IPSec - Configurazione modalità da PIX a Cisco VPN Client Wild-card, precondivisa con autenticazione estesa

Sommario

Introduzione

In questo esempio di configurazione viene mostrato come connettere un client VPN a un firewall PIX utilizzando i caratteri jolly, mode-config, il comando **syspot connection allow-ipsec** e l'autenticazione estesa (Xauth).

Per visualizzare la configurazione di TACACS+ e RADIUS per PIX 6.3 e versioni successive, fare riferimento agli esempi di configurazione di TACACS+ e RADIUS per PIX 6.3 e PIX/ASA 7.x.

Il client VPN supporta AES (Advanced Encryption Standard) come algoritmo di crittografia in Cisco VPN Client versione 3.6.1 e successive e con PIX Firewall 6.3. Il client VPN supporta solo dimensioni della chiave di 128 bit e 256 bit. Per ulteriori informazioni su come configurare AES, consultare il documento sulla configurazione del client VPN Cisco su PIX con AES.

Per configurare la connessione VPN di accesso remoto tra un client VPN Cisco (4.x per Windows) e l'appliance di sicurezza PIX serie 500 7.x con un server RADIUS Microsoft Windows 2003 Internet Authentication Service (IAS), fare riferimento agli esempi di configurazione dell'autenticazione RADIUS PIX/ASA 7.x e Cisco VPN Client 4.x per Windows con Microsoft Windows 2003.

Per stabilire un tunnel IPsec tra un concentratore VPN 3000 e un client VPN 4.x per Windows che utilizza RADIUS per l'autenticazione utente e la configurazione dell'accounting, fare riferimento a IPsec Between a VPN 3000 Concentrator for Windows for User Authentication and Accounting (IPsec tra un concentratore VPN 3000 e un client VPN 4.x per Windows che utilizza RADIUS per l'autenticazione utente e l'accounting).

Per configurare una connessione tra un router e il client VPN Cisco 4.x con RADIUS per l'autenticazione dell'utente, consultare il documento sulla configurazione di IPsec tra un router Cisco IOS e un client VPN Cisco 4.x per Windows con RADIUS per l'autenticazione dell'utente.

Prerequisiti

Requisiti

Nessun requisito specifico previsto per questo documento.

Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- Cisco VPN Client 4.x. A differenza di Cisco Secure VPN Client 1.x, questo prodotto dispone di funzionalità VPN avanzate.
- PIX Firewall 515E versione 6.3(3).

Nota: la tecnologia di crittografia è soggetta ai controlli sulle esportazioni. È tua responsabilità conoscere la legge relativa all'esportazione della tecnologia di crittografia. Per ulteriori informazioni, visitare il sito Web Bureau of Export Administration . In caso di domande sul controllo delle esportazioni, invia un'e-mail a export@cisco.com.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

Convenzioni

Per ulteriori informazioni sulle convenzioni usate, consultare il documento Cisco sulle convenzioni nei suggerimenti tecnici.

Premesse

Il comando **sysost connection allow-ipsec** consente in modo implicito a qualsiasi pacchetto proveniente da un tunnel IPsec di ignorare la verifica di un comando **access-list**, **conduit** o **access-group** associato per le connessioni IPsec. Xauth autentica l'utente IPsec in un server TACACS+ o RADIUS esterno. Oltre alla chiave già condivisa con caratteri jolly, l'utente deve fornire un nome utente/password.

Un utente con un client VPN riceve un indirizzo IP dal proprio ISP. Viene sostituito da un indirizzo IP del pool di indirizzi IP sul PIX. L'utente ha accesso a tutto ciò che si trova all'interno del firewall,

incluse le reti. Gli utenti che non eseguono il client VPN possono connettersi solo al server Web utilizzando l'indirizzo esterno fornito dall'assegnazione statica.

