

IPSec - PIX zu Cisco VPN Client Wild Card, vorinstallierte Modus-Konfiguration mit erweiterter Authentifizierung

Inhalt

[Einführung](#)

[Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

[Konventionen](#)

[Hintergrundinformationen](#)

[Konfigurieren](#)

[Netzwerkdiagramm](#)

[Konfigurationen](#)

[Überprüfen](#)

[Fehlerbehebung](#)

[Befehle zur Fehlerbehebung](#)

[Beispiel für das PIX-Debuggen](#)

[Debugger mit VPN Client 4.x](#)

[Debugger mit VPN Client 1.1](#)

[Zugehörige Informationen](#)

[Einführung](#)

In diesem Konfigurationsbeispiel wird veranschaulicht, wie ein VPN-Client mithilfe von Platzhaltern, mode-config, dem Befehl **sysopt connection permit-ipsec** und erweiterter Authentifizierung (Xauth) mit einer PIX-Firewall verbunden wird.

Informationen zur TACACS+- und RADIUS-Konfiguration für PIX 6.3 und höher finden Sie unter [TACACS+ und RADIUS für PIX 6.3 und PIX/ASA 7.x Konfigurationsbeispiel](#).

Der VPN-Client unterstützt Advanced Encryption Standard (AES) als Verschlüsselungsalgorithmus in Version 3.6.1 und höher des Cisco VPN-Clients und mit PIX Firewall 6.3. Der VPN-Client unterstützt nur Schlüsselgrößen von 128 Bit und 256 Bit. Weitere Informationen zur Konfiguration von AES finden Sie unter [Konfigurieren des Cisco VPN-Clients für PIX mit AES](#).

Weitere Informationen finden Sie unter [PIX/ASA 7.x und Cisco VPN Client 4.x für Windows mit Microsoft Windows 2003 IAS RADIUS Authentication Configuration Example](#) zum Einrichten der Remotezugriff-VPN-Verbindung zwischen einem Cisco VPN Client (4.x für Windows) und der PIX 500 Security Appliance 7.x unter Verwendung eines Microsoft Windows 2003 Internet

Authentication Service (IAS) RADIUS-Servers.

Weitere Informationen [zu IPsec Between a VPN 300 Concentrator and a VPN Client 4.x for Windows using RADIUS for User Authentication and Accounting Configuration Example](#) zum Einrichten eines IPsec-Tunnels zwischen einem Cisco VPN 300 Concentrator und einem Cisco VPN Client 4.x für Windows unter Verwendung von RADIUS für die Benutzerauthentifizierung und -abrechnung.

Unter [Konfigurieren von IPsec zwischen einem Cisco IOS-Router und einem Cisco VPN-Client 4.x für Windows mithilfe von RADIUS für die Benutzerauthentifizierung](#) konfigurieren Sie eine Verbindung zwischen einem Router und dem Cisco VPN Client 4.x mithilfe von RADIUS für die Benutzerauthentifizierung.

Voraussetzungen

Anforderungen

Für dieses Dokument bestehen keine speziellen Anforderungen.

Verwendete Komponenten

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

- Cisco VPN Client 4.x Im Gegensatz zum Cisco Secure VPN Client 1.x bietet dieses Produkt erweiterte VPN-Funktionen.
- PIX Firewall 515E Version 6.3(3).

Hinweis: Verschlüsselungstechnologie unterliegt Exportkontrollen. Es liegt in Ihrer Verantwortung, das Gesetz über den Export von Verschlüsselungstechnologien zu kennen. Weitere Informationen finden Sie auf der [Website des Bureau of Export Administration](#). Wenn Sie Fragen zur Exportkontrolle haben, senden Sie eine E-Mail an export@cisco.com.

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 unter [Cisco Technical Tips Conventions](#) (Technische Tipps zu Konventionen von Cisco).

Hintergrundinformationen

Der Befehl **sysopt connection permit-ipsec** erlaubt implizit jedem Paket, das von einem IPsec-Tunnel kommt, die Überprüfung eines zugeordneten **Zugriffslisten-**, **Kabelkanal-** oder **Zugriffsgruppen-**Befehls für IPsec-Verbindungen. Xauth authentifiziert den IPsec-Benutzer an einem externen TACACS+- oder RADIUS-Server. Zusätzlich zum vorinstallierten Schlüssel einer Wildcard muss der Benutzer einen Benutzernamen/ein Kennwort eingeben.

Ein Benutzer mit einem VPN-Client erhält eine IP-Adresse von seinem ISP. Diese wird durch eine IP-Adresse aus dem IP-Adresspool auf dem PIX ersetzt. Der Benutzer hat Zugriff auf alle Komponenten innerhalb der Firewall, einschließlich Netzwerke. Benutzer, die den VPN-Client nicht ausführen, können sich nur über die von der statischen Zuweisung angegebene externe Adresse mit dem Webserver verbinden.

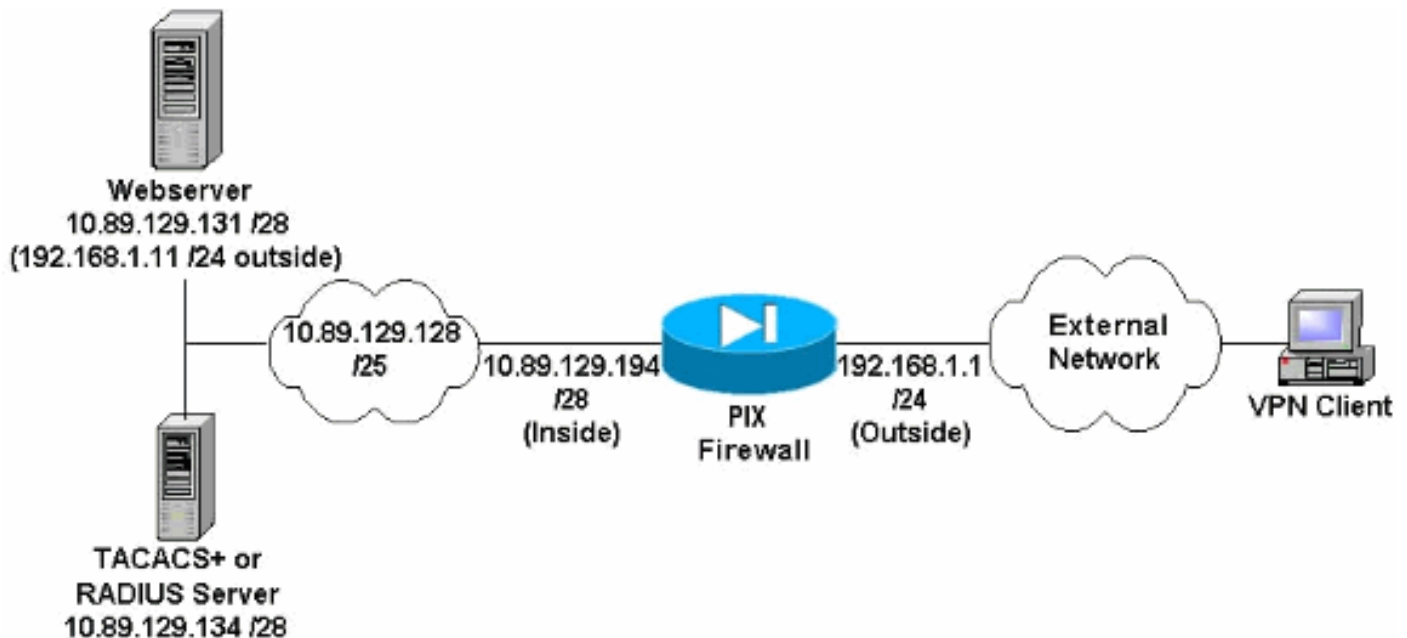
Konfigurieren

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 Dokument verwendeten Befehlen zu erhalten.

