PIX Firewall für Inbound Host Translation in einem Remote-Netzwerk verbunden über L2L IPsec-Tunnel - Konfigurationsbeispiel

Inhalt

Einführung
Voraussetzungen
Anforderungen
Verwendete Komponenten
Konventionen
Konfigurieren
Netzwerkdiagramm
Konfigurationen
Clear Security Associations (SAs)
Überprüfen
PIXfirst verifizieren
PIXsecond verifizieren
Fehlerbehebung
Befehle zur Fehlerbehebung
Zugehörige Informationen

Einführung

Dieses Dokument beschreibt die Schritte zur Übersetzung der Quell-IP eines Hosts, der über einen LAN-zu-LAN-IPsec-Tunnel zwischen zwei Cisco Secure PIX-Firewalls übertragen wird. Jede PIX-Firewall verfügt über ein privates, geschütztes Netzwerk. Dieses Konzept gilt auch, wenn Sie statt einzelner Hosts Subnetze übersetzen.

Hinweis: Führen Sie die folgenden Schritte aus, um dasselbe Szenario in PIX/ASA 7.x zu konfigurieren:

- Informationen zum Konfigurieren eines Site-to-Site-VPN-Tunnels für PIX/ASA 7.x finden Sie unter PIX/ASA 7.x: Einfaches PIX-to-PIX VPN-Tunnel-Konfigurationsbeispiel.
- Der **statische** Befehl für die eingehende Kommunikation ist für 6.x und 7.x ähnlich, wie in diesem Dokument beschrieben.
- Die in diesem Dokument verwendeten Befehle **show**, **clear** und **debug** sind in PIX 6.x und 7.x ähnlich.

Voraussetzungen

Anforderungen

Vergewissern Sie sich, dass Sie die PIX-Firewall mit IP-Adressen an den Schnittstellen konfiguriert haben und über eine grundlegende Konnektivität verfügen, bevor Sie mit diesem Konfigurationsbeispiel fortfahren.

Verwendete Komponenten

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

- Cisco PIX 506E-Firewall
- Cisco Secure PIX Firewall-Software Version 6.3(3)

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).

