

PIX/ASA: Multiple VPN Group Clients to use Different VLANs after Connecting to a Security Appliance Configuration Example

Document ID: 69393

Introduction

Prerequisites

- Requirements
- Components Used
- Related Products
- Conventions

Background Information

Configure

- Network Diagram
- Configurations
- VPN Client 4.8 Configuration

Verify

Troubleshoot

- Troubleshooting Commands
- Sample debug Output

Related Information

Introduction

This sample configuration shows how to set up multiple VPN Group Clients to use different VLANs after the IPsec tunnel is established with the PIX 500 Series Security Appliance.

Prerequisites

Requirements

Ensure that you meet this requirement before you attempt this configuration:

- The PIX 500 Series Security Appliance 7.x and VPN Client 4.x are reachable from the Internet.

Components Used

The information in this document is based on these software and hardware versions:

- PIX 515E Series Security Appliance Software Release 7.1(1)
- Cisco VPN Client version 4.8 for Windows

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

Related Products

You can also use this configuration with the Cisco ASA 5500 Series Adaptive Security Appliance.

Conventions

Refer to the Cisco Technical Tips Conventions for more information on document conventions.

Background Information

In this configuration example, there are two VPN Clients (user1 and user2) and there are two different VLANs named vlan2 and vlan3. Once the IPsec tunnel is established, user1 should be able to connect only to vlan2 and user2 should be able to connect only to vlan3.

Vlan2 is created under the sub-interface (Ethernet 1.1) and vlan3 is created under the sub-interface (Ethernet 1.2) of the Ethernet 1 interface of the PIX Security Appliance. You must enable the physical interface before any traffic can pass through an enabled sub-interface.

In general, if the IPsec tunnel is established from the Cisco VPN Client to the PIX Firewall, all the traffic is sent through the tunnel to the PIX Firewall. This can become very costly in terms of resource usage if many clients are connected at once. In order to avoid such heavy resource usage, you can use split tunneling. Split tunneling encrypts only the interesting traffic and the rest of the traffic goes to the Internet and does not get encrypted into the tunnel.

Note: If you would like to tunnel all traffic before you send it out to the Internet, refer to PIX/ASA 7.x and VPN Client for Public Internet VPN on a Stick Configuration Example for more information.

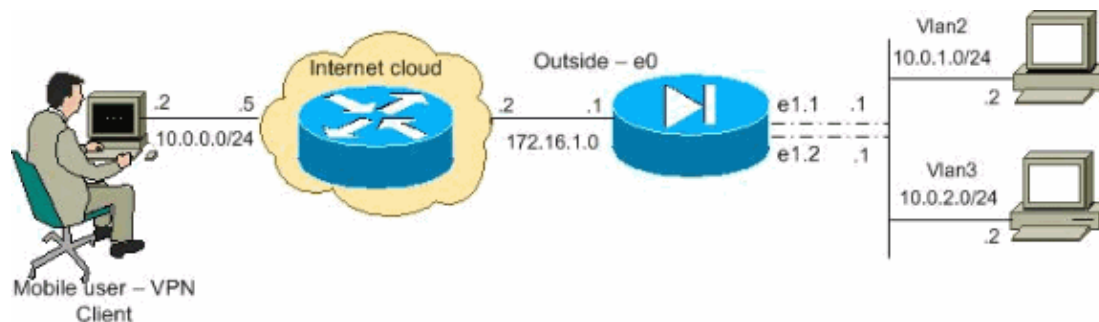
Configure

In this section, you are presented with the information to configure the multiple remote access VPN connections with the different VLANs in the PIX Security Appliance.

Note: Use the Command Lookup Tool (registered customers only) to obtain more information on the commands used in this section.