Netzwerkdiagramm

In diesem Dokument wird die folgende Netzwerkeinrichtung verwendet:



Hinweise zu Netzwerkdiagrammen

- Internethosts, die über die globale IP-Adresse 192.168.1.1 auf den Webserver zugreifen, werden selbst dann authentifiziert, wenn keine VPN-Verbindung hergestellt wurde. Dieser Datenverkehr wird *nicht* verschlüsselt.
- VPN-Clients können auf alle Hosts im internen Netzwerk (10.89.129.128/25) zugreifen, sobald der IPsec-Tunnel eingerichtet ist. Der gesamte Datenverkehr vom VPN-Client zur PIX-Firewall wird verschlüsselt. Ohne einen IPsec-Tunnel können sie nur über ihre globale IP-Adresse auf den Webserver zugreifen, müssen sich aber trotzdem authentifizieren.
- VPN-Clients stammen aus dem Internet, und ihre IP-Adressen sind nicht im Voraus bekannt.

Konfigurationen

In diesem Dokument werden diese Konfigurationen verwendet.

- [PIX-Konfiguration 6.3\(3\)](#)
- [Konfiguration des VPN-Clients 4.0.5](#)
- [Konfiguration des VPN-Clients 3.5](#)
- [Konfiguration des VPN-Clients 1.1](#)

PIX-Konfiguration 6.3(3)

```

pixfirewall#show run
: Saved
:
PIX Version 6.3(3)
interface ethernet0 100full
interface ethernet1 100full
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
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
!--- Do not use Network Address Translation (NAT) for
inside-to-pool !--- traffic. This should not go through
NAT. access-list 101 permit ip 10.89.129.128
255.255.255.240 10.89.129.192 255.255.255.240 !---
Permits Internet Control Message Protocol (ICMP) !---
Transmission Control Protocol (TCP) and User Datagram
Protocol (UDP) !--- traffic from any host on the
Internet (non-VPN) to the web server. access-list 120
permit icmp any host 10.89.129.131 access-list 120
permit tcp any host 10.89.129.131 access-list 120 permit
udp any host 10.89.129.131 pager lines 24 mtu outside
1500 mtu inside 1500 ip address outside 192.168.1.1
255.255.255.0 ip address inside 10.89.129.194
255.255.255.240 ip audit info action alarm ip audit
attack action alarm !--- Specifies the inside IP address
range to be assigned !--- to the VPN Clients. ip local
pool VPNpool 10.89.129.200-10.89.129.204 no failover
failover timeout 0:00:00 failover poll 15 no failover ip
address outside no failover ip address inside pdm
history enable arp timeout 14400 !--- Defines a pool of
global addresses to be used by NAT. global (outside) 1
192.168.1.6-192.168.1.10 nat (inside) 0 access-list 101
nat (inside) 1 0.0.0.0 0.0.0.0 0 0 !--- Specifies which
outside IP address to apply to the web server. static
(inside,outside) 192.168.1.11 10.89.129.131 netmask
255.255.255.255 0 0 !--- Apply ACL 120 to the outside
interface in the inbound direction. access-group 120 in
interface outside !--- Defines a default route for the
PIX. route outside 0.0.0.0 0.0.0.0 192.168.1.3 1 !---

```

```

Defines a route for traffic within the PIX's !--- subnet to reach other inside hosts.
route inside 10.89.129.128 255.255.255.128 10.89.129.193 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
!--- Authentication, authorization, and accounting (AAA) statements !--- for authentication. !--- Use either of these statements to define the protocol of the group AuthInbound. !--- You cannot use both.
aaa-server AuthInbound protocol tacacs+

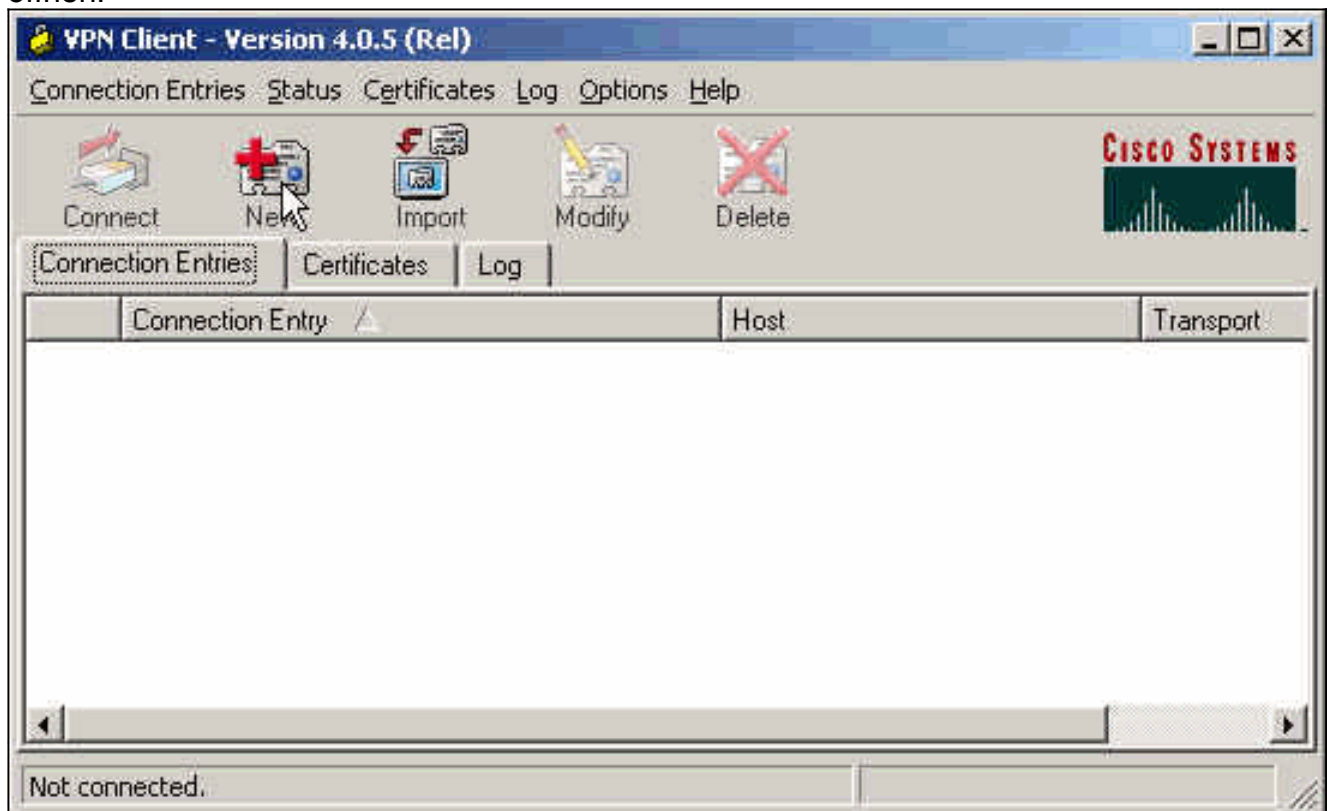
!--- OR aaa-server AuthInbound protocol radius !---
After you define the protocol of the group AuthInbound, define !--- a server of the same type. !--- In this case we specify the TACACS+ server and key of "secretkey".
aaa-server AuthInbound (inside) host 10.89.129.134 secretkey timeout 10 !--- Authenticate HTTP, FTP, and Telnet traffic to the web server.
aaa authentication include http outside 10.89.129.131 255.255.255.255 0.0.0.0 0.0.0.0 AuthInbound aaa authentication include ftp outside 10.89.129.131 255.255.255.255 0.0.0.0 0.0.0.0 AuthInbound aaa authentication include telnet outside 10.89.129.131 255.255.255.255 0.0.0.0 0.0.0.0 AuthInbound
no snmp-server location no snmp-server contact snmp-server community public no snmp-server enable traps floodguard enable !--- Trust IPsec traffic and avoid going through ACLs/NAT.
sysopt connection permit-ipsec !--- IPsec and dynamic map configuration.
crypto ipsec transform-set myset esp-des esp-md5-hmac
crypto dynamic-map dynmap 10 set transform-set myset
crypto map mymap 10 ipsec-isakmp dynamic dynmap !--- Assign IP address for VPN 1.1 Clients.
crypto map mymap client configuration address initiate crypto map mymap client configuration address respond !--- Use the AAA server for authentication (AuthInbound).
crypto map mymap client authentication AuthInbound !--- Apply the IPsec/AAA/ISAKMP configuration to the outside interface.
crypto map mymap interface outside isakmp enable outside !--- Pre-shared key for VPN 1.1 Clients.
isakmp key ***** address 0.0.0.0 netmask 0.0.0.0 isakmp identity address !--- Assign address from "VPNpool" pool for VPN 1.1 Clients.
isakmp client configuration address-pool local VPNpool outside !--- ISAKMP configuration for VPN Client 3.x/4.x.
isakmp policy 10 authentication pre-share isakmp policy 10 encryption des isakmp policy 10 hash md5 isakmp policy 10 group 2 isakmp policy 10 lifetime 86400 !--- ISAKMP configuration for VPN Client 1.x.
isakmp policy 20 authentication pre-share isakmp policy 20 encryption des isakmp policy 20 hash md5 isakmp policy 20 group 1 isakmp policy 20 lifetime 86400 !--- Assign addresses from "VPNpool" for VPN Client 3.x/4.x.
vpngroup vpn3000 address-pool VPNpool vpngroup vpn3000 idle-time 1800 !--- Group password for VPN Client 3.x/4.x (not shown in configuration).
vpngroup vpn3000 password ***** telnet timeout 5 ssh timeout 5 console timeout 0 terminal width 80
Cryptochecksum:ba54c063d94989cbd79076955dbfeefc : end
pixfirewall#