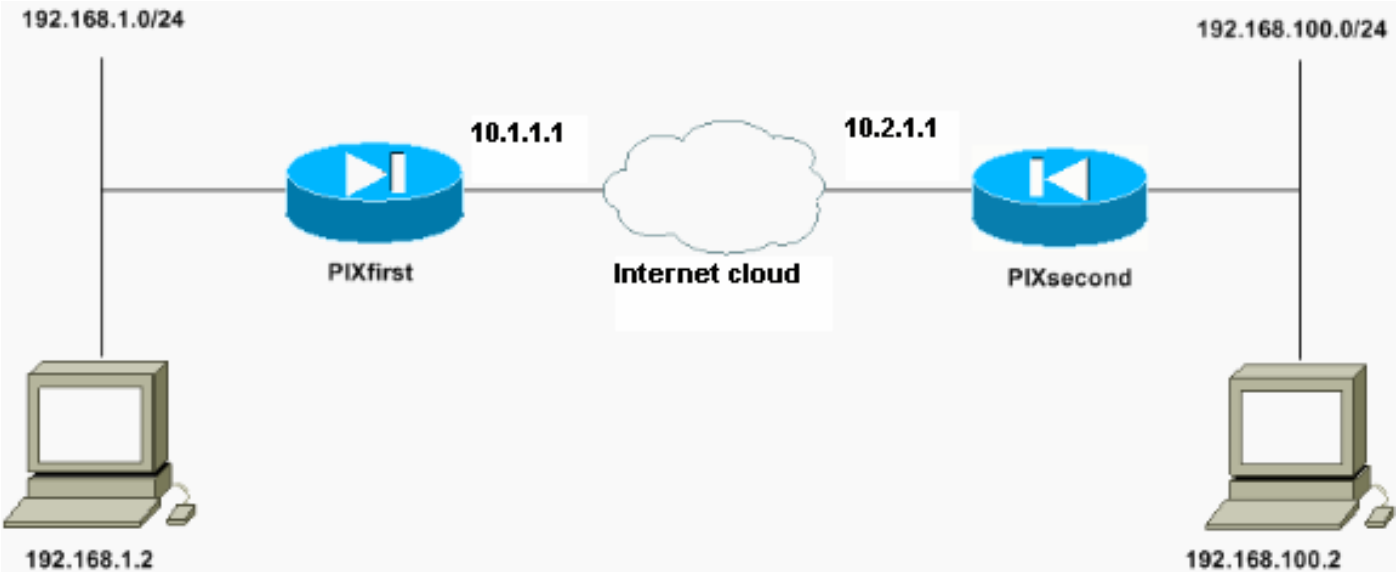
Konfigurieren

In diesem Abschnitt erhalten Sie Informationen zum Konfigurieren der in diesem Dokument beschriebenen Funktionen.

Hinweis: Verwenden Sie das Command Lookup Tool (nur registrierte Kunden), um weitere Informationen zu den in diesem Abschnitt verwendeten Befehlen zu erhalten.

Netzwerkdiagramm

In diesem Dokument wird die folgende Netzwerkeinrichtung verwendet:



Der Host mit der IP-Adresse 192.168.100.2 wird in 192.168.50.2 auf der PIX-Firewall mit dem Hostnamen PIXfirst übersetzt. Diese Übersetzung ist für den Host und sein Ziel transparent.

Hinweis: Eingebettete IP-Adressen werden nicht standardmäßig übersetzt, es sei denn, eine Fixup-Datei für diese Anwendung ist aktiviert. Eine eingebettete IP-Adresse ist eine, die die Anwendung innerhalb des Datennutzlastteils eines IP-Pakets enthält. Network Address Translation (NAT) ändert nur den äußeren IP-Header eines IP-Pakets. Sie ändert nicht die Datennutzlast des ursprünglichen Pakets, in das IPs von bestimmten Anwendungen eingebettet werden können. Dies führt manchmal dazu, dass diese Anwendungen nicht ordnungsgemäß funktionieren.

Konfigurationen

In diesem Dokument werden folgende Konfigurationen verwendet:

- PIXfirst-Konfiguration
- PIXsecond-Konfiguration

PIXfirst-Konfiguration

```
PIXfirst(config)#write terminal

Building configuration...

: Saved

:

PIX Version 6.3(3)
interface ethernet0 auto
interface ethernet1 auto
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 2KFQnbNIdI.2KYOU encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname PIXfirst
fixup protocol dns maximum-length 512
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol sip 5060
fixup protocol sip udp 5060
fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol tftp 69
names
!--- Define encryption domain (interesting traffic) !---
for the IPsec tunnel. access-list 110 permit ip host
192.168.1.2 host 192.168.100.2

!--- Accept the private network traffic from the NAT
process. access-list 120 permit ip host 192.168.1.2 host
192.168.50.2
pager lines 24
```

```
mtu outside 1500
mtu inside 1500
ip address outside 10.1.1.1 255.255.255.0
ip address inside 192.168.1.1 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
pdm history enable
arp timeout 14400

!--- Bypass translation for traffic that goes over the
IPsec tunnel. nat (inside) 0 access-list 120

!--- Inbound translation for the host located on the
remote network. static (outside,inside) 192.168.50.2
192.168.100.2 netmask 255.255.255.255 0 0
route outside 0.0.0.0 0.0.0.0 10.1.1.2 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc
0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 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 LOCAL protocol local
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable


!--- Accept traffic that comes over the IPsec tunnel
from !--- Adaptive Security Algorithm (ASA) rules and !-
-- access control lists (ACLs) configured on the outside
interface. sysopt connection permit-ipsec

!--- Create the Phase 2 policy for actual data
encryption. crypto ipsec transform-set chevelle esp-des
esp-md5-hmac
crypto map transam 1 ipsec-isakmp
crypto map transam 1 match address 110
crypto map transam 1 set peer 10.2.1.1
crypto map transam 1 set transform-set chevelle
crypto map transam interface outside
isakmp enable outside

!--- Pre-shared key for the IPsec peer. isakmp key
******** address 10.2.1.1 netmask 255.255.255.255

!--- Create the Phase 1 policy. isakmp identity address
isakmp policy 1 authentication pre-share
isakmp policy 1 encryption des
isakmp policy 1 hash md5
isakmp policy 1 group 1
isakmp policy 1 lifetime 1000
telnet timeout 5
ssh timeout 5
console timeout 0
terminal width 80
Cryptochecksum:778f934d42c037a978b8b5236a93b5f4

: end
```

```
[OK]

PIXfirst(config)#
```

