Fehlerbehebung beim PIX zur Weiterleitung des Datenverkehrs auf einem etablierten IPSec-Tunnel

Inhalt

Einführung
Voraussetzungen
Anforderungen
Verwendete Komponenten
Konventionen
Fehlerbehebung für PIX
Netzwerkdiagramm
Problematische Beispielkonfiguration
Allgemeine Ereignisreihenfolge
Kennenlernen der problematischen Ereignisserie auf dem PIX
Kennenlernen der problematischen Ereignisserie auf dem PIX
Verständnis der Lösung
Routerkonfiguration und Ausgabe von Befehlen
Zugehörige Informationen

Einführung

Dieses Dokument behandelt und bietet eine Lösung für das Problem, warum ein erfolgreich eingerichteter IPsec-Tunnel von einem Cisco VPN-Client zu einem PIX keine Daten übergeben kann.

Die Unfähigkeit, Daten in einem etablierten IPsec-Tunnel zwischen einem VPN-Client und einem PIX zu übertragen, tritt häufig auf, wenn Sie von einem VPN-Client aus keine Ping- oder Telnet-Daten an Hosts im LAN hinter dem PIX senden können. Anders ausgedrückt: Der VPN-Client und PIX können keine verschlüsselten Daten zwischen ihnen weitergeben. Dies liegt daran, dass der PIX über einen LAN-zu-LAN-IPsec-Tunnel zu einem Router und auch einen VPN-Client verfügt. Die Unfähigkeit, Daten zu übergeben, ist das Ergebnis einer Konfiguration mit derselben Zugriffskontrollliste (ACL) für die nat 0 und die statische Crypto Map für den LAN-to-LAN IPsec-Peer.

Voraussetzungen

Anforderungen

Für dieses Dokument bestehen keine speziellen Anforderungen.

Verwendete Komponenten

Die Informationen in diesem Dokument basieren auf den folgenden Software- und Hardwareversionen:

- Cisco Secure PIX Firewall 6.0.1
- Cisco 1720 Router mit Cisco IOS® Softwareversion 12.2(6)

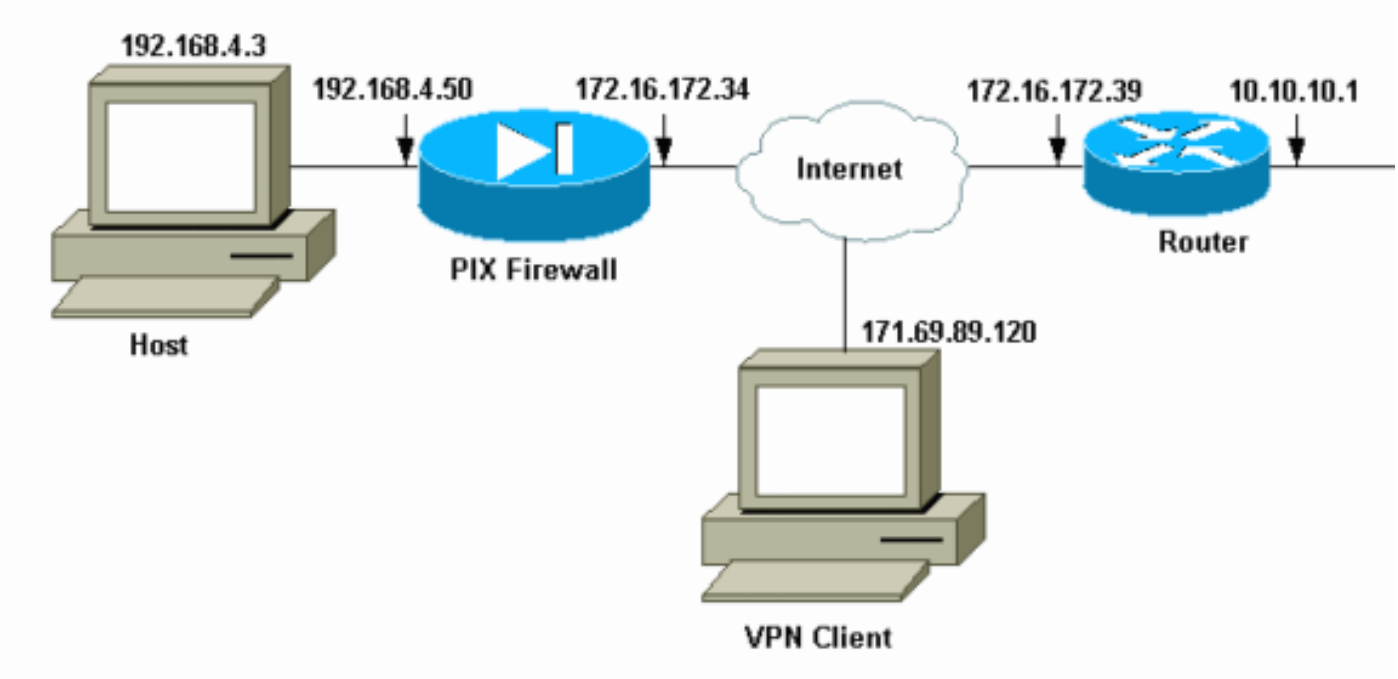
Die Informationen in diesem Dokument wurden von den Geräten in einer bestimmten Laborumgebung erstellt. Alle in diesem Dokument verwendeten Geräte haben mit einer leeren (Standard-)Konfiguration begonnen. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die potenziellen Auswirkungen eines Befehls verstehen.