```

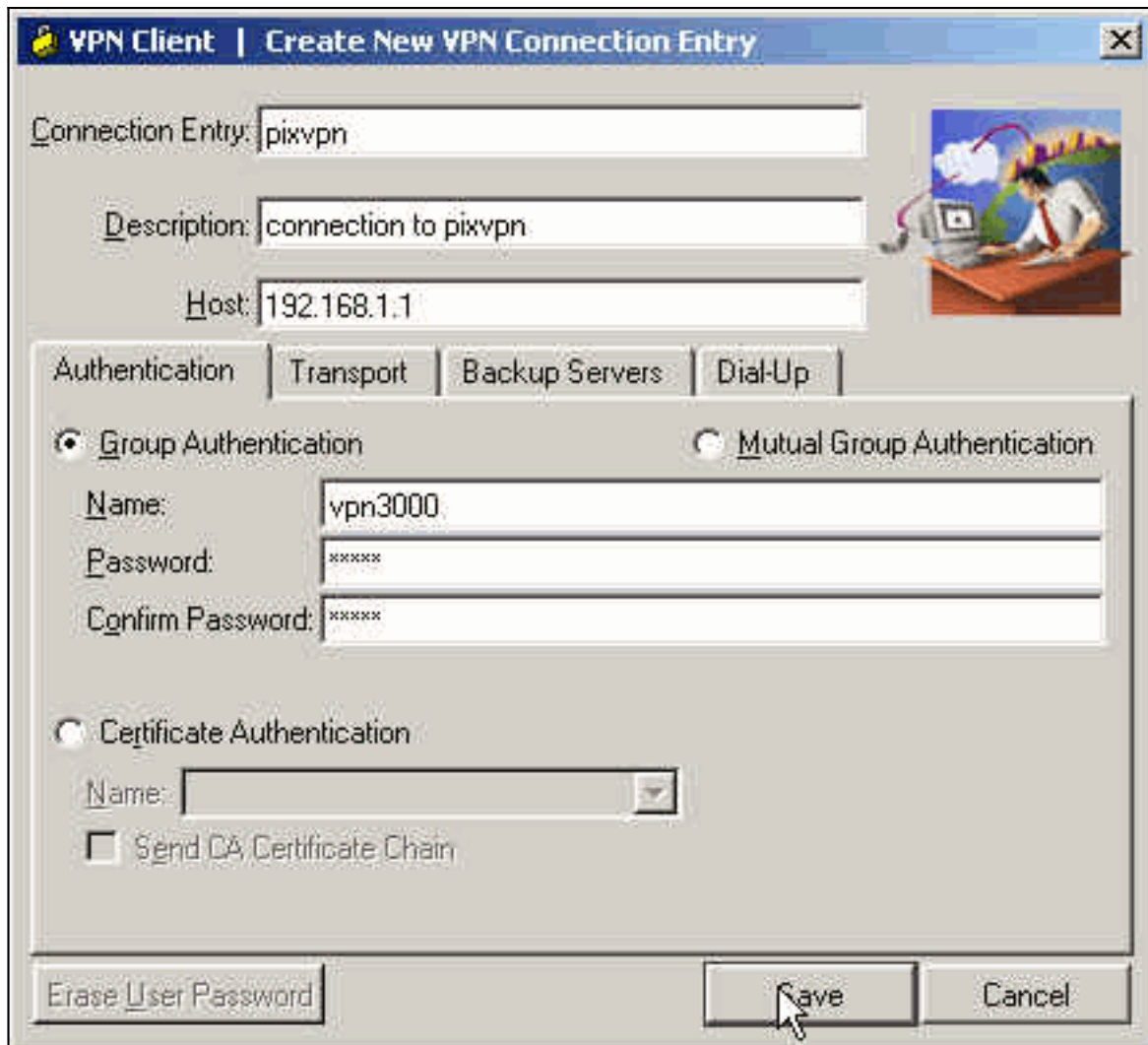
Konfiguration des VPN-Clients 4.0.5

Führen Sie diese Schritte aus, um den VPN-Client 4.0.5 zu konfigurieren.

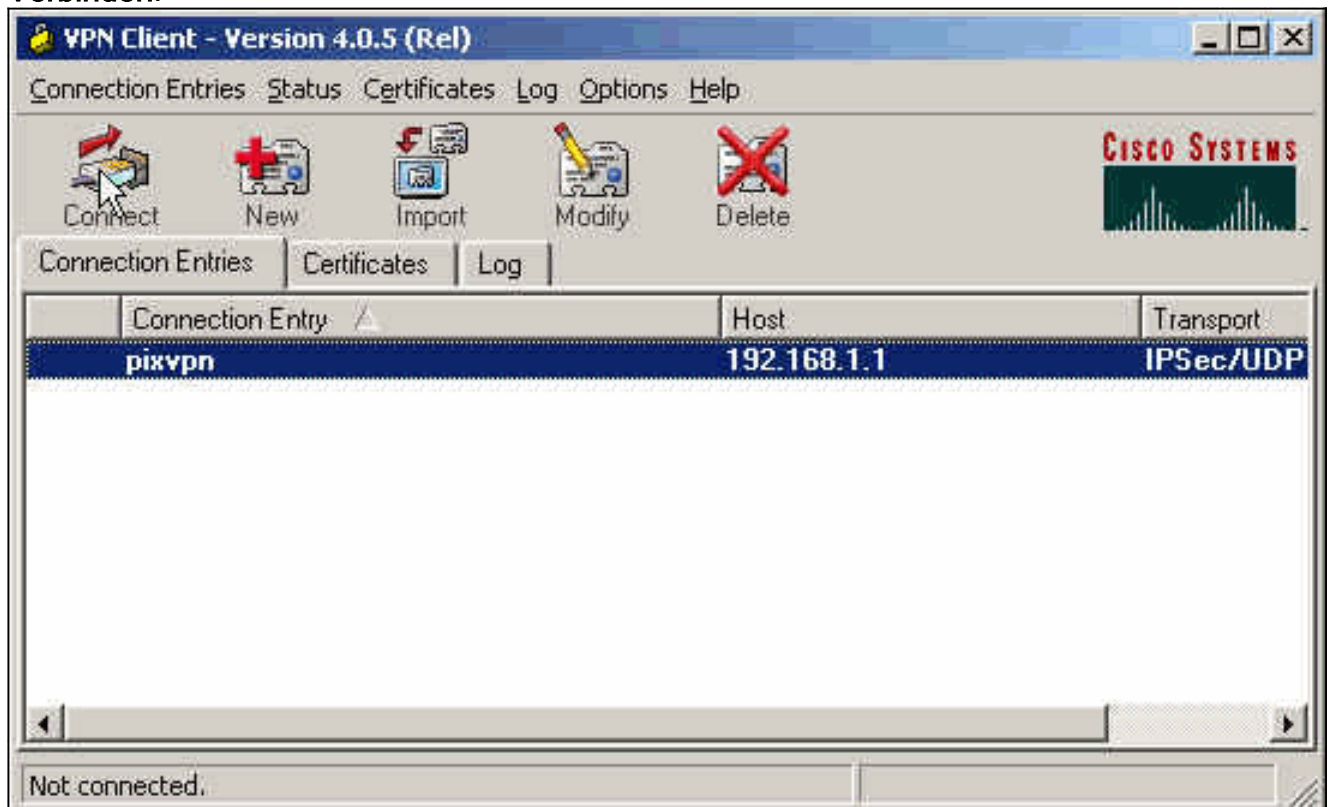
1. Wählen Sie **Start > Programme > Cisco Systems VPN Client > VPN Client** aus.
2. Klicken Sie auf **Neu**, um das Fenster Create New VPN Connection Entry (Neue VPN-Verbindung erstellen) zu öffnen.