PIXsecond-Konfiguration

```
PIXsecond(config)#write terminal

Building configuration...

: Saved

:

PIX Version 6.3(3)
interface ethernet0 auto
interface ethernet1 auto
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 2KFQnbNIdI.2KYOU encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname PIXsecond
fixup protocol dns maximum-length 512
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol sip 5060
fixup protocol sip udp 5060
fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol tftp 69
names

!--- Accept the private network traffic from the NAT
process. access-list nonat permit ip host 192.168.100.2
host 192.168.1.2

!--- Define encryption domain (interesting traffic) for
the IPsec tunnel. access-list 110 permit ip host
192.168.100.2 host 192.168.1.2
pager lines 24
mtu outside 1500
mtu inside 1500
ip address outside 10.2.1.1 255.255.255.0
ip address inside 192.168.100.1 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
pdm history enable
arp timeout 14400

!--- Bypass translation for traffic that goes over the
IPsec tunnel. nat (inside) 0 access-list nonat
route outside 0.0.0.0 0.0.0.0 10.2.1.2 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc
0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 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 LOCAL protocol local
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable

!--- Accept traffic that comes over the IPsec tunnel
from ASA rules and !--- ACLs configured on the outside
interface. sysopt connection permit-ipsec

!--- Create the Phase 2 policy for actual data
encryption. crypto ipsec transform-set chevelle esp-des
esp-md5-hmac
crypto map transam 1 ipsec-isakmp
crypto map transam 1 match address 110
crypto map transam 1 set peer 10.1.1.1
crypto map transam 1 set transform-set chevelle
crypto map transam interface outside
isakmp enable outside

!--- Pre-shared key for the IPsec peer. isakmp key
******** address 10.1.1.1 netmask 255.255.255.255

!--- Create the Phase 1 policy. isakmp identity address
isakmp policy 1 authentication pre-share
isakmp policy 1 encryption des
isakmp policy 1 hash md5
isakmp policy 1 group 1
isakmp policy 1 lifetime 1000
telnet timeout 5
ssh timeout 5
console timeout 0
terminal width 80
Cryptochecksum:a686f71a023d1cd7078728a38acf529e

: end

[OK]

PIXsecond(config)#
```

Wenn Sie mehr als einen Crypto Map-Eintrag für eine bestimmte Schnittstelle erstellen, müssen Sie die Sequenznummer jedes Eintrags verwenden, um ihn einzustufen. Je niedriger die Sequenznummer, desto höher ist die Priorität. An der Schnittstelle, für die die Crypto Map festgelegt ist, wird der Datenverkehr zuerst anhand der Einträge der Zuordnung mit höherer Priorität bewertet.

Erstellen Sie mehrere Crypto Map-Einträge für eine bestimmte Schnittstelle, wenn verschiedene Peers unterschiedliche Datenflüsse behandeln oder wenn Sie unterschiedliche IPsec-Sicherheit auf verschiedene Arten von Datenverkehr (auf dieselben oder separate Peers) anwenden möchten. Wenn beispielsweise der Datenverkehr zwischen einem Satz von Subnetzen authentifiziert werden soll und der Datenverkehr zwischen anderen Subnetzen sowohl authentifiziert als auch verschlüsselt werden soll. Definieren Sie in diesem Fall die verschiedenen Datenverkehrstypen in zwei separaten Zugriffslisten, und erstellen Sie für jede Crypto Map-Liste einen separaten Crypto Map-Eintrag.

Clear Security Associations (SAs)

Verwenden Sie im privilegierten Modus des PIX die folgenden Befehle:

- **clear [crypto] ipsec sa**: Löscht die aktiven IPsec-SAs. Das Schlüsselwort *crypto* ist optional.
- **clear [crypto] isakmp sa**: Löscht die aktiven IKE-SAs. Das Schlüsselwort *crypto* ist optional.