Konventionen

Weitere Informationen zu Dokumentkonventionen finden Sie in den Cisco Technical Tips Conventions (Technische Tipps zu Konventionen von Cisco).

Fehlerbehebung für PIX

Netzwerkdiagramm



Problematische Beispielkonfiguration

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
```

In der problematischen Konfiguration wird der interessante Datenverkehr bzw. der für den LAN-zu-LAN-Tunnel zu verschlüsselnde Datenverkehr durch die ACL 140 definiert. Die Konfiguration verwendet dieselbe ACL wie die nat 0-ACL.

Allgemeine Ereignisreihenfolge

Wenn ein IP-Paket an der internen Schnittstelle des PIX ankommt, wird die Network Address Translation (NAT) überprüft. Danach werden die Zugriffskontrolllisten für die Crypto-Maps überprüft.

- **Wie wird nat 0 angewendet?**Die ACL Nat 0 definiert, was nicht in NAT enthalten sein sollte. Die ACL im Befehl **nat 0** definiert die Quell- und Zieladresse, für die die NAT-Regeln auf dem PIX deaktiviert sind. Aus diesem Grund umgeht ein IP-Paket mit einer Quell- und Zieladresse, die der im Befehl **nat 0** definierten ACL entspricht, alle NAT-Regeln auf dem PIX.Um LAN-zu-LAN-Tunnel zwischen einem PIX und einem anderen VPN-Gerät mithilfe der privaten Adressen zu implementieren, verwenden Sie den Befehl **nat 0**, um NAT zu umgehen. Die Regeln der PIX-Firewall verhindern, dass private Adressen in NAT enthalten sind, während diese Regeln über den IPsec-Tunnel zum Remote-LAN gelangen.
- **Verwendung der Krypto-ACL**Nach den NAT-Inspektionen überprüft das PIX die Quelle und das Ziel jedes IP-Pakets, das an seiner internen Schnittstelle eingeht, auf die in den statischen und dynamischen Crypto Maps definierten ACLs. Wenn das PIX eine Übereinstimmung mit der ACL findet, führt das PIX einen der folgenden Schritte aus:Wenn keine aktuelle IPsec Security Association (SA) vorhanden ist, die bereits mit dem Peer-IPsec-Gerät für den Datenverkehr erstellt wurde, initiiert das PIX die IPsec-Aushandlungen. Nachdem die SAs erstellt wurden, verschlüsselt sie das Paket und sendet es über den IPsec-Tunnel an den IPsec-Peer.Wenn bereits eine mit dem Peer erstellte IPsec-SA vorhanden ist, verschlüsselt das PIX das IP-Paket und sendet das verschlüsselte Paket an das Peer-IPsec-Gerät.
- **Dynamische ACL.**Sobald ein VPN-Client mithilfe von IPsec eine Verbindung mit dem PIX herstellt, erstellt das PIX eine dynamische ACL, die die Quell- und Zieladresse angibt, die zum Definieren des interessanten Datenverkehrs für diese IPsec-Verbindung verwendet werden soll.

Kennenlernen der problematischen Ereignisserie auf dem PIX

