Dépannage de PIX de sorte qu'il permette le passage du trafic de données sur un tunnel IPSec établi

Contenu

Introduction

Ce document présente une solution au problème lors duquel un tunnel IPsec allant d'un client VPN Cisco à un PIX créé avec succès ne parvient pas à transférer des données.

L'incapacité à transmettre des données sur un tunnel IPsec établi entre un client VPN et un PIX est fréquemment rencontrée lorsque vous ne pouvez pas envoyer de requête ping ou Telnet d'un client VPN à un hôte du réseau local derrière le PIX. En d'autres termes, le client VPN et PIX ne peuvent pas transmettre des données chiffrées entre eux. Cela se produit parce que le PIX a un tunnel IPsec LAN à LAN vers un routeur et aussi un client VPN. L'incapacité à transmettre des données est le résultat d'une configuration avec la même liste de contrôle d'accès (ACL) pour la carte de chiffrement nat 0 et la carte de chiffrement statique pour l'homologue IPsec LAN à LAN.

Conditions préalables

Conditions requises

Aucune spécification déterminée n'est requise pour ce document.

Components Used

Les informations contenues dans ce document sont basées sur les versions de matériel et de logiciel suivantes :

- Cisco Secure PIX Firewall 6.0.1
- Routeur Cisco 1720 qui exécute le logiciel Cisco IOS® Version 12.2(6)

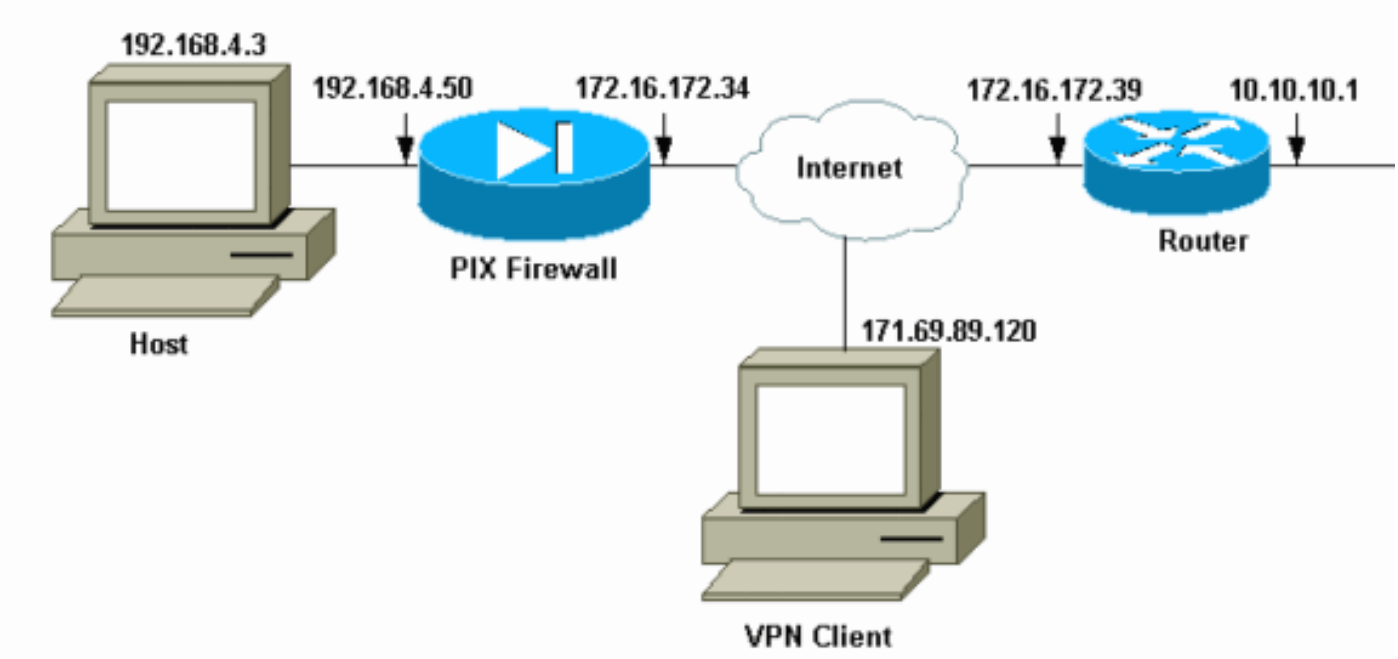
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Pour plus d'informations sur les conventions utilisées dans ce document, reportez-vous à Conventions relatives aux conseils techniques Cisco.