Überprüfen

In diesem Abschnitt überprüfen Sie, ob Ihre Konfiguration ordnungsgemäß funktioniert.

Das Output Interpreter Tool (nur registrierte Kunden) (OIT) unterstützt bestimmte **show**-Befehle. Verwenden Sie das OIT, um eine Analyse der **Ausgabe des** Befehls **show anzuzeigen**.

- **show crypto isakmp sa** - Zeigt die Sicherheitszuordnungen (SAs) für Phase 1 an.
- **show crypto ipsec sa** - Zeigt SAs der Phase 2 an.
- **ping** - Dient zur Diagnose der grundlegenden Netzwerkverbindungen. Ein Ping von einem PIX zum anderen überprüft die Verbindung zwischen den beiden PIXs. Ein Ping kann auch vom Host hinter PIXsecond zum Host hinter PIXfirst ausgeführt werden, um den IPsec-Tunnel aufzurufen.
- **show local-host <IP_address>**: Zeigt die Übersetzungs- und Verbindungssteckplätze für den lokalen Host an, für den die IP-Adresse angegeben wurde.
- **show xlate detail**: Zeigt den Inhalt der Übersetzungssteckplätze an. Diese wird verwendet, um zu überprüfen, ob der Host übersetzt wurde.

PIXfirst verifizieren

Dies ist die Ausgabe des **Ping-**Befehls.

```
PIXfirst(config)#ping 10.2.1.1

!--- PIX pings the outside interface of the peer. !--- This implies that connectivity between
peers is available. 10.2.1.1 response received -- 0ms
10.2.1.1 response received -- 0ms
10.2.1.1 response received -- 0ms
PIXfirst(config)#
```

Dies ist die Ausgabe des Befehls **show crypto isakmp sa**.

```
PIXfirst(config)#show crypto isakmp sa
Total : 1
Embryonic : 0

!--- Phase 1 SA is authenticated and established. dst src state pending created 10.1.1.1
10.2.1.1     QM_IDLE   0           1
```

Dies ist die Ausgabe des Befehls **show crypto ipsec sa**.

```
!--- Shows Phase 2 SAs. PIXfirst(config)#show crypto ipsec sa
```

```
interface: outside
Crypto map tag: transam, local addr. 10.1.1.1
!--- Shows addresses of hosts that !--- communicate over this tunnel. local ident
(addr/mask/prot/port): (192.168.1.2/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (192.168.100.2/255.255.255.255/0/0)
current_peer: 10.2.1.1:500

PERMIT, flags={origin_is_acl,}
!--- Shows if traffic passes over the tunnel or not. !--- Encapsulated packets translate to
packets that are sent. !--- Decapsulated packets translate to packets that are received. #pkts
encaps: 21, #pkts encrypt: 21, #pkts digest 21
#pkts decaps: 21, #pkts decrypt: 21, #pkts verify 21
#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.: 10.1.1.1, remote crypto endpt.: 10.2.1.1
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: 6ef53756

!--- If an inbound Encapsulating Security Payload (ESP) !--- SA and outbound ESP SA exists with
a !--- security parameter index (SPI) !--- number, it implies that the Phase 2 SAs !--- are
established successfully. inbound esp sas:

     spi: 0x1cf45b9f(485776287)

          transform: esp-des esp-md5-hmac ,
          in use settings ={Tunnel, }
          slot: 0, conn id: 2, crypto map: transam
          sa timing: remaining key lifetime (k/sec): (4607998/28756)
          IV size: 8 bytes
          replay detection support: Y

inbound ah sas:

inbound pcp sas:


outbound esp sas:

   spi: 0x6ef53756(1861564246)

        transform: esp-des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 1, crypto map: transam
        sa timing: remaining key lifetime (k/sec): (4607998/28756)
        IV size: 8 bytes
        replay detection support: Y

outbound ah sas:

outbound pcp sas:
```