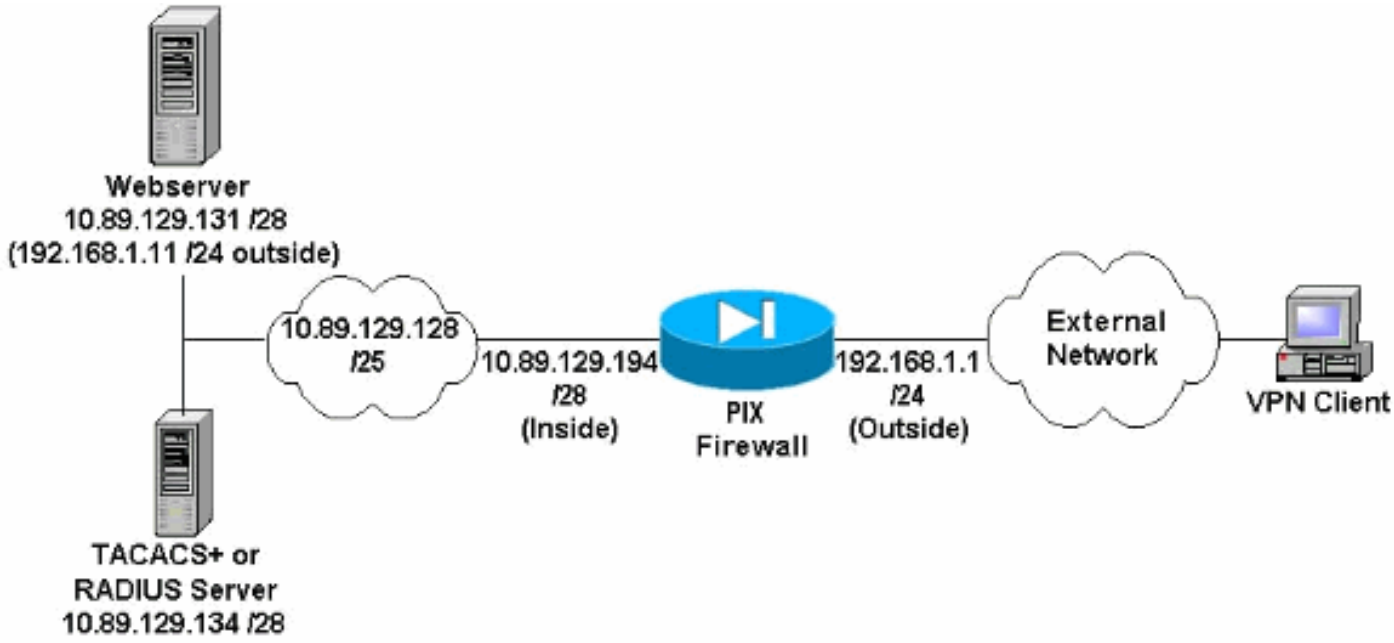
Configurazione

In questa sezione vengono presentate le informazioni necessarie per configurare le funzionalità descritte più avanti nel documento.

Nota: per ulteriori informazioni sui comandi menzionati in questo documento, usare lo strumento di ricerca dei comandi (solo utenti registrati).

Esempio di rete

Nel documento viene usata questa impostazione di rete:



Note diagramma reticolare

- Gli host Internet che accedono al server Web utilizzando l'indirizzo IP globale 192.168.1.1 vengono autenticati anche se non viene stabilita una connessione VPN. Il traffico *non* è crittografato.
- I client VPN sono in grado di accedere a tutti gli host della rete interna (10.89.129.128 /25) una volta stabilito il tunnel IPsec. Tutto il traffico tra il client VPN e il firewall PIX è crittografato. Senza un tunnel IPsec, possono accedere al server Web solo tramite il relativo indirizzo IP globale, ma devono comunque eseguire l'autenticazione.
- I client VPN provengono da Internet e i relativi indirizzi IP non sono noti in anticipo.

Configurazioni

Nel documento vengono usate queste configurazioni.