Ein häufiger Konfigurationsfehler besteht darin, dieselbe ACL für Nat 0 und die statischen Crypto Maps zu verwenden. In diesen Abschnitten wird erläutert, warum dies zu einem Fehler führt und wie das Problem behoben werden kann.

Die PIX-Konfiguration zeigt, dass die NAT von der nat 0 ACL 140 umgangen wird, wenn IP-Pakete von Netzwerk 192.168.4.0/24 zu den Netzwerken 10.10.10.0/24 und 10.1.2.0/24 (Netzwerkadresse definiert im IP Local Pool) gehen. Darüber hinaus definiert die ACL 140 den interessanten Datenverkehr für die statische Crypto Map für Peer 172.16.172.39.

Wenn ein IP-Paket an die PIX-interne Schnittstelle gesendet wird, wird die NAT-Prüfung abgeschlossen, und anschließend überprüft das PIX die ACLs in den Crypto Maps. Der PIX beginnt mit der Crypto Map mit der niedrigsten Instanznummer. Der Grund hierfür ist, dass die statische Crypto Map im vorherigen Beispiel die niedrigste Instanznummer hat, die ACL 140 aktiviert ist. Als Nächstes wird die dynamische ACL für die dynamische Crypto Map überprüft. In dieser Konfiguration ist die ACL 140 definiert, um Datenverkehr zu verschlüsseln, der vom Netzwerk 192.168.4.0/24 zu den Netzwerken 10.10.10.0/24 0 und 10.1.2.0 /24 geleitet wird. Für den LAN-to-LAN-Tunnel sollten Sie jedoch nur den Datenverkehr zwischen den Netzwerken 192.168.4.0 /24 und 10.10.10.0 /24 verschlüsseln. So definiert der IPsec-Peer-Router seine Krypto-ACL.

Kennenlernen der problematischen Ereignisserie auf dem PIX

Wenn ein Client eine IPsec-Verbindung mit dem PIX herstellt, wird ihm eine IP-Adresse aus dem lokalen IP-Pool zugewiesen. In diesem Fall wird dem Client 10.1.2.1 zugewiesen. Das PIX generiert auch eine dynamische Zugriffskontrollliste, wie die Ausgabe des Befehls **crypto map** zeigt:

```
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)#
```

Der Befehl **show crypto map** zeigt auch die statische Crypto Map an:

```
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,}
```

Sobald der IPsec-Tunnel zwischen dem Client und dem PIX erstellt wurde, initiiert der Client einen Ping an den Host 192.168.4.3. Wenn der Host die Echo-Anforderung empfängt, antwortet der Host 192.168.4.3 mit einer Echo-Antwort, da diese Ausgabe des Befehls **debug icmp trace** angezeigt wird.

```
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
```

Die Echoantwort erreicht jedoch nicht den VPN-Client (Host 10.1.2.1), und der Ping schlägt fehl. Sie können dies mithilfe des Befehls **show crypto ipsec sa** auf dem PIX sehen. Diese Ausgabe zeigt, dass das PIX 120 vom VPN-Client stammende Pakete entschlüsselt, jedoch keine Pakete verschlüsselt oder verschlüsselte Pakete an den Client sendet. Aus diesem Grund ist die Anzahl der gekapselten Pakete Null.