Dies ist die Ausgabe des Befehls **show local-host**.

```
!--- Shows translation for the host on a remote network. PIXfirst(config)#show local-host
192.168.100.2

Interface outside: 1 active, 1 maximum active, 0 denied
local host: <192.168.100.2>,
TCP connection count/limit = 0/unlimited
TCP embryonic count = 0
```

```
TCP intercept watermark = unlimited
UDP connection count/limit = 0/unlimited
AAA:
Xlate(s):
Global 192.168.50.2 Local 192.168.100.2
Conn(s):
```

Dies ist die Ausgabe des Befehls **show xlate detail**.

```
!--- Shows translation for the host on a remote network. PIXfirst(config)#show xlate detail
1 in use, 1 most used
Flags: D - DNS, d - dump, I - identity, i - inside, n - no random,
o - outside, r - portmap, s - static
NAT from outside:192.168.100.2 to inside:192.168.50.2 flags s
PIXfirst(config)#
```

PIXsecond verifizieren

Dies ist die Ausgabe des **Ping**-Befehls.

```
PIXsecond(config)#ping 10.1.1.1

!--- PIX can ping the outside interface of the peer. !--- This implies that connectivity between
peers is available. 10.1.1.1 response received -- 0ms
10.1.1.1 response received -- 0ms
10.1.1.1 response received -- 0ms
PIXsecond(config)#
```

Dies ist die Ausgabe des Befehls **show crypto isakmp sa**.

```
PIXsecond(config)#show crypto isakmp sa

Total : 1
Embryonic : 0
!--- Phase 1 SA is authenticated and established. dst src state pending created 10.1.1.1
10.2.1.1    QM_IDLE      0             1
```

Dies ist die Ausgabe des Befehls **show crypto ipsec sa**.

```
!--- Shows Phase 2 SAs. PIXsecond(config)#show crypto ipsec sa

interface: outside
Crypto map tag: transam, local addr. 10.2.1.1
!--- Shows addresses of hosts that communicate !--- over this tunnel. local ident
(addr/mask/prot/port): (192.168.100.2/255.255.255.255/0/0)
remote ident (addr/mask/prot/port): (192.168.1.2/255.255.255.255/0/0)
current_peer: 10.1.1.1:500

PERMIT, flags={origin_is_acl,}
!--- Shows if traffic passes over the tunnel or not. !--- Encapsulated packets translate to
packets that are sent. !--- Decapsulated packets translate to packets that are received. #pkts
encaps: 21, #pkts encrypt: 21, #pkts digest 21
#pkts decaps: 21, #pkts decrypt: 21, #pkts verify 21
#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.: 10.2.1.1, remote crypto endpt.: 10.1.1.1
```

```
path mtu 1500, ipsec overhead 56, media mtu 1500
current outbound spi: 1cf45b9f

!--- If an inbound ESP SA and outbound ESP SA exists with an SPI !--- number, it implies that
the Phase 2 SAs are established successfully. inbound esp sas:

   spi: 0x6ef53756(1861564246)

      transform: esp-des esp-md5-hmac ,
      in use settings ={Tunnel, }
      slot: 0, conn id: 2, crypto map: transam
      sa timing: remaining key lifetime (k/sec): (4607990/28646)
      IV size: 8 bytes
      replay detection support: Y
inbound ah sas:

inbound pcp sas:

outbound esp sas:

    spi: 0x1cf45b9f(485776287)

       transform: esp-des esp-md5-hmac ,
       in use settings ={Tunnel, }
       slot: 0, conn id: 1, crypto map: transam
       sa timing: remaining key lifetime (k/sec): (4607993/28645)
       IV size: 8 bytes
       replay detection support: Y


outbound ah sas:

outbound pcp sas:

PIXsecond(config)#
```

Fehlerbehebung

Dieser Abschnitt enthält Informationen zur Fehlerbehebung in Ihrer Konfiguration.

Befehle zur Fehlerbehebung

Das Output Interpreter Tool (nur registrierte Kunden) (OIT) unterstützt bestimmte **show**-Befehle. Verwenden Sie das OIT, um eine Analyse der **Ausgabe des** Befehls **show anzuzeigen**.

Hinweis: Beachten Sie vor der Verwendung von **Debug-**Befehlen die Informationen zu Debug-Befehlen.