- PIX Configuration 6.3(3)
- Configurazione VPN Client 4.0.5

- Configurazione VPN Client 3.5
- Configurazione VPN Client 1.1

PIX Configuration 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
```

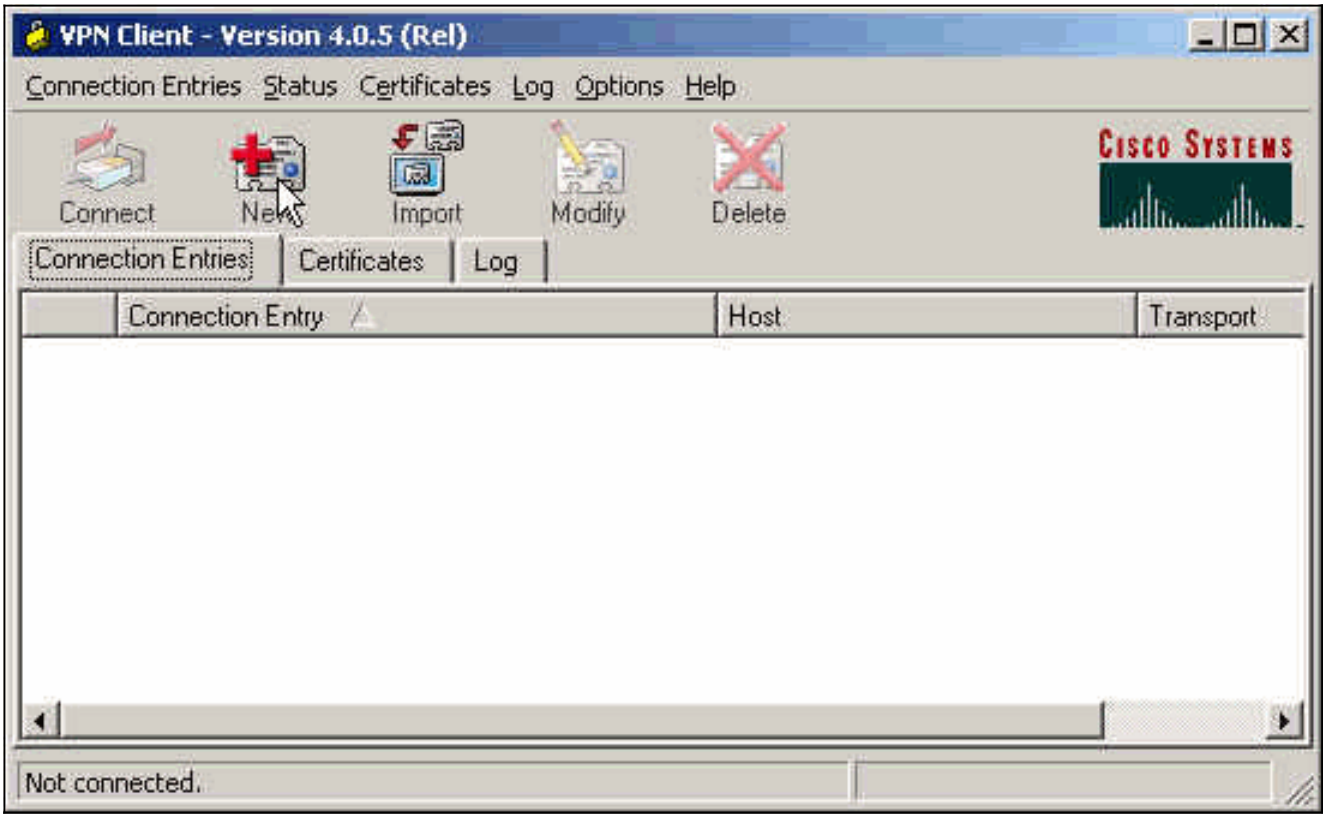
```
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc
0:10:00 h225 1:00:00 timeout h323 0:05:00 mgcp 0:05:00
sip 0:30:00 sip_media 0:02:00 timeout uauth 0:05:00
absolute aaa-server TACACS+ protocol tacacs+ aaa-server
RADIUS protocol radius aaa-server LOCAL protocol local
