/Protocolos IKE
- Suporte Técnico e Documentação - Cisco Systems