- **debug crypto ipsec**: Zeigt Informationen über IPsec-Ereignisse an.
- **debug crypto isakmp**: Zeigt Meldungen über IKE-Ereignisse (Internet Key Exchange) an.
- **debugpaket if_name [src source_ip [Netzmaske] [dst dest_ip [Netzmaske]] [[proto icmp]] | [proto tcp [sport src_port] [dport dest_port] | [proto udp [sport src_port] [dport dest_port] [rx] | tx | both]**: Zeigt die Pakete an, die die angegebene Schnittstelle erreichen. Dieser Befehl ist hilfreich, wenn Sie zuerst den Typ des Datenverkehrs auf der internen Schnittstelle von PIX bestimmen. Mit diesem Befehl wird auch überprüft, ob die beabsichtigte Übersetzung vorhanden ist.
- **logging buffered level** - Sendet Syslog-Meldungen an einen internen Puffer, der mit dem

Befehl **show logging** angezeigt wird. Verwenden Sie den Befehl **clear logging**, um den Nachrichtenpuffer zu löschen. Neue Nachrichten werden am Ende des Puffers angefügt. Mit diesem Befehl wird die erstellte Übersetzung angezeigt. Die Anmeldung am Puffer muss bei Bedarf aktiviert werden. Deaktivieren Sie die Protokollierung in einen Puffer, ohne dass **die Protokollierung** und/oder **keine Anmeldung erfolgt**.

- **debug icmp trace** - Zeigt die Internet Control Message Protocol (ICMP)-Paketinformationen, die Quell-IP-Adresse und die Zieladresse der Pakete an, die die PIX-Firewall eintreffen, von ihr abweichen und die PIX-Firewall durchlaufen. Dazu gehören auch Pings zu den eigenen Schnittstellen der PIX Firewall-Einheit. Verwenden Sie **keinen debug icmp trace**, um die **Debugging-ICMP-Ablaufverfolgung** zu deaktivieren.

Dies ist die Ausgabe der **debug crypto isakmp** und der **debug crypto ipsec**-Befehle.

```
PIXfirst(config)#debug crypto isakmp
PIXfirst(config)#debug crypto ipsec
PIXfirst(config)#debug crypto engine
PIXfirst(config)#show debug
debug crypto ipsec 1
debug crypto isakmp 1
debug crypto engine
PIXfirst(config)#

PIXfirst(config)#

crypto_isakmp_process_block:src:10.2.1.1, dest:10.1.1.1 spt:500 dpt:500
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_IDLE
ISAKMP (0): processing SA payload. message ID = 137660894

ISAKMP : Checking IPSec proposal 1

ISAKMP: transform 1, ESP_DES
ISAKMP: attributes in transform:
ISAKMP: encaps is 1
ISAKMP: SA life type in seconds
ISAKMP: SA life duration (basic) of 28800
ISAKMP: SA life type in kilobytes
ISAKMP: SA life duration (VPI) of 0x0 0x46 0x50 0x0
ISAKMP: authenticator is HMAC-MD5

!--- Phase 1 policy accepted. ISAKMP (0): atts are acceptable. IPSEC(validate_proposal_request):
proposal part #1,
(key eng. msg.) dest= 10.1.1.1, src= 10.2.1.1,
!--- Encryption domain (interesting traffic) that invokes the tunnel. dest_proxy=
192.168.1.2/255.255.255.255/0/0 (type=1),
src_proxy= 192.168.100.2/255.255.255.255/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 0s and 0kb,
spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4

ISAKMP (0): processing NONCE payload. message ID = 137660894
ISAKMP (0): processing ID payload. message ID = 137660894
ISAKMP (0): ID_IPV4_ADDR src 192.168.100.2 prot 0 port 0
ISAKMP (0): processing ID payload. message ID = 137660894
ISAKMP (0): ID_IPV4_ADDR dst 192.168.1.2 prot 0 port 0IPSEC(key_engine):
got a queue event...
IPSEC(spi_response): getting spi 0x15ee92d9(367956697) for SA
from 10.2.1.1 to 10.1.1.1 for prot 3
```

```
return status is IKMP_NO_ERROR
crypto_isakmp_process_block:src:10.2.1.1, dest:10.1.1.1 spt:500 dpt:500
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_AUTH_AWAITmap_alloc_entry: allocating entry 2
map_alloc_entry: allocating entry 1

ISAKMP (0): Creating IPSec SAs
inbound SA from 10.2.1.1 to 10.1.1.1 (proxy 192.168.100.2 to 192.168.1.2)
has spi 367956697 and conn_id 2 and flags 4
lifetime of 28800 seconds
lifetime of 4608000 kilobytes
outbound SA from 10.1.1.1 to 10.2.1.1 (proxy 192.168.1.2 to 192.168.100.2)
has spi 1056204195 and conn_id 1 and flags 4
lifetime of 28800 seconds
lifetime of 4608000 kilobytesIPSEC(key_engine): got a queue event...
IPSEC(initialize_sas): ,
(key eng. msg.) dest= 10.1.1.1, src= 10.2.1.1,
dest_proxy= 192.168.1.2/0.0.0.0/0/0 (type=1),
src_proxy= 192.168.100.2/0.0.0.0/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 28800s and 4608000kb,
spi= 0x15ee92d9(367956697), conn_id= 2, keysize= 0, flags= 0x4
IPSEC(initialize_sas): ,
(key eng. msg.) src= 10.1.1.1, dest= 10.2.1.1,
src_proxy= 192.168.1.2/0.0.0.0/0/0 (type=1),
dest_proxy= 192.168.100.2/0.0.0.0/0/0 (type=1),
protocol= ESP, transform= esp-des esp-md5-hmac ,
lifedur= 28800s and 4608000kb,
spi= 0x3ef465a3(1056204195), conn_id= 1, keysize= 0, flags= 0x4

VPN Peer: IPSEC: Peer ip:10.2.1.1/500 Ref cnt incremented to:2 Total VPN Peers:1
VPN Peer: IPSEC: Peer ip:10.2.1.1/500 Ref cnt incremented to:3 Total VPN Peers:1
return status is IKMP_NO_ERROR

PIXfirst(config)#
```

Dies ist die Ausgabe des **Debugpakets im src-**Befehl.

```
!--- Shows that the remote host packet is translated. PIXfirst(config)#debug packet inside src
192.168.50.2 dst 192.168.1.2
PIXfirst(config)# show debug
debug packet inside src 192.168.50.2 dst 192.168.1.2 both

--------- PACKET ---------


-- IP --

!--- Source IP is translated to 192.168.50.2. 192.168.50.2 ==> 192.168.1.2


ver = 0x4 hlen = 0x5 tos = 0x0 tlen = 0x3c

id = 0x82 flags = 0x0 frag off=0x0

ttl = 0x80 proto=0x1 chksum = 0x85ea

!--- ICMP echo packet, as expected. -- ICMP --

type = 0x8 code = 0x0 checksum=0x425c
```

```
identifier = 0x200 seq = 0x900

-- DATA --

0000001c: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 | abcdefghijklmnop

0000002c: 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 | qrstuvwabcdefghi

0000003c: 01 | .

--------- END OF PACKET ---------

--------- PACKET ---------

-- IP --

192.168.50.2 ==> 192.168.1.2

ver = 0x4 hlen = 0x5 tos = 0x0 tlen = 0x3c

id = 0x83 flags = 0x0 frag off=0x0

ttl = 0x80 proto=0x1 chksum = 0x85e9

-- ICMP --

type = 0x8 code = 0x0 checksum=0x415c

identifier = 0x200 seq = 0xa00

-- DATA --

0000001c: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 | abcdefghijklmnop

0000002c: 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 | qrstuvwabcdefghi

0000003c: 01 | .

--------- END OF PACKET ---------

--------- PACKET ---------

-- IP --

192.168.50.2 ==> 192.168.1.2

ver = 0x4 hlen = 0x5 tos = 0x0 tlen = 0x3c

id = 0x84 flags = 0x0 frag off=0x0

ttl = 0x80 proto=0x1 chksum = 0x85e8

-- ICMP --
```

```
type = 0x8 code = 0x0 checksum=0x405c

identifier = 0x200 seq = 0xb00

-- DATA --

0000001c: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 | abcdefghijklmnop

0000002c: 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 | qrstuvwabcdefghi

0000003c: 01 | .


--------- END OF PACKET ---------


--------- PACKET ---------


-- IP --

192.168.50.2 ==> 192.168.1.2


ver = 0x4 hlen = 0x5 tos = 0x0 tlen = 0x3c

id = 0x85 flags = 0x0 frag off=0x0

ttl = 0x80 proto=0x1 chksum = 0x85e7


-- ICMP --

type = 0x8 code = 0x0 checksum=0x3f5c

identifier = 0x200 seq = 0xc00

-- DATA --

0000001c: 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 | abcdefghijklmnop

0000002c: 71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69 | qrstuvwabcdefghi

0000003c: 01 | .


--------- END OF PACKET ---------

PIXfirst(config)#
```