```
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:
```

Hinweis: Wenn der Host 192.168.4.3 auf die Echoanforderung antwortet, gelangt das IP-Paket zur internen Schnittstelle des PIX.

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

Sobald das IP-Paket an der internen Schnittstelle ankommt, überprüft das PIX die nat 0 ACL 140 und bestimmt, dass die Quell- und Zieladresse des IP-Pakets mit der ACL übereinstimmt. Daher umgeht dieses IP-Paket alle NAT-Regeln auf dem PIX. Als Nächstes werden die Krypto-ACLs überprüft. Da die statische Crypto Map die niedrigste Instanznummer hat, wird die zugehörige ACL zuerst überprüft. Da in diesem Beispiel ACL 140 für die statische Crypto Map verwendet wird, überprüft das PIX diese ACL. Das IP-Paket hat nun die Quelladresse 192.168.4.3 und das Ziel 10.1.2.1. Da dies mit der ACL 140 übereinstimmt, geht das PIX davon aus, dass dieses IP-Paket für den LAN-zu-LAN IPsec-Tunnel mit Peer 172.16.172.39 vorgesehen ist (im Gegensatz zu unseren Zielen). Daher überprüft sie die SA-Datenbank, um festzustellen, ob für diesen Datenverkehr bereits ein aktuelles SA mit Peer 172.16.72.39 vorhanden ist. Wie die Ausgabe des Befehls **show crypto ipsec sa** zeigt, existiert für diesen Datenverkehr kein SA. Das PIX verschlüsselt das Paket nicht und sendet es nicht an den VPN-Client. Stattdessen wird eine weitere IPsec-Aushandlung mit Peer 172.16.172.39 initiiert, wie in dieser Ausgabe Folgendes angezeigt wird:

```
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)
```

Die IPsec-Aushandlung schlägt aus folgenden Gründen fehl:

- Der Peer 172.16.172.39 definiert nur die Netzwerke 10.10.10.0/24 und 192.168.4.0/24 als den interessanten Datenverkehr in seiner ACL für die Crypto Map Peer 172.16.172.34.
- Die Proxy-Identitäten stimmen bei der IPsec-Aushandlung zwischen den beiden Peers nicht überein.
- Wenn der Peer die Aushandlung initiiert und die lokale Konfiguration PFS (Perfect Forward Secrecy) angibt, muss der Peer einen PFS-Austausch durchführen, oder die Aushandlung schlägt fehl. Wenn in der lokalen Konfiguration keine Gruppe angegeben ist, wird von der

Standardeinstellung für group1 ausgegangen, und es wird ein Angebot von group1 oder group2 akzeptiert. Wenn die lokale Konfiguration die Gruppe2 angibt, muss diese Gruppe Teil des Angebots des Peers sein, oder die Aushandlung schlägt fehl. Wenn die lokale Konfiguration kein PFS angibt, akzeptiert sie jedes PFS-Angebot vom Peer. Die 1024-Bit-Diffie-Hellman-Primmodulusgruppe group2 bietet mehr Sicherheit als group1, erfordert jedoch mehr Verarbeitungszeit als group1.**Hinweis:** Der Befehl **crypto map set pfs** gibt IPsec so ein, dass er PFS anfordert, wenn er neue SAs für diesen Crypto Map-Eintrag anfordert. Verwenden Sie den Befehl **no crypto map set pfs**, um anzugeben, dass IPsec kein PFS anfordert. Dieser Befehl ist nur für Krypto-Map-Einträge in IPsec-ISAKMP und dynamische Einträge in der Crypto Map verfügbar. PFS wird standardmäßig nicht angefordert. Bei PFS findet jedes Mal, wenn eine neue SA ausgehandelt wird, ein neuer Diffie-Hellman-Austausch statt. Dies erfordert zusätzliche Verarbeitungszeit. PFS bietet eine weitere Sicherheitsstufe, da nur die mit diesem Schlüssel gesendeten Daten kompromittiert werden, wenn ein Schlüssel jemals von einem Angreifer geknackt wird. Während der Aushandlung veranlasst dieser Befehl IPsec, PFS anzufordern, wenn er neue SAs für den Crypto Map-Eintrag anfordert. Der Standardwert (group1) wird gesendet, wenn die **set pfs-**Anweisung keine Gruppe angibt.**Hinweis:** IKE-Verhandlungen mit einem Remote-Peer können hängen, wenn eine PIX-Firewall über zahlreiche Tunnel verfügt, die von der PIX-Firewall stammen und auf einem einzigen Remote-Peer enden. Dieses Problem tritt auf, wenn PFS nicht aktiviert ist und der lokale Peer viele gleichzeitige rekey-Anfragen anfordert. Wenn dieses Problem auftritt, wird die IKE SA erst nach einer Zeitüberschreitung wiederhergestellt oder manuell mit dem Befehl **clear [crypto] isakmp sa**. PIX-Firewall-Einheiten, die mit vielen Tunneln zu vielen Peers konfiguriert sind, oder viele Clients, die denselben Tunnel gemeinsam nutzen, sind von diesem Problem nicht betroffen. Wenn Ihre Konfiguration betroffen ist, aktivieren Sie PFS mit dem Befehl **crypto map mapname seqnum set pfs**.

Die IP-Pakete auf dem PIX werden letztendlich verworfen.

Verständnis der Lösung

Die richtige Methode zur Behebung dieses Fehlers besteht in der Definition zweier separater ACLs für Nat 0 und der statischen Crypto Maps. Dazu wird im Beispiel die ACL 190 für den Befehl **nat 0** definiert und die geänderte ACL 140 für die statische Crypto Map verwendet, wie diese Ausgabe zeigt.

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
```