!--- Authentication, authorization, 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#
```

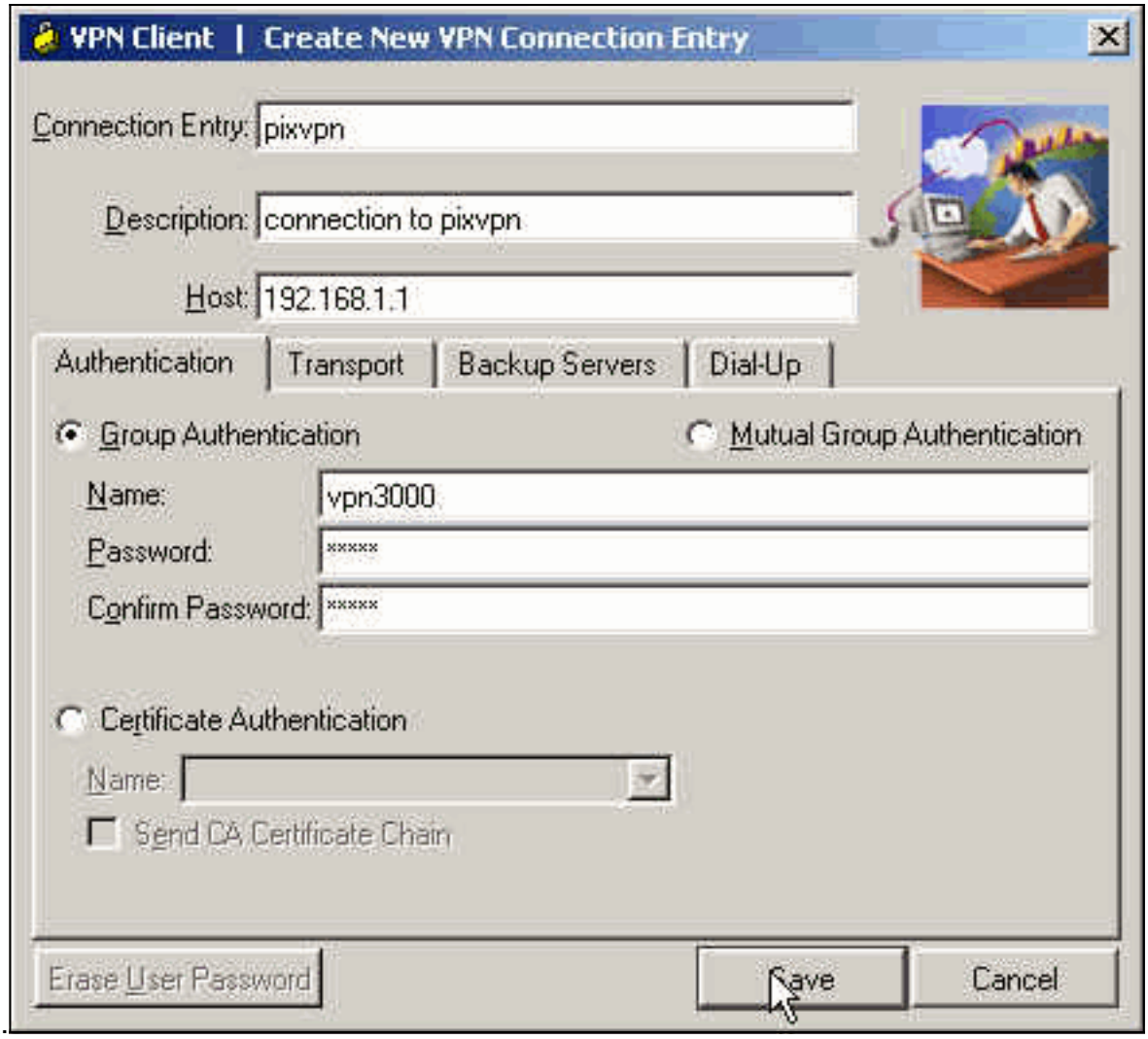
Configurazione VPN Client 4.0.5

Completare questa procedura per configurare il client VPN 4.0.5.

1. Selezionare **Start > Programmi > Cisco Systems VPN Client > VPN Client**.
2. Fare clic su **Nuovo** per avviare la finestra Crea nuova voce di connessione VPN.

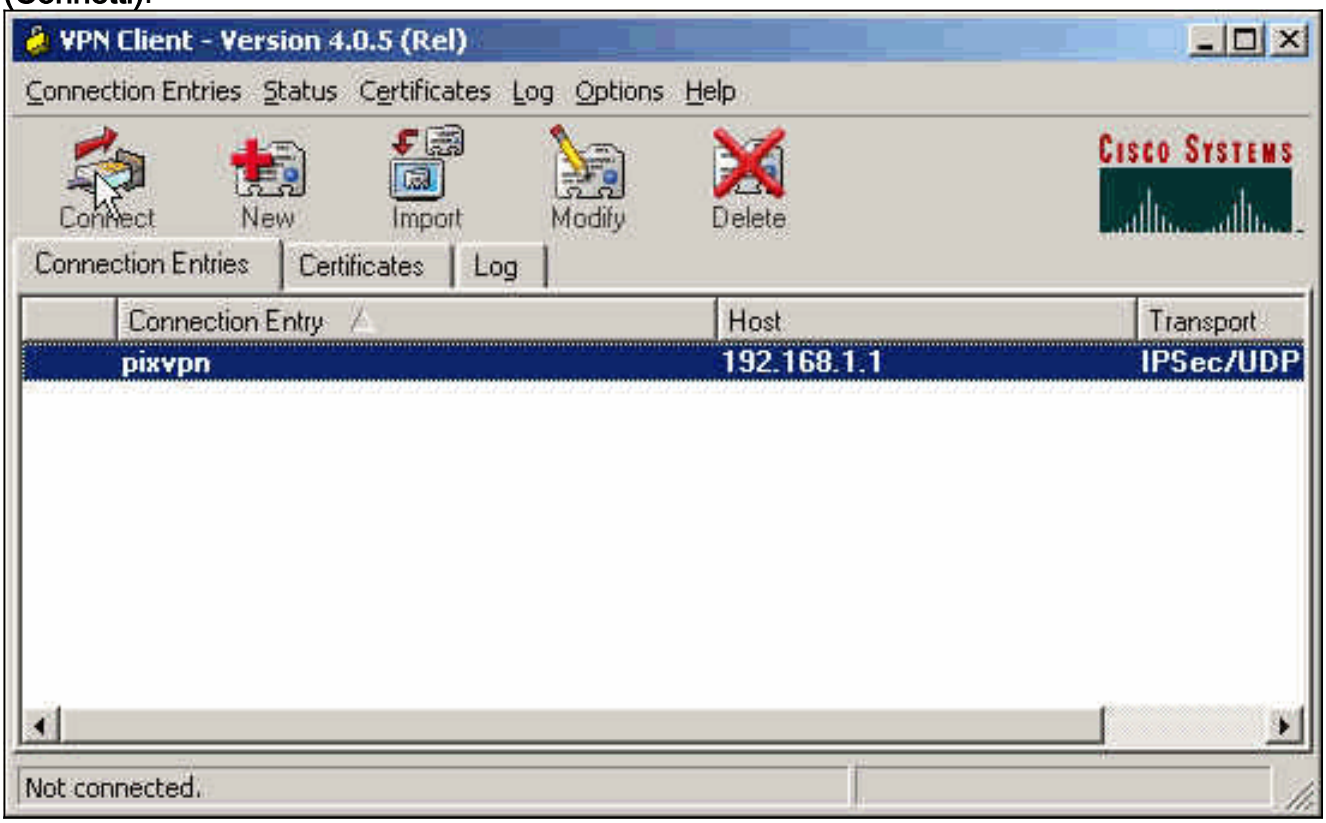


3. Immettere il nome della voce di connessione insieme a una descrizione. Immettere