Dépannage du PIX

Diagramme du réseau



Exemple de configuration problématique

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
```

Dans la configuration problématique le trafic intéressant, ou le trafic à chiffrer pour le tunnel LAN à LAN, est défini par la liste de contrôle d'accès 140. La configuration utilise la même liste de contrôle d'accès que la liste de contrôle d'accès nat 0.

Comprendre la séquence générale des événements

Lorsqu'un paquet IP arrive à l'interface interne du PIX, la traduction d'adresses de réseau (NAT) est vérifiée. Après cela, les listes de contrôle d'accès pour les cartes de chiffrement sont vérifiées.

- **Utilisation de nat 0.**La liste de contrôle d’accès nat 0 définit ce qui ne doit pas être inclus dans la NAT. La liste de contrôle d'accès de la commande **nat 0** définit l'adresse source et de destination pour laquelle les règles NAT sur le PIX sont désactivées. Par conséquent, un paquet IP dont l'adresse source et de destination correspond à la liste de contrôle d'accès définie dans la commande **nat 0** ignore toutes les règles NAT du PIX.Afin de mettre en oeuvre des tunnels LAN à LAN entre un PIX et un autre périphérique VPN à l'aide d'adresses privées, utilisez la commande **nat 0** pour contourner NAT. Les règles du pare-feu PIX empêchent l'inclusion des adresses privées dans NAT pendant que ces règles vont au réseau local distant via le tunnel IPsec.
- **Utilisation de la liste de contrôle d’accès de chiffrement.**Après les inspections NAT, le PIX vérifie la source et la destination de chaque paquet IP qui arrive à son interface interne pour qu'elles correspondent aux listes de contrôle d'accès définies dans les cartes de chiffrement statiques et dynamiques. Si le PIX trouve une correspondance avec la liste de contrôle d'accès, le PIX effectue l'une des étapes suivantes :S'il n'y a pas d'association de sécurité IPsec (SA) déjà créée avec le périphérique IPsec homologue pour le trafic, le PIX lance les négociations IPsec. Une fois les SA créées, elles chiffrent le paquet et l'envoient par le tunnel IPsec à l'homologue IPsec.S'il existe déjà une SA IPsec créée avec l'homologue, le PIX chiffre le paquet IP et envoie le paquet chiffré au périphérique IPsec homologue.
- **ACL dynamique.**Une fois qu'un client VPN se connecte au PIX à l'aide d'IPsec, le PIX crée une liste de contrôle d'accès dynamique qui spécifie l'adresse source et de destination à utiliser afin de définir le trafic intéressant pour cette connexion IPsec.

Comprendre la série d'événements problématiques sur PIX

Une erreur de configuration courante consiste à utiliser la même liste de contrôle d’accès pour nat 0 et les crypto-cartes statiques. Ces sections expliquent pourquoi cela entraîne une erreur et comment corriger le problème.

La configuration PIX montre que la liste de contrôle d'accès nat 0 140 contourne NAT lorsque des paquets IP passent du réseau 192.168.4.0/24 aux réseaux 10.10.10.0/24 et 10.1.2.0/24 (adresse réseau définie dans le pool local d'adresses IP ipool). En outre, la liste de contrôle d'accès 140 définit le trafic intéressant pour la carte de chiffrement statique pour l'homologue 172.16.172.39.

Lorsqu'un paquet IP arrive à l'interface interne PIX, la vérification NAT se termine, puis le PIX vérifie les ACL dans les crypto-cartes. Le PIX commence par la crypto-carte avec le numéro d'instance le plus bas. En effet, la crypto-carte statique de l'exemple précédent a le numéro d'instance le plus bas, la liste de contrôle d'accès ACL 140 est cochée. Ensuite, la liste de contrôle d'accès dynamique de la carte de chiffrement dynamique est vérifiée. Dans cette configuration, la liste de contrôle d’accès 140 est définie pour chiffrer le trafic qui va du réseau 192.168.4.0 /24 aux réseaux 10.10.10.0/24 0 et 10.1.2.0 /24. Cependant, pour le tunnel LAN à LAN, vous voulez seulement chiffrer le trafic entre les réseaux 192.168.4.0 /24 et 10.10.10.0 /24. Voici comment le routeur homologue IPsec définit sa liste de contrôle d'accès de chiffrement.

Comprendre la série d'événements problématiques sur PIX

Lorsqu'un client établit une connexion IPsec au PIX, une adresse IP lui est attribuée à partir du pool local d'adresses IP. Dans ce cas, le client est affecté à 10.1.2.1. Le PIX génère également une liste de contrôle d'accès dynamique, comme le montre cette sortie de commande **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)#
```