3. Geben Sie den Namen des Verbindungseintrags und eine Beschreibung ein. Geben Sie die externe IP-Adresse der PIX-Firewall in das Host-Feld ein. Geben Sie dann den Namen und das Kennwort der VPN-Gruppe ein, und klicken Sie auf **Speichern**.

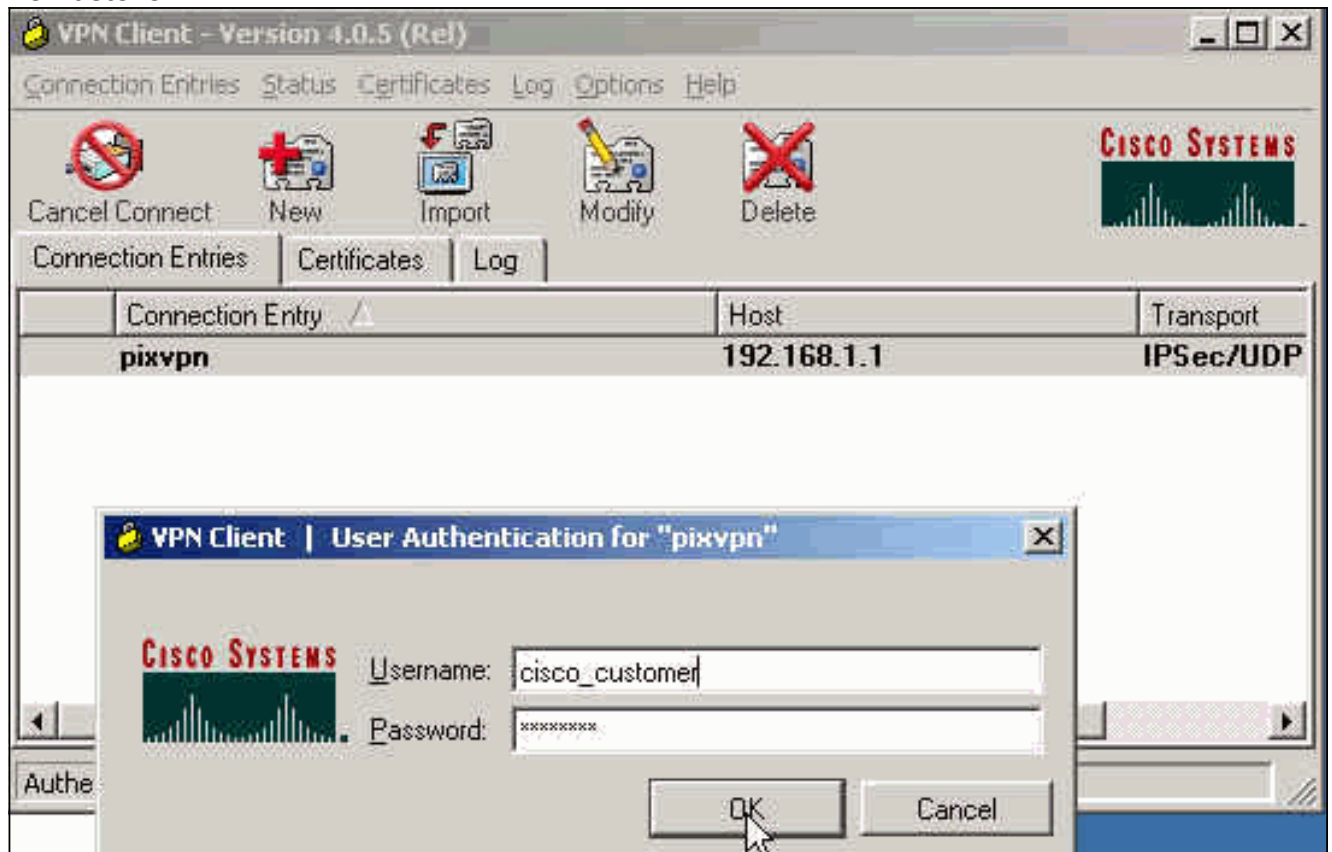


4. Klicken Sie im Hauptfenster des VPN-Clients auf die Verbindung, die Sie verwenden möchten, und klicken Sie auf die Schaltfläche **Verbinden**.



5. Geben Sie bei Aufforderung die Informationen zu Benutzernamen und Kennwort für Xauth ein,

und klicken Sie auf **OK**, um eine Verbindung zum Remote-Netzwerk herzustellen.



[Konfiguration des VPN-Clients 3.5](#)