l'indirizzo IP esterno del firewall PIX nella casella Host. Immettere il nome e la password del gruppo VPN e fare clic su



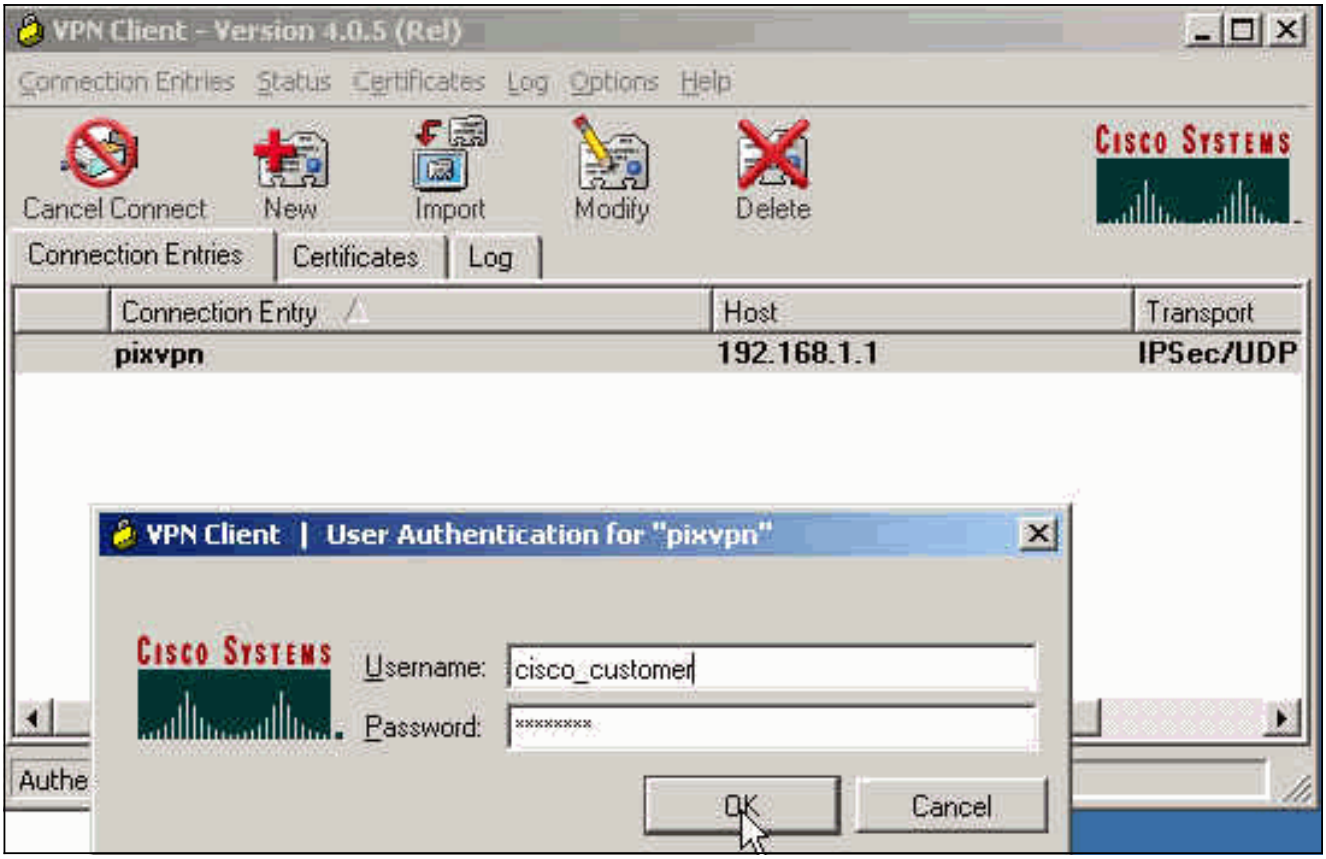
Salva.

4. Dalla finestra principale di VPN Client, fare clic sulla connessione che si desidera utilizzare e fare clic sul pulsante **Connect (Connetti)**.



5. Quando richiesto, immettere il nome utente e la password per Xauth e fare clic su **OK** per

connettersi alla rete
remota.



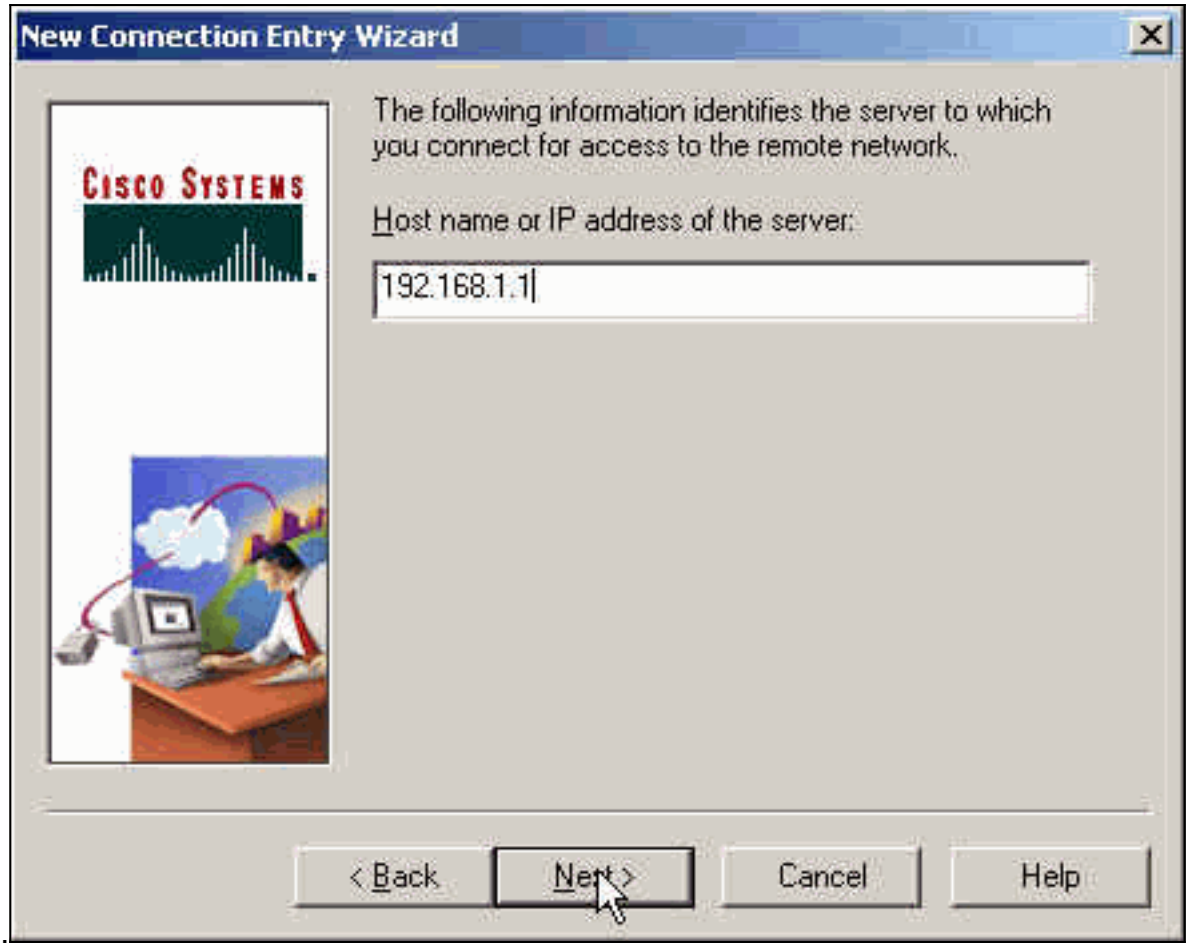
Configurazione VPN Client 3.5