Network Diagram

This document uses this network setup:



Configurations

This document uses these configurations:

- PIX 515E Security Appliance Configuration
- Cisco VPN Client 4.8 for Windows Configuration

PIX 515E Security Appliance Configuration

```
PIX Version 7.1(1)
!
hostname pix
enable password 9jNfZuG3TC5tCVH0 encrypted
names
!
interface Ethernet0
  nameif outside
  security-level 0
  ip address 172.16.1.1 255.255.255.0
!
interface Ethernet1
  no nameif
  no security-level
  no ip address

!--- Configure the sub-interfaces on the inside interface.
!--- Configure VLAN to the respective sub-interfaces.

!
interface Ethernet1.1
  vlan 2
  nameif vlan2
  security-level 100
  ip address 10.0.1.1 255.255.255.0
!
interface Ethernet1.2
  vlan 3
  nameif vlan3
  security-level 100
  ip address 10.0.2.1 255.255.255.0
!

!--- Output is suppressed.

!
passwd 9jNfZuG3TC5tCVH0 encrypted
ftp mode passive

!--- This access list is used for a nat zero command that prevents
!--- traffic from undergoing network address translation (NAT).

access-list no-nat-vpn1-group extended permit ip 10.0.1.0 255.255.255.0 10.0.1.0 255.255.255.0
access-list no-nat-vpn2-group extended permit ip 10.0.2.0 255.255.255.0 10.0.2.0 255.255.255.0

!--- This access list is used for the split tunneling
!--- to be downloaded to the VPN Client to tell the interesting traffic to be encrypted.
```

```
access-list SPLIT-Tunnel-vpn1group standard permit 10.0.1.0 255.255.255.0
access-list SPLIT-Tunnel-vpn2group standard permit 10.0.2.0 255.255.255.0

pager lines 24
logging console debugging
mtu outside 1500
mtu vlan2 1500
mtu vlan3 1500

!--- Create a pool of addresses from which IP addresses are assigned
!--- dynamically to the remote VPN Clients.
!--- The pool user1 IP address is assigned to the tunnel group (vpn1).
!--- The pool user2 IP address is assigned to the tunnel group (vpn2).

ip local pool user1 10.0.1.10-10.0.1.15 mask 255.255.255.0
ip local pool user2 10.0.2.10-10.0.2.15 mask 255.255.255.0

no failover
no asdm history enable
arp timeout 14400

!--- NAT 0 prevents NAT for the networks specified in the access list.
!--- The nat 1 command specifies port address translation (PAT)
!--- using the outside interface IP address for all other traffic.

global (outside) 1 interface
nat (vlan2) 0 access-list no-nat-vpn1-group
nat (vlan2) 1 0.0.0.0 0.0.0.0
nat (vlan3) 0 access-list no-nat-vpn2-group
nat (vlan3) 1 0.0.0.0 0.0.0.0

route outside 0.0.0.0 0.0.0.0 172.16.1.2 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 icmp 0:00:02
timeout sunrpc 0:10:00 h323 0:05:00 h225 1:00:00 mgcp 0:05:00
timeout mgcp-pat 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute

!--- Enter group-policy attributes mode for the group policy (vpn2).

group-policy vpn2 internal
group-policy vpn2 attributes

!--- The split tunnel policy tunnels all traffic from or to the specified networks.

split-tunnel-policy tunnelspecified

!--- Split tunnel in group-policy configuration mode identifies
!--- an access list (SPLIT-Tunnel-vpn2group) that enumerates the network to be
!--- tunneled from the VPN Client.
!--- After the IPsec tunnel formation, the access list (SPLIT-Tunnel-vpn2group) has to be
!--- downloaded to the VPN Client of vpn2 (tunnel group).

split-tunnel-network-list value SPLIT-Tunnel-vpn2group
```

```
group-policy vpn1 internal
group-policy vpn1 attributes