Nachdem die Änderungen vorgenommen wurden und der Client einen IPsec-Tunnel mit dem PIX erstellt hat, geben Sie den Befehl **show crypto map** ein. Dieser Befehl zeigt, dass für die statische Crypto Map der durch die ACL 140 definierte interessante Datenverkehr nur 192.168.4.0/24 und 10.10.10.0/24 ist, was das ursprüngliche Ziel war. Darüber hinaus zeigt die dynamische Zugriffsliste den als Client (10.1.2.1) und PIX (172.16.172.34) definierten interessanten Datenverkehr an.

```
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, }
```

Wenn der VPN-Client 10.1.2.1 einen Ping an den Host 192.168.4.3 sendet, wird die Echoantwort an die interne Schnittstelle des PIX gesendet. Das PIX überprüft die nat 0 ACL 190 und stellt fest, dass das IP-Paket mit der ACL übereinstimmt. Daher umgeht das Paket die NAT-Regeln auf dem PIX. Anschließend überprüft das PIX die statische Crypto Map ACL 140, um eine Übereinstimmung zu finden. Diesmal stimmen Quelle und Ziel des IP-Pakets nicht mit der ACL 140 überein. Daher überprüft das PIX die dynamische ACL und findet eine Übereinstimmung. Das PIX überprüft dann seine SA-Datenbank, um festzustellen, ob bereits eine IPsec-SA mit dem Client erstellt wurde. Da der Client bereits eine IPsec-Verbindung mit dem PIX aufgebaut hat, existiert eine IPsec-SA. Das PIX verschlüsselt die Pakete anschließend und sendet sie an den VPN-Client. Verwenden Sie den Befehl **show crypto ipsec als** Ausgabe des PIX, um zu überprüfen, ob Pakete verschlüsselt und entschlüsselt sind. In diesem Fall verschlüsselte das PIX sechzehn Pakete und schickte sie an den Client. Die PIX empfing außerdem verschlüsselte Pakete vom VPN-Client und entschlüsselte sechzehn Pakete.

```
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
```

Routerkonfiguration und Ausgabe von Befehlen

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 authenticathims.

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
```

Zugehörige Informationen

- Cisco PIX Firewall-Software
- Cisco Secure PIX Firewall - Befehlsreferenzen
- Problemhinweise zu Sicherheitsprodukten (einschließlich PIX)
- Anforderungen für Kommentare (RFCs)
- IPsec-Aushandlung/IKE-Protokolle
- Technischer Support und Dokumentation - Cisco Systems