Completare questa procedura per configurare la configurazione di VPN Client 3.5.

1. Selezionare **Start > Programmi > Cisco Systems VPN Client > VPN Dialer**.
2. Fare clic su **Nuovo** per avviare la Creazione guidata nuova voce di connessione.
3. Immettere il nome della nuova voce di connessione e fare clic su

Avanti.

4. Immettere il nome host o l'indirizzo IP del server utilizzato per connettersi al server remoto e fare clic su



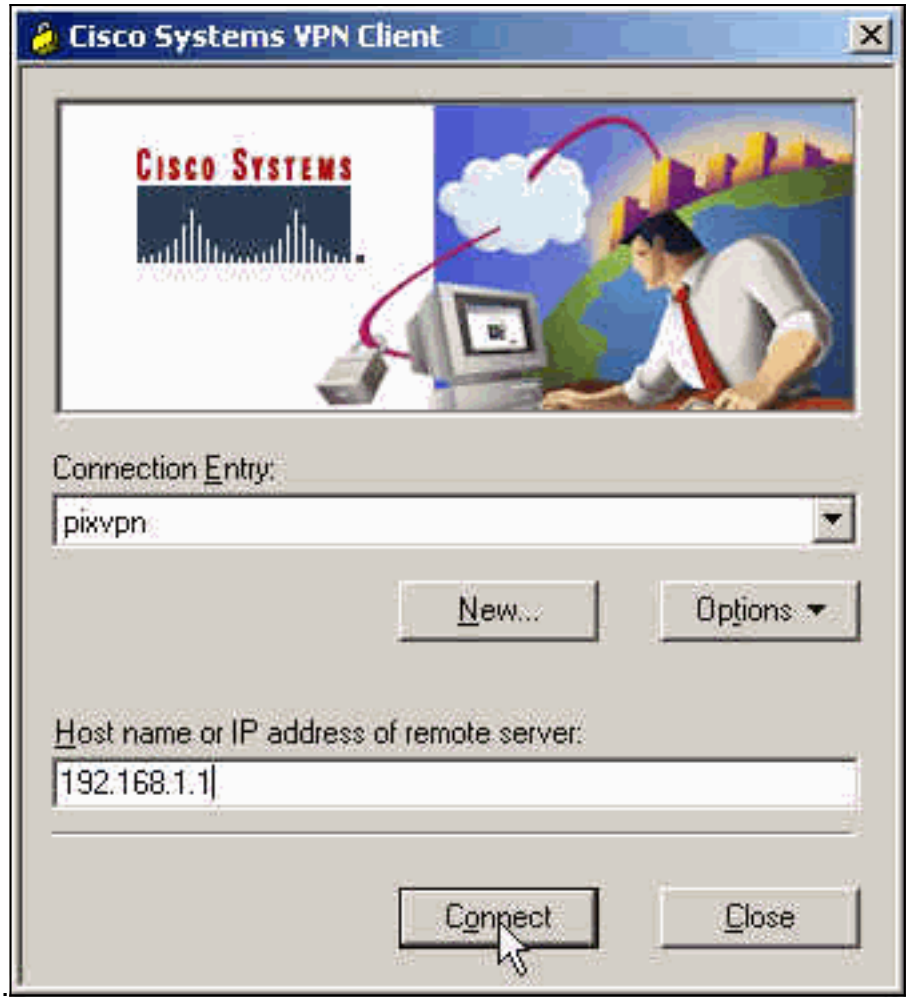
Avanti.