  split-tunnel-policy tunnelspecified
  split-tunnel-network-list value SPLIT-Tunnel-vpn1group

!--- Configure usernames and passwords
!--- to identify remote access users to the PIX Security Appliance.

username vpn2 password 5RBT6B6kO6ZsK4e3 encrypted
username vpn1 password Rgp2OnMV8tB9079o encrypted

no snmp-server location
no snmp-server contact
snmp-server enable traps snmp authentication linkup linkdown coldstart

!--- PHASE 2 CONFIGURATION ---!
!--- The encryption types for Phase 2 are defined here.
!--- A single DES encryption with
!--- the md5 hash algorithm is used.

crypto ipsec transform-set my-set esp-des esp-md5-hmac

!--- Defines a dynamic crypto map with
!--- the specified encryption settings.

crypto dynamic-map dynmap 10 set transform-set my-set

!--- Enable Reverse Route Information (RRI), which allows the
!--- PIX Security Appliance to learn routing information for connected clients.

crypto dynamic-map dynmap 10 set reverse-route

!--- Binds the dynamic map to the IPsec/ISAKMP process.

crypto map mymap 10 ipsec-isakmp dynamic dynmap

!--- Specifies the interface to be used with
!--- the settings defined in this configuration.

crypto map mymap interface outside

!--- PHASE 1 CONFIGURATION ---!
!--- This configuration uses ISAKMP policy 10.
!--- Policy 65535 is included in the configuration by default.
!--- The configuration commands here define the Phase
!--- 1 policy parameters that are used.

isakmp enable outside
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption des
isakmp policy 10 hash sha
isakmp policy 10 group 2
```

```
isakmp policy 10 lifetime 1000

isakmp policy 65535 authentication pre-share
isakmp policy 65535 encryption 3des
isakmp policy 65535 hash sha
isakmp policy 65535 group 2
isakmp policy 65535 lifetime 86400
tunnel-group vpn type ipsec-ra

!--- Sets the connection type to IPsec remote access (ipsec-ra).

tunnel-group vpn1 type ipsec-ra

!--- Configures an address pool for the tunnel group and enters the general-attributes mode.
!--- Associates the user1 pool to the tunnel group (vpn1) that uses the address pool.

tunnel-group vpn1 general-attributes
address-pool user1

!--- Specifies the set of attributes that the user inherits by default
!--- in tunnel-group general-attributes configuration mode.
!--- Tunnel groups identify the group policy for a specific connection.

default-group-policy vpn1

!--- Enter the ipsec-attributes mode to configure the authentication method
!--- by entering the preshared key.
!--- You need to use the same preshared key on both
!--- devices (PIX and VPN Client) for this remote access connection.

tunnel-group vpn1 ipsec-attributes
pre-shared-key *

tunnel-group vpn2 type ipsec-ra
tunnel-group vpn2 general-attributes
address-pool user2
default-group-policy vpn2
tunnel-group vpn2 ipsec-attributes
pre-shared-key *

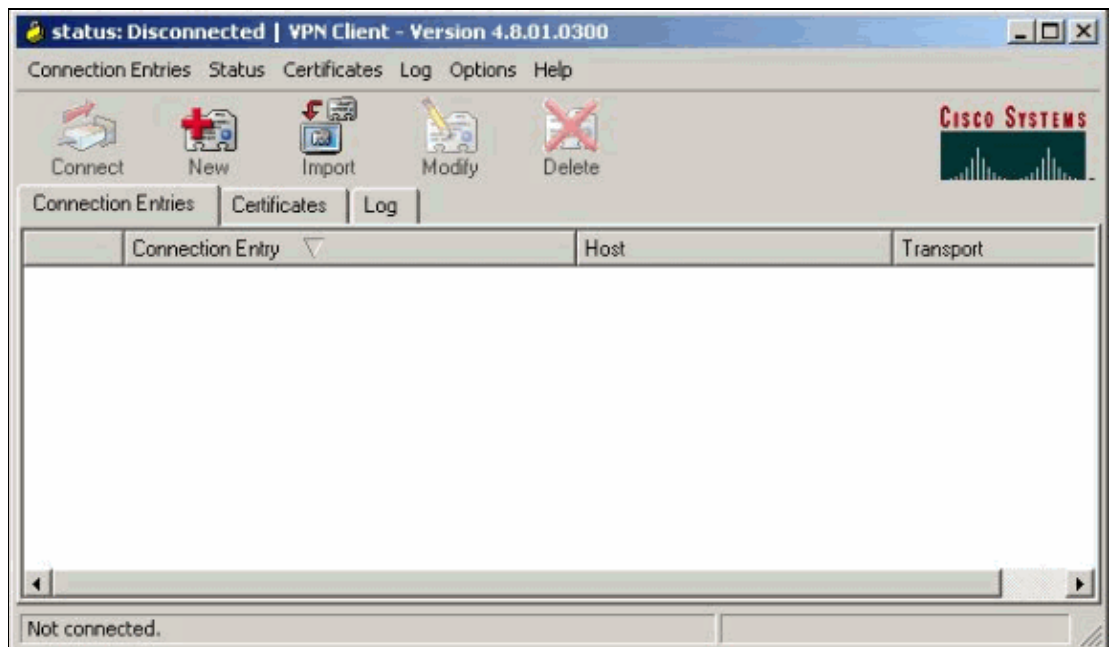
telnet timeout 5
ssh timeout 5
console timeout 0
!
class-map inspection_default
match default-inspection-traffic
!
!
policy-map global_policy
class inspection_default
inspect dns maximum-length 512
inspect ftp
inspect h323 h225
inspect h323 ras
inspect netbios
inspect rsh
inspect rtsp
inspect skinny
```

```
inspect esmtp
inspect sqlnet
inspect sunrpc
inspect tftp
inspect sip
inspect xdmcp
!
service-policy global_policy global
Cryptochecksum:0becb57df25d69a098b25bf07994b6b6
: end
pix#
```