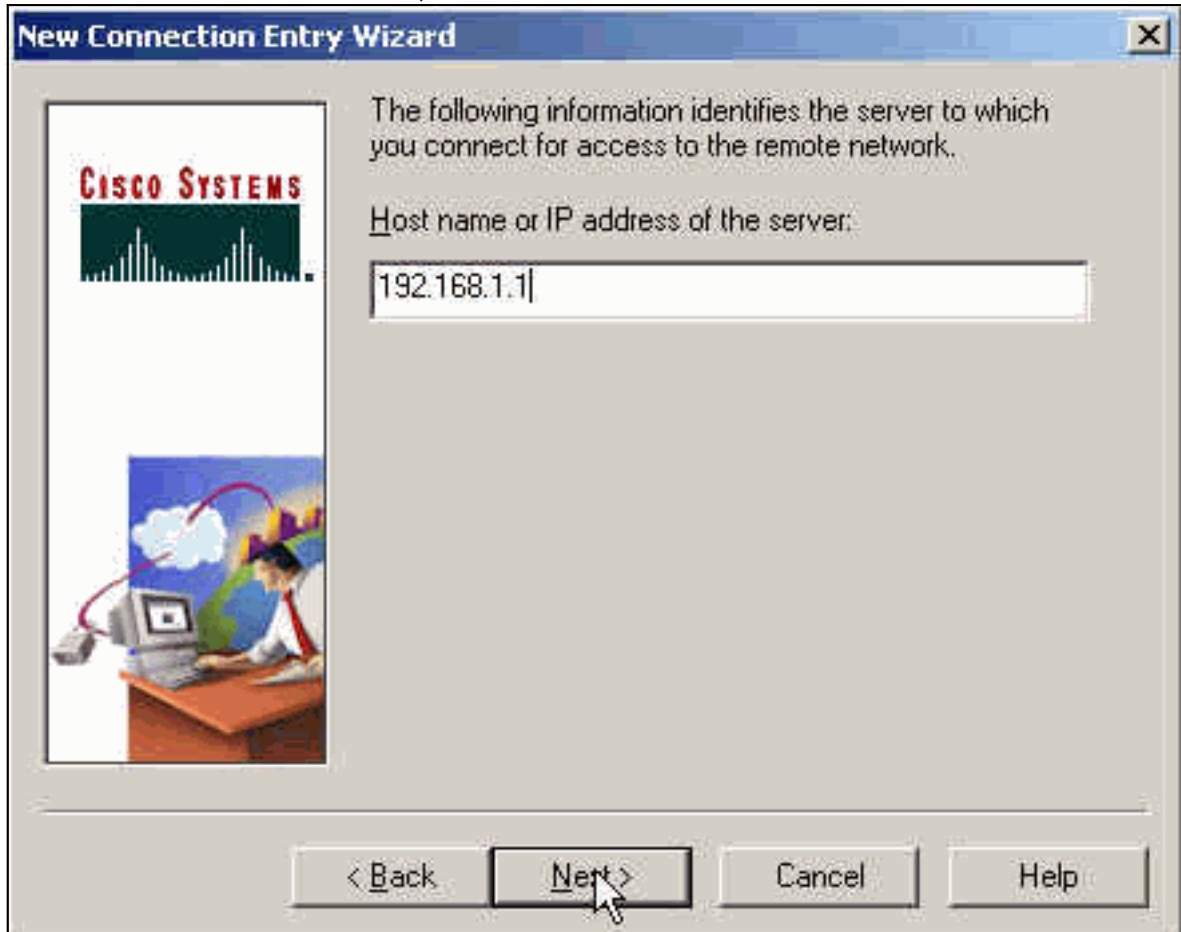
Führen Sie diese Schritte aus, um die Konfiguration von VPN Client 3.5 zu konfigurieren.

1. Wählen Sie **Start > Programme > Cisco Systems VPN Client > VPN Dialer** aus.
2. Klicken Sie auf **Neu**, um den Assistenten für neue Verbindungseinträge zu starten.
3. Geben Sie den Namen Ihres neuen Verbindungsprotokolls ein, und klicken Sie auf



Weiter.

4. Geben Sie den Hostnamen oder die IP-Adresse des Servers ein, der für die Verbindung mit dem Remote-Server verwendet wird, und klicken Sie auf



Weiter.

5. Wählen Sie **Gruppenzugriffsinformationen** aus, und geben Sie den Namen und das Kennwort

ein, mit dem der Zugriff auf den Remote-Server authentifiziert wird. Klicken Sie auf

New Connection Entry Wizard

Your administrator may have provided you with group parameters or a digital certificate to authenticate your access to the remote server. If so, select the appropriate authentication method and complete your entries .

Group Access Information

Name:

Password:

Confirm Password:

Certificate

Name:

< Back Next > Cancel Help

Weiter.

6. Klicken Sie auf **Fertig stellen**, um den neuen Eintrag zu speichern.

New Connection Entry Wizard

You have successfully created a new virtual private networking connection entry named:

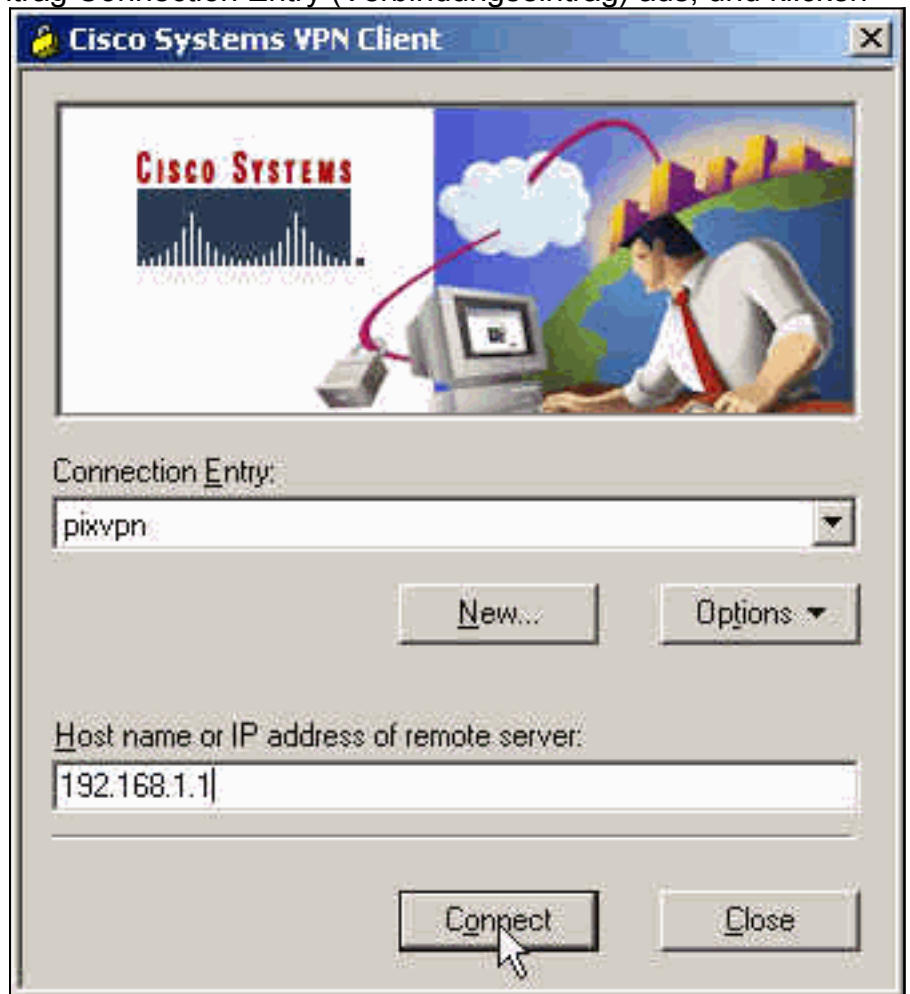
Click Finish to save this entry.

To connect to the remote network, select the Connect button from the main window.

To modify this connection entry, click Options on the main window and select Properties from the menu that appears.

< Back Finish Cancel Help

7. Wählen Sie im Dialer den Eintrag Connection Entry (Verbindungseintrag) aus, und klicken



Sie auf **Connect (Verbinden)**.

8. Geben Sie bei Aufforderung die Informationen zu Benutzernamen und Kennwort für Xauth ein, und klicken Sie auf **OK**, um eine Verbindung zum Remote-Netzwerk



herzustellen.

Konfiguration des VPN-Clients 1.1

```
Network Security policy:
1- TACconn
  My Identity
    Connection security: Secure
    Remote Party Identity and addressing
    ID Type: IP subnet
    10.89.129.128
    255.255.255.128
    Port all Protocol all

  Connect using secure tunnel

    ID Type: IP address
    192.168.1.1

  Pre-shared Key=cisco1234

  Authentication (Phase 1)

  Proposal 1
    Authentication method: pre-shared key
    Encryp Alg: DES
    Hash Alg: MD5
    SA life: Unspecified
    Key Group: DH 1
```

```
Key exchange (Phase 2)

Proposal 1
  Encapsulation ESP
  Encrypt Alg: DES
  Hash Alg: MD5
  Encap: tunnel
  SA life: Unspecified
  no AH

2- Other Connections
  Connection security: Non-secure
  Local Network Interface
    Name: Any
    IP Addr: Any
    Port: All
```

Accounting hinzufügen

Die Syntax des Befehls zum Hinzufügen von Accounting lautet:

```
aaa accounting include acctg_service inbound|outbound l_ip l_mask [f_ip f_mask] server_tag
```

In der PIX-Konfiguration wird dieser Befehl beispielsweise hinzugefügt:

```
aaa accounting include any inbound
0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound
```

Hinweis: Der Befehl **sysopt connection permit-ipsec**, nicht der **sysopt ipsec pl-kompatible** Befehl, ist erforderlich, damit Xauth Accounting funktioniert. Xauth Accounting funktioniert nicht nur mit dem **sysopt ipsec pl-kompatiblen** Befehl. Xauth Accounting gilt für TCP-Verbindungen, nicht für ICMP oder UDP.