5. Selezionare **Group Access Information** e immettere il nome e la password utilizzati per

autenticare l'accesso al server remoto. Fare clic su **Next** (Avanti).

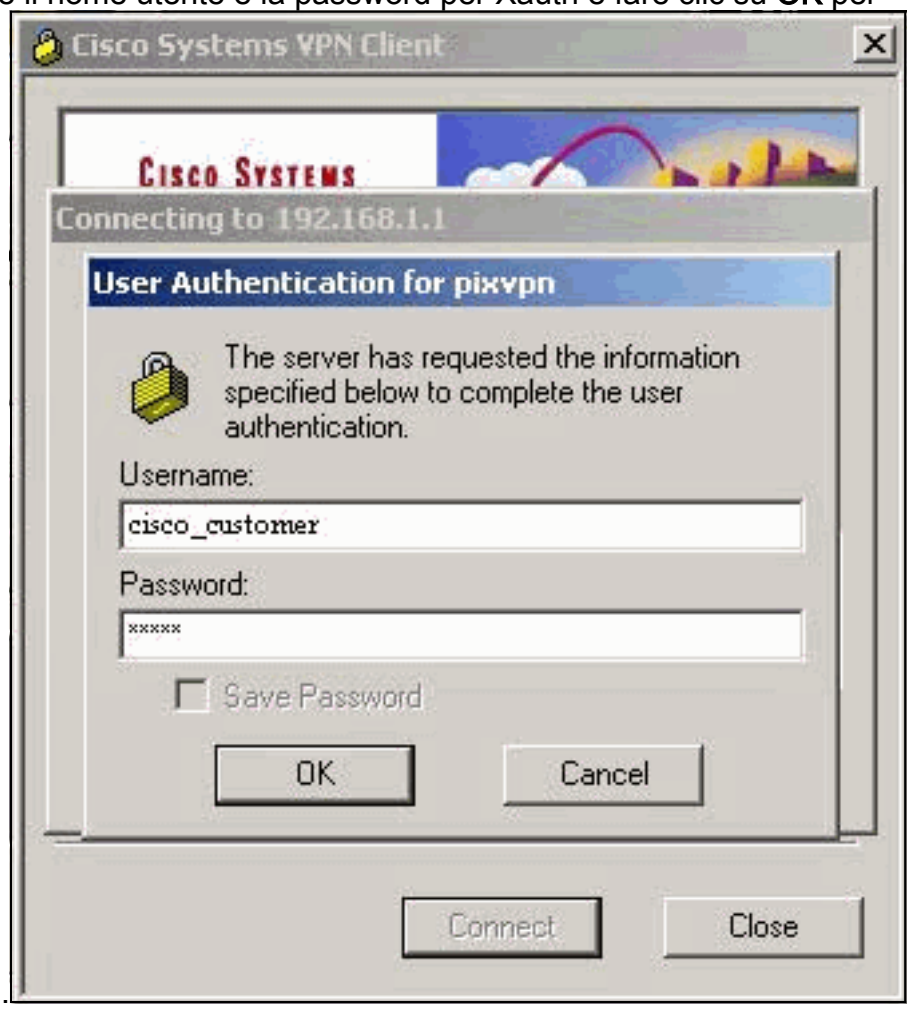
6. Fare clic su **Fine** per salvare la nuova voce.

7. Selezionare la voce di connessione nella finestra di composizione e fare clic su



Connetti.

8. Quando richiesto, immettere il nome utente e la password per Xauth e fare clic su **OK** per connettersi alla rete remota.



Configurazione VPN Client 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
```