La commande **show crypto map** affiche également la carte de chiffrement statique :

```
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,}
```

Une fois que le tunnel IPsec est établi entre le client et le PIX, le client lance une requête ping vers l'hôte 192.168.4.3. Lorsqu'il reçoit la requête d'écho, l'hôte 192.168.4.3 répond avec une réponse d'écho comme le montre la sortie de la commande **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
```

Cependant, la réponse d'écho n'atteint pas le client VPN (hôte 10.1.2.1) et la requête ping échoue. Vous pouvez le voir à l'aide de la commande **show crypto ipsec sa** sur le PIX. Cette sortie montre que le PIX déchiffre 120 paquets qui proviennent du client VPN, mais il ne chiffre aucun paquet ni n'envoie les paquets chiffrés au client. Par conséquent, le nombre de paquets encapsulés est égal à zéro.

```
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:
```

Remarque : lorsque l'hôte 192.168.4.3 répond à la requête d'écho, le paquet IP arrive à l'interface interne du PIX.

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

Une fois que le paquet IP arrive à l'interface interne, le PIX vérifie la liste de contrôle d'accès nat 0 140 et détermine que les adresses source et de destination du paquet IP correspondent à la liste de contrôle d'accès. Par conséquent, ce paquet IP contourne toutes les règles NAT sur le PIX. Ensuite, les listes de contrôle d'accès de chiffrement sont vérifiées. Puisque la carte de chiffrement statique a le numéro d'instance le plus bas, sa liste de contrôle d'accès est d'abord vérifiée. Puisque cet exemple utilise ACL 140 pour la crypto-carte statique, le PIX vérifie cette ACL. Maintenant, le paquet IP a une adresse source 192.168.4.3 et une destination 10.1.2.1. Puisque cela correspond à la liste de contrôle d'accès 140, le PIX pense que ce paquet IP est destiné au tunnel IPsec LAN à LAN avec l'homologue 172.16.172.39 (contrairement à nos objectifs). Par conséquent, il vérifie la base de données SA pour voir s'il existe déjà une SA actuelle avec l'homologue 172.16.72.39 pour ce trafic. Comme le montre la sortie de la commande **show crypto ipsec sa**, il n'existe aucune SA pour ce trafic. Le PIX ne chiffre ni n'envoie le paquet au client VPN. Au lieu de cela, il initie une autre négociation IPsec avec l'homologue 172.16.172.39 comme le montre le résultat suivant :

```
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 négociation IPsec échoue pour les raisons suivantes :

- L'homologue 172.16.172.39 définit uniquement les réseaux 10.10.10.0/24 et 192.168.4.0/24 comme le trafic intéressant de sa liste de contrôle d'accès pour l'homologue de carte de chiffrement 172.16.172.34.
- Les identités proxy ne correspondent pas lors de la négociation IPsec entre les deux homologues.
- Si l'homologue initie la négociation et que la configuration locale spécifie le secret de transmission parfait (PFS), l'homologue doit effectuer un échange PFS ou la négociation échoue. Si la configuration locale ne spécifie pas de groupe, une valeur par défaut de group1 est supposée et une offre de group1 ou group2 est acceptée. Si la configuration locale spécifie le groupe2, ce groupe doit faire partie de l'offre de l'homologue ou la négociation échoue. Si la configuration locale ne spécifie pas PFS, elle accepte toute offre de PFS de l'homologue. Le groupe de modules principaux Diffie-Hellman de 1 024 bits, group2, fournit plus de sécurité que group1, mais nécessite plus de temps de traitement que