Diese Ausgabe ist ein Beispiel für TACACS+-Accounting-Datensätze:

```
07/27/2004 15:17:54 cisco_customer Default Group 10.89.129.200 stop 15 .. 99 1879 .. ..
0x5 .. PIX 10.89.129.194 telnet
07/27/2004 15:17:39 cisco_customer Default Group 10.89.129.200 start .. .. .. .. ..
0x5 .. PIX 10.89.129.194 telnet
```

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.

Hinweis: Beachten Sie [vor der](#) Verwendung von **Debug**-Befehlen die [Informationen](#) zu [Debug-Befehlen](#).

Aktivieren Sie den Cisco Secure Log Viewer, um die clientseitigen Debugging anzuzeigen.

- **debug crypto ipsec** - Wird verwendet, um die IPsec-Verhandlungen von Phase 2 anzuzeigen.
- **debug crypto isakmp** - Wird verwendet, um die ISAKMP-Verhandlungen von Phase 1 anzuzeigen.

Fehlerbehebung

Dieser Abschnitt enthält Informationen zur Fehlerbehebung in Ihrer Konfiguration. Ein Beispiel für eine Debugausgabe wird ebenfalls angezeigt.

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 engine** - Dient zum Debuggen des Crypto Engine-Prozesses.

Beispiel für das PIX-Debuggen

```
pixfirewall#show debug
debug crypto ipsec 1
debug crypto isakmp 1
debug crypto engine
debug fover status
  tx      Off
  rx      Off
  open    Off
  cable   Off
  txdmp   Off
  rxdmp   Off
  ifc     Off
  rxip    Off
  txip    Off
  get     Off
  put     Off
  verify  Off
  switch  Off
  fail    Off
  fmsg    Off
```

Debugger mit VPN Client 4.x

```
pixfirewall#
crypto_isakmp_process_block: src 192.168.1.2, dest 192.168.1.1
VPN Peer: ISAKMP: Added new peer: ip:192.168.1.2
Total VPN Peers:1
VPN Peer: ISAKMP: Peer ip:192.168.1.2 Ref cnt incremented
to:1 Total VPN Peers:1
OAK_AG exchange
ISAKMP (0): processing SA payload. message ID = 0

ISAKMP (0): Checking ISAKMP transform 1 against priority 10 policy
```

ISAKMP: encryption 3DES-CBC
ISAKMP: hash SHA
ISAKMP: default group 2
ISAKMP: extended auth pre-share
ISAKMP: life type in seconds
ISAKMP: life duration (VPI) of 0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 2 against priority 10 policy
ISAKMP: encryption 3DES-CBC
ISAKMP: hash MD5
ISAKMP: default group 2
ISAKMP: extended auth pre-share
ISAKMP: life type in seconds
ISAKMP: life duration (VPI) of 0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 3 against priority 10 policy
ISAKMP: encryption 3DES-CBC
ISAKMP: hash SHA
ISAKMP: default group 2
ISAKMP: auth pre-shared
ISAKMP: life type in seconds
ISAKMP: life duration (VPI) of 0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 4 against priority 10 policy
ISAKMP: encryption 3DES-CBC
ISAKMP: hash MD5
ISAKMP: default group 2
ISAKMP: auth pre-share
ISAKMP: life type in seconds
ISAKMP: life duration (VPI) of 0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 5 against priority 10 policy
ISAKMP: encryption DES-CBC
ISAKMP: hash SHA
ISAKMP: default group 2
ISAKMP: extended auth pre-share
ISAKMP: life type in seconds
ISAKMP: life duration (VPI) of 0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 6 against priority 10 policy
ISAKMP: encryption DES-CBC
ISAKMP: hash MD5
ISAKMP: default group 2
ISAKMP: extended auth pre-share
ISAKMP: life type in seconds
ISAKMP: life duration (VPI) of 0x0 0x20 0xc4 0x9b
ISAKMP (0): atts are acceptable. Next payload is 3
!--- Attributes offered by the VPN Client are accepted by the PIX. ISAKMP (0): processing KE payload. message ID = 0 ISAKMP (0): processing NONCE payload. message ID = 0 ISAKMP (0): processing ID payload. message ID = 0 ISAKMP (0): processing vendor id payload ISAKMP (0): processing vendor id payload ISAKMP (0): remote peer supports dead peer detection ISAKMP (0): processing vendor id payload ISAKMP (0): speaking to a Unity client ISAKMP (0): ID payload next-payload: 10 type : 1 protocol : 17 port : 500 length : 8 ISAKMP (0) : Total payload length: 12 return status is IKMP_NO_ERROR crypto_isakmp_process_block: src 192.168.1.2, dest 192.168.1.1 OAK_AG exchange ISAKMP (0): processing HASH payload. message ID = 0 ISAKMP (0): processing NOTIFY payload 24578 protocol 1 spi 0, message ID = 0 ISAKMP (0): processing notify INITIAL_CONTACT IPSEC(key_engine): got a queue event... IPSEC(key_engine_delete_sas): rec'd delete notify from ISAKMP IPSEC(key_engine_delete_sas): delete all SAs shared with 192.168.1.2 ISAKMP (0): SA has been authenticated return status is IKMP_NO_ERROR ISAKMP/xauth: request attribute XAUTH_TYPE ISAKMP/xauth: request attribute XAUTH_USER_NAME ISAKMP/xauth: request attribute XAUTH_USER_PASSWORD ISAKMP (0:0): initiating peer config to 192.168.1.2. ID = 1623347510 (0x60c25136) crypto_isakmp_process_block: src 192.168.1.2, dest 192.168.1.1 ISAKMP_TRANSACTION exchange ISAKMP (0:0): processing transaction payload from 192.168.1.2. message ID = 84 ISAKMP: Config payload CFG_REPLY return status is IKMP_ERR_NO_RETRANS ISAKMP