Aggiungi accounting

La sintassi del comando per aggiungere l'accounting è:

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

Ad esempio, nella configurazione PIX, viene aggiunto questo comando:

aaa accounting include any inbound

```
0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 AuthInbound
```

Nota: per il funzionamento dell'accounting Xauth è necessario il comando **sysopt connection allow-ipsec** e non il comando **sysopt ipsec pl-compatible**. L'accounting Xauth non funziona solo con il comando **sysopt ipsec compatibile con pl**. L'accounting Xauth è valido per le connessioni TCP, non per ICMP o UDP.

Questo output è un esempio di record contabili TACACS+:

```
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
```

Verifica

Per verificare che la configurazione funzioni correttamente, consultare questa sezione.

Lo strumento Output Interpreter (solo utenti registrati) (OIT) supporta alcuni comandi **show**. Usare l'OIT per visualizzare un'analisi dell'output del comando **show**.

Nota: consultare le informazioni importanti sui comandi di debug prima di usare i comandi di **debug**.

Abilitare Cisco Secure Log Viewer per visualizzare i debug sul lato client.

- **debug crypto ipsec**: utilizzato per visualizzare le negoziazioni IPsec della fase 2.
- **debug crypto isakmp**: utilizzato per visualizzare le negoziazioni ISAKMP della fase 1.

Risoluzione dei problemi

Le informazioni contenute in questa sezione permettono di risolvere i problemi relativi alla configurazione. Viene visualizzato anche l'output di esempio del comando debug.

Comandi per la risoluzione dei problemi

Lo strumento Output Interpreter (solo utenti registrati) (OIT) supporta alcuni comandi **show**. Usare l'OIT per visualizzare un'analisi dell'output del comando **show**.

Nota: consultare le informazioni importanti sui comandi di debug prima di usare i comandi di **debug**.

- **debug crypto engine**: utilizzato per eseguire il debug del processo del motore di crittografia.

Esempio di debug PIX

```
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
```

Debug con 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#

Debug con 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
```

Informazioni correlate

- PIX serie 500 Security Appliance
- Riferimenti per i comandi PIX
- Negoziazione IPSec/protocolli IKE
- Introduzione a IPSec
- Definizione della connettività tramite i firewall Cisco PIX
- RFC (Requests for Comments)
- Documentazione e supporto tecnico – Cisco Systems