group1.**Remarque :** La commande **crypto map set pfs** définit IPsec pour demander PFS lorsqu'il demande de nouvelles SA pour cette entrée de crypto map. Utilisez la commande **no crypto map set pfs** pour spécifier qu'IPsec ne demande pas PFS. Cette commande est uniquement disponible pour les entrées crypto-map IPsec-ISAKMP et les entrées crypto-map dynamiques. Par défaut, PFS n'est pas demandé. Avec PFS, chaque fois qu'une nouvelle SA est négociée, un nouvel échange Diffie-Hellman se produit. Cela nécessite du temps de traitement supplémentaire. PFS ajoute un autre niveau de sécurité, car si une clé est craquée par un pirate, seules les données envoyées avec cette clé sont compromises. Pendant la négociation, cette commande entraîne IPsec à demander PFS lorsqu'il demande de nouvelles SA pour l'entrée de crypto-carte. La valeur par défaut (group1) est envoyée si l'instruction **set pfs** ne spécifie pas de groupe.**Remarque :** les négociations IKE avec un homologue distant peuvent se suspendre lorsqu'un pare-feu PIX a de nombreux tunnels qui proviennent du pare-feu PIX et se terminent sur un homologue distant unique. Ce problème se produit lorsque PFS n'est pas activé et que l'homologue local demande de nombreuses demandes de retouche simultanée. Si ce problème se produit, l'association de sécurité IKE ne se rétablit pas tant qu'elle n'est pas arrivée à expiration ou jusqu'à ce que vous l'effaciez manuellement à l'aide de la commande **clear [crypto] isakmp sa**. Les unités de pare-feu PIX configurées avec de nombreux tunnels vers de nombreux homologues ou de nombreux clients qui partagent le même tunnel ne sont pas affectées par ce problème. Si votre configuration est affectée, activez PFS avec la commande **crypto map mapname seqnum set pfs**.

Les paquets IP sur le PIX sont finalement abandonnés.

Comprendre la solution

La méthode correcte pour corriger cette erreur consiste à définir deux listes de contrôle d'accès distinctes pour nat 0 et les crypto-cartes statiques. Pour ce faire, l'exemple définit la liste de contrôle d'accès 190 pour la commande **nat 0** et utilise la liste de contrôle d'accès 140 modifiée pour la carte de chiffrement statique, comme le montre ce résultat.

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
```

Une fois les modifications effectuées et que le client établit un tunnel IPsec avec le PIX, émettez la commande **show crypto map**. Cette commande montre que pour la carte de chiffrement statique, le trafic intéressant défini par la liste de contrôle d'accès 140 est seulement 192.168.4.0/24 et 10.10.10.0/24, ce qui était l'objectif initial. En outre, la liste d'accès dynamique affiche le trafic intéressant défini comme le client (10.1.2.1) et le 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, }
```

Lorsque le client VPN 10.1.2.1 envoie une requête ping à l'hôte 192.168.4.3, la réponse d'écho arrive à l'interface interne du PIX. Le PIX vérifie la liste de contrôle d'accès nat 0 190 et détermine que le paquet IP correspond à la liste de contrôle d'accès. Par conséquent, le paquet contourne les règles NAT sur le PIX. Ensuite, le PIX vérifie l'ACL 140 de crypto-carte statique afin de trouver une correspondance. Cette fois, la source et la destination du paquet IP ne correspondent pas à la liste de contrôle d'accès 140. Par conséquent, le PIX vérifie la liste de contrôle d'accès dynamique et trouve une correspondance. Le PIX vérifie ensuite sa base de données SA pour voir si une SA IPsec est déjà établie avec le client. Puisque le client a déjà établi une connexion IPsec avec le PIX, une SA IPsec existe. Le PIX chiffre ensuite les paquets et les envoie au client VPN. Utilisez la sortie de commande **show crypto ipsec sa** du PIX pour voir que les paquets sont chiffrés et déchiffrés. Dans ce cas, le PIX a chiffré seize paquets et les a envoyés au client. Le PIX a également reçu des paquets chiffrés du client VPN et déchiffré seize paquets.

```
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
```

Configuration du routeur et sortie de la commande 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
```

Informations connexes

- Logiciels pare-feu Cisco PIX
- Références des commandes du pare-feu Cisco Secure PIX
- Notices de champs relatives aux produits de sécurité (y compris PIX)
- Demandes de commentaires (RFC)
- Négociation IPSec/Protocoles IKE
- Support et documentation techniques - Cisco Systems