(0:0): initiating peer config to 192.168.1.2. ID = 2620656926 (0x9c340d1e)
crypto_isakmp_process_block: src 192.168.1.2, dest 192.168.1.1 ISAKMP_TRANSACTION exchange
ISAKMP (0:0): processing transaction payload from 192.168.1.2. message ID = 60 ISAKMP: Config
payload CFG_ACK return status is IKMP_NO_ERROR crypto_isakmp_process_block: src 192.168.1.2,
dest 192.168.1.1 ISAKMP_TRANSACTION exchange ISAKMP (0:0): processing transaction payload from
192.168.1.2. message ID = 0 ISAKMP: Config payload CFG_REQUEST ISAKMP (0:0): checking request:
ISAKMP: attribute IP4_ADDRESS (1) ISAKMP: attribute IP4_NETMASK (2) ISAKMP: attribute IP4_DNS
(3) ISAKMP: attribute IP4_NBNS (4) ISAKMP: attribute ADDRESS_EXPIRY (5) Unsupported Attr: 5
ISAKMP: attribute APPLICATION_VERSION (7) Unsupported Attr: 7 ISAKMP: attribute UNKNOWN (28672)
Unsupported Attr: 28672 ISAKMP: attribute UNKNOWN (28673) Unsupported Attr: 28673 ISAKMP:
attribute UNKNOWN (28674) ISAKMP: attribute UNKNOWN (28676) ISAKMP: attribute UNKNOWN (28679)
Unsupported Attr: 28679 ISAKMP: attribute UNKNOWN (28680) Unsupported Attr: 28680 ISAKMP:
attribute UNKNOWN (28677) Unsupported Attr: 28677 ISAKMP (0:0): responding to peer config from
192.168.1.2. ID = 177917346 return status is IKMP_NO_ERROR crypto_isakmp_process_block: src
192.168.1.2, dest 192.168.1.1 OAK_QM exchange oakley_process_quick_mode: OAK_QM_IDLE ISAKMP (0):
processing SA payload. message ID = 942875080 ISAKMP : Checking IPsec proposal 1 ISAKMP:
transform 1, ESP_3DES ISAKMP: attributes in transform: ISAKMP: authenticator is HMAC-MD5 ISAKMP:
encaps is 1 ISAKMP: SA life type in seconds ISAKMP: SA life duration (VPI) of 0x0 0x20 0xc4 0x9b
IPSEC(validate_proposal): transform proposal (prot 3, trans 3, hmac_alg 1) not supported ISAKMP
(0): atts not acceptable. Next payload is 0 ISAKMP (0): skipping next ANDED proposal (1) ISAKMP
: Checking IPsec proposal 2 ISAKMP: transform 1, ESP_3DES ISAKMP: attributes in transform:
ISAKMP: authenticator is HMAC-SHA ISAKMP: encaps is 1 ISAKMP: SA life type in seconds ISAKMP: SA
life duration (VPI) of 0x0 0x20 0xc4 0x9b IPSEC(validate_proposal): transform proposal (prot 3,
trans 3, hmac_alg 2) not supported ISAKMP (0): atts not acceptable. Next payload is 0 ISAKMP
(0): skipping next ANDED proposal (2) ISAKMP: Checking IPsec proposal 3 ISAKMP: transform 1,
ESP_3DES ISAKMP: attributes in transform: ISAKMP: authenticator is HMAC-MD5 ISAKMP: encaps is 1
ISAKMP: SA life type in seconds ISAKMP: SA life duration (VPI) of 0x0 0x20 0xc4 0x9b
IPSEC(validate_proposal): transform proposal (prot 3, trans 3, hmac_alg 1) not supported ISAKMP
(0): atts not acceptable. Next payload is 0 ISAKMP: Checking IPsec proposal 4 ISAKMP: transform
1, ESP_3DES ISAKMP: attributes in transform: ISAKMP: authenticator is HMAC-SHA ISAKMP: encaps is
1 ISAKMP: SA life type in seconds ISAKMP: SA life duration (VPI) of 0x0 0x20 0xc4 0x9b
IPSEC(validate_proposal): transform proposal (prot 3, trans 3, hmac_alg 2) not supported ISAKMP
(0): atts not acceptable. Next payload is 0 ISAKMP : Checking IPsec proposal 5 ISAKMP: transform
1, ESP_DES ISAKMP: attributes in transform: ISAKMP: authenticator is HMAC-MD5 ISAKMP: encaps is
1 ISAKMP: SA life type in seconds ISAKMP: SA life duration (VPI) of 0x0 0x20 0xc4 0x9b ISAKMP
(0): atts are acceptable. ISAKMP (0): bad SPI size of 2 octets! ISAKMP: Checking IPsec proposal
6 ISAKMP: transform 1, ESP_DES ISAKMP: attributes in transform: ISAKMP: authenticator is HMAC-
SHA ISAKMP: encaps is 1 ISAKMP: SA life type in seconds ISAKMP: SA life duration (VPI) of 0x0
0x20 0xc4 0x9b IPSEC(validate_proposal): transform proposal (prot 3, trans 2, hmac_alg 2) not
supported ISAKMP (0): atts not acceptable. Next payload is 0 ISAKMP (0): skipping next ANDED
proposal (6) ISAKMP : Checking IPsec proposal 7 ISAKMP: transform 1, ESP_DES ISAKMP: attributes
in transform: ISAKMP: authenticator is HMAC-MD5 ISAKMP: encaps is 1 ISAKMP: SA life type in
seconds ISAKMP: SA life duration (VPI) of 0x0 0x20 0xc4 0x9b ISAKMP (0): atts are
acceptable.IPSEC(validate_proposal_request): proposal part #1, (key eng. msg.) dest=
192.168.1.1, src= 192.168.1.2, dest_proxy= 192.168.1.1/255.255.255.255/0/0 (type=1), src_proxy=
10.89.129.200/255.255.255.255/0/0 (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 = 942875080 ISAKMP (0): processing ID payload. message ID = 942875080
ISAKMP (0): ID_IPV4_ADDR src 10.89.129.200 prot 0 port 0 ISAKMP (0): processing ID payload.
message ID = 942875080 ISAKMP (0): ID_IPV4_ADDR dst 192.168.1.1 prot 0 port 0IPSEC(key_engine):
got a queue event... IPSEC(spi_response): getting spi 0x64d7a518(1691854104) for SA from
192.168.1.2 to 192.168.1.1 for prot 3 return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 192.168.1.2, dest 192.168.1.1 OAK_QM exchange
oakley_process_quick_mode: OAK_QM_IDLE ISAKMP (0): processing SA payload. message ID =
3008609960 ISAKMP: Checking IPsec proposal 1 ISAKMP: transform 1, ESP_3DES ISAKMP: attributes in
transform: ISAKMP: authenticator is HMAC-MD5 crypto_isakmp_process_block: src 192.168.1.2, dest
192.168.1.1 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 192.168.1.2 to 192.168.1.1 (proxy 10.89.129.200 to 192.168.1.1) has spi 1691854104 and
conn_id 2 and flags 4 lifetime of 2147483 seconds outbound SA from 192.168.1.1 to 192.168.1.2
(proxy 192.168.1.1 to 10.89.129.200) has spi 1025193431 and conn_id 1 and flags 4 lifetime of
2147483 seconds IPSEC(key_engine): got a queue event... IPSEC(initialize_sas): ,(key eng. msg.)
dest= 192.168.1.1, src= 192.168.1.2, dest_proxy= 192.168.1.1/0.0.0.0/0/0 (type=1), src_proxy=
10.89.129.200/0.0.0.0/0/0 (type=1), protocol= ESP, transform= esp-des esp-md5-hmac , lifedur=


```
2147483s and 0kb, spi= 0x64d7a518(1691854104),conn_id= 2, keysize= 0, flags= 0x4
IPSEC(initialize_sas): , (key eng. msg.) src= 192.168.1.1, dest=192.168.1.2, src_proxy=
192.168.1.1/0.0.0.0/0/0 (type=1), dest_proxy= 10.89.129.200/0.0.0.0/0/0 (type=1), protocol= ESP,
transform=esp-des esp-md5-hmac , lifedur= 2147483s and 0kb, spi= 0x3d1b35d7(1025193431),conn_id=
1, keysize= 0, flags= 0x4 VPN Peer: IPSEC: Peer ip:192.168.1.2 Ref cnt incremented to:2 Total
VPN Peers:1 VPN Peer: IPSEC: Peer ip:192.168.1.2 Ref cnt incremented to:3 Total VPN Peers:1
return status is IKMP_NO_ERROR crypto_isakmp_process_block: src 192.168.1.2, dest 192.168.1.1
OAK_QM exchange oakley_process_quick_mode: OAK_QM_AUTH_AWAITmap_alloc_entry: allocating entry 4
map_alloc_entry: allocating entry 3 ISAKMP (0): Creating IPsec SAs inbound SA from 192.168.1.2
to 192.168.1.1 (proxy 10.89.129.200 to 0.0.0.0) has spi 3415657865 and conn_id 4 and flags 4
lifetime of 2147483 seconds outbound SA from 192.168.1.1 to 192.168.1.2 (proxy 0.0.0.0 to
10.89.129.200) has spi 2383969893 and conn_id 3 and flags 4 lifetime of 2147483
secondsIPSEC(key_engine): got a queue event... IPSEC(initialize_sas): , (key eng. msg.) dest=
192.168.1.1, src=192.168.1.2, dest_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4), src_proxy=
10.89.129.200/0.0.0.0/0/0 (type=1), protocol= ESP, transform=esp-des esp-md5-hmac , lifedur=
2147483s and 0kb, spi= 0xcb96cd89(3415657865),conn_id= 4, keysize= 0, flags= 0x4
IPSEC(initialize_sas): , (key eng. msg.) src= 192.168.1.1, dest=192.168.1.2, src_proxy=
0.0.0.0/0.0.0.0/0/0 (type=4), dest_proxy= 10.89.129.200/0.0.0.0/0/0 (type=1), protocol= ESP,
transform=esp-des esp-md5-hmac , lifedur= 2147483s and 0kb, spi= 0x8e187e65(2383969893),conn_id=
3, keysize= 0, flags= 0x4 VPN Peer: IPSEC: Peer ip:192.168.1.2 Ref cnt incremented to:4 Total
VPN Peers:1 VPN Peer: IPSEC: Peer ip:192.168.1.2 Ref cnt incremented to:5 Total VPN Peers:1
return status is IKMP_NO_ERROR pixfirewall#show uauth
Current      Most Seen
Authenticated Users
1            1
Authen In Progress
0            1
ipsec user 'cisco_customer' at 10.89.129.200, authenticated
pixfirewall#
```