VPN Client 4.8 Configuration

Complete these steps to configure VPN Client 4.8.

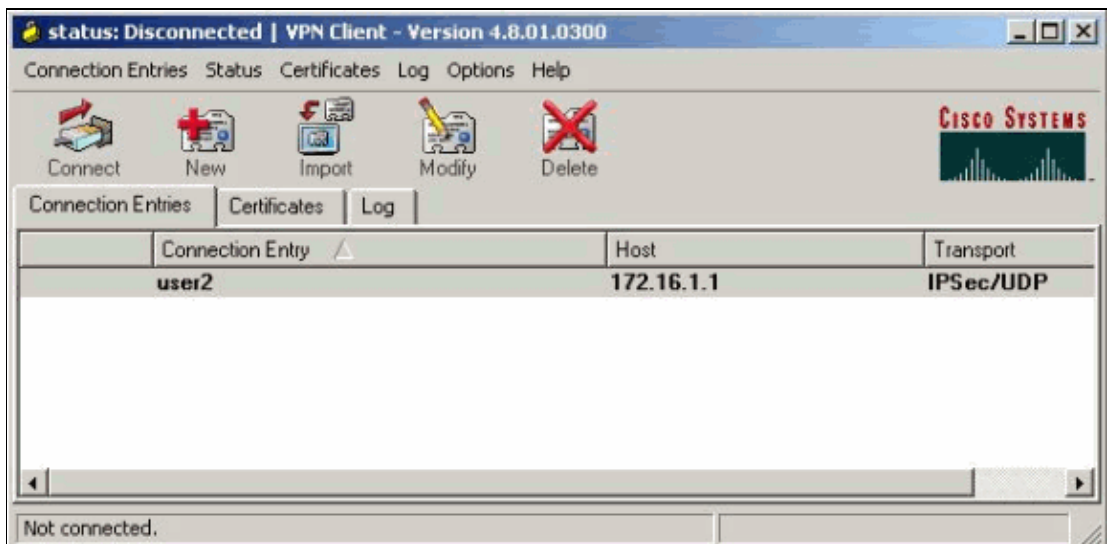
1. Select **Start > Programs > Cisco Systems VPN Client > VPN Client**.
2. Click **New** to launch the Create New VPN Connection Entry window.