Dies ist die Ausgabe des Befehls **Protokollierungspuffer**.

```
!--- Logs show translation is built. PIXfirst(config)#logging buffer 7
PIXfirst(config)#logging on
PIXfirst(config)#show logging

Syslog logging: enabled
Facility: 20
Timestamp logging: disabled
Standby logging: disabled
Console logging: disabled
Monitor logging: disabled
```

```
Buffer logging: level debugging, 53 messages logged
Trap logging: disabled
History logging: disabled
Device ID: disabled

111009: User 'enable_15' executed cmd: show logging
602301: sa created, (sa) sa_dest= 10.1.1.1, sa_prot= 50,
sa_spi= 0xb1274c19(2972142617), sa_trans= esp-des esp-md5-hmac , sa_conn_id= 2
602301: sa created, (sa) sa_dest= 10.2.1.1, sa_prot= 50,
sa_spi= 0x892de1df(2301485535), sa_trans= esp-des esp-md5-hmac , sa_conn_id= 1
!--- Translation is built. 609001: Built local-host outside:192.168.100.2
305009: Built static translation from outside:192.168.100.2 to inside:192.168.50.2
PIXfirst(config)#
```

Dies ist die Ausgabe des Befehls **debug icmp trace**.

```
!--- Shows ICMP echo and echo-reply with translations !--- that take place.
PIXfirst(config)#debug icmp trace

ICMP trace on

Warning: this may cause problems on busy networks


PIXfirst(config)# 5: ICMP echo-request from outside:192.168.100.2 to 192.168.1.2
ID=1024 seq=1280 length=40
6: ICMP echo-request: translating outside:192.168.100.2 to inside:192.168.50.2
7: ICMP echo-reply from inside:192.168.1.2 to 192.168.50.2 ID=1024 seq=1280 length=40
8: ICMP echo-reply: untranslating inside:192.168.50.2 to outside:192.168.100.2
9: ICMP echo-request from outside:192.168.100.2 to 192.168.1.2 ID=1024 seq=1536 length=40
10: ICMP echo-request: translating outside:192.168.100.2 to inside:192.168.50.2
11: ICMP echo-reply from inside:192.168.1.2 to 192.168.50.2 ID=1024 seq=1536 length=40
12: ICMP echo-reply: untranslating inside:192.168.50.2 to outside:192.168.100.2
13: ICMP echo-request from outside:192.168.100.2 to 192.168.1.2 ID=1024 seq=1792 length=40
14: ICMP echo-request: translating outside:192.168.100.2 to inside:192.168.50.2
15: ICMP echo-reply from inside:192.168.1.2 to 192.168.50.2 ID=1024 seq=1792 length=40
16: ICMP echo-reply: untranslating inside:192.168.50.2 to outside:192.168.100.2
17: ICMP echo-request from outside:192.168.100.2 to 192.168.1.2 ID=1024 seq=2048 length=40
18: ICMP echo-request: translating outside:192.168.100.2 to inside:192.168.50.2
19: ICMP echo-reply from inside:192.168.1.2 to 192.168.50.2 ID=1024 seq=2048 length=40
20: ICMP echo-reply: untranslating inside:192.168.50.2 to outside:192.168.100.2


PIXfirst(config)#
```

Zugehörige Informationen

- Support für Security Appliances der Serie PIX 500
- PIX-Befehlsreferenzen
- Anforderungen für Kommentare (RFCs)
- Support-Seite für IPsec-Aushandlung/IKE-Protokolle
- Technischer Support und Dokumentation - Cisco Systems