[Debugger mit VPN Client 1.1](#)

```
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
VPN Peer: ISAKMP: Added new peer: ip:192.168.1.3
Total VPN Peers:1
VPN Peer: ISAKMP: Peer ip:192.168.1.3 Ref cnt incremented to:1
Total VPN Peers:1
OAK_MM exchange
ISAKMP (0): processing SA payload. message ID = 0

ISAKMP (0): Checking ISAKMP transform 1 against priority 10 policy
encryption DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 1
ISAKMP:      auth pre-share
ISAKMP (0): atts are not acceptable. Next payload is 0
ISAKMP (0): Checking ISAKMP transform 1 against priority 20 policy
ISAKMP:      encryption DES-CBC
ISAKMP:      hash MD5
ISAKMP:      default group 1
ISAKMP:      auth pre-share
ISAKMP (0): atts are acceptable. Next payload is 0
ISAKMP (0): SA is doing pre-shared key authentication
using id type ID_IPV4_ADDR
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
OAK_MM exchange
ISAKMP (0): processing KE payload. message ID = 0

ISAKMP (0): processing NONCE payload. message ID = 0

ISAKMP (0): processing vendor id payload
```

```
ISAKMP (0): processing vendor id payload

return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
OAK_MM exchange
ISAKMP (0): processing ID payload. message ID = 0
ISAKMP (0): processing HASH payload. message ID = 0
ISAKMP (0): processing NOTIFY payload 24578 protocol 1
spi 0, message ID = 0
ISAKMP (0): SA has been authenticated

ISAKMP (0): ID payload
next-payload : 8
type          : 1
protocol      : 17
port          : 500
length        : 8
ISAKMP (0): Total payload length: 12
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
ISAKMP: Created a peer node for 192.168.1.3
OAK_QM exchange
ISAKMP (0:0): Need XAUTH
ISAKMP/xauth: request attribute XAUTH_TYPE
ISAKMP/xauth: request attribute XAUTH_USER_NAME
ISAKMP/xauth: request attribute XAUTH_USER_PASSWORD
ISAKMP (0:0): initiating peer config to 192.168.1.3.
ID = 3196940891 (0xbe8d725b)
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
ISAKMP_TRANSACTION exchange
ISAKMP (0:0): processing transaction payload
from 192.168.1.3. message ID = 84
ISAKMP: Config payload CFG_REPLY
return status is IKMP_ERR_NO_RETRANS
ISAKMP (0:0): initiating peer config to 192.168.1.3.
ID = 3196940891 (0xbe8d725b)
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
ISAKMP_TRANSACTION exchange
ISAKMP (0:0): processing transaction payload
from 192.168.1.3. message ID = 60
ISAKMP: Config payload CFG_ACK
ISAKMP (0:0): initiating peer config to 192.168.1.3.
ID = 1647424595 (0x6231b453)
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
ISAKMP_TRANSACTION exchange
ISAKMP (0:0): processing transaction payload
from 192.168.1.3. message ID = 60
ISAKMP: Config payload CFG_ACK
ISAKMP (0:0): peer accepted the address!
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_IDLE
ISAKMP (0): processing SA payload. message ID = 802013669

ISAKMP : Checking IPsec proposal 1

ISAKMP: transform 1, ESP_DES
ISAKMP:  attributes in transform:
ISAKMP:      authenticator is HMAC-MD5
ISAKMP:      encaps is 1
```

```

ISAKMP (0): atts are acceptable.IPSEC(validate_proposal_request)
:proposal part #1,
  (key eng. msg.) dest= 192.168.1.1, src = 192.168.1.3,
  dest_proxy= 10.89.129.128/255.255.255.128/0/0 (type=4),
  src_proxy= 10.89.129.200/255.255.255.255/0/0 (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 = 802013669

ISAKMP (0): processing ID payload. message ID = 802013669
ISAKMP (0): ID_IPV4_ADDR src 10.89.129.200 prot 0 port 0
ISAKMP (0): processing ID payload. message ID = 802013669
ISAKMP (0): ID_IPV4_ADDR_SUBNET dst 10.89.129.128/255.255.255.128
prot 0 port 0IPSEC(key_engine): got a queue event...
IPSEC(spi_response): getting spi 0xd7cef5ba(3620664762)for SA
  from 192.168.1.3 to 192.168.1.1 for prot 3

return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 192.168.1.3, dest 192.168.1.1
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_AUTH_AWAITmap_alloc_entry: allocating entry 1
map_alloc_entry: allocating entry 2

ISAKMP (0): Creating IPsec SAs
  inbound SA from 192.168.1.3 to 192.168.1.1
  (proxy 10.89.129.200 to 10.89.129.128)
  has spi 3620664762 and conn_id 1 and flags 4
  outbound SA from 192.168.1.1 to 192.168.1.3
  (proxy 10.89.129.128 to 10.89.129.200)
  has spi 541375266 and conn_id 2 and flags 4
IPSEC(key_engine): got a queue event...

IPSEC(initialize_sas): ,
  (key eng. msg.) dest= 192.168.1.1, src=192.168.1.3,
  dest_proxy= 10.89.129.128/255.255.255.128/0/0 (type=4),
  src_proxy= 10.89.129.200/0.0.0.0/0/0 (type=1),
  protocol= ESP, transform=esp-des esp-md5-hmac ,
  lifedur= 0s and 0kb,
  spi= 0xd7cef5ba(3620664762),conn_id= 1, keysize= 0, flags= 0x4
IPSEC(initialize_sas): ,
  (key eng. msg.) src= 192.168.1.1, dest=192.168.1.3,
  src_proxy= 10.89.129.128/255.255.255.128/0/0 (type=4),
  dest_proxy= 10.89.129.200/0.0.0.0/0/0 (type=1),
  protocol= ESP, transform=esp-des esp-md5-hmac ,
  lifedur= 0s and 0kb,
  spi= 0x2044bb22(541375266),conn_id= 2, keysize= 0, flags= 0x4

VPN Peer: IPSEC: Peer ip:192.168.1.3 Ref cnt incremented
to:2 Total VPN Peers:1
VPN Peer: IPSEC: Peer ip:192.168.1.3 Ref cnt incremented
to:3 Total VPN Peers:1
return status is IKMP_NO_ERROR

```

[Zugehörige Informationen](#)

- [Security Appliances der Serie PIX 500](#)
- [PIX-Befehlsreferenzen](#)
- [IPsec-Aushandlung/IKE-Protokolle](#)
- [Einführung in IPsec](#)

- [Konnektivität über Cisco PIX-Firewalls](#)
- [Anforderungen für Kommentare \(RFCs\)](#)
- [Technischer Support und Dokumentation - Cisco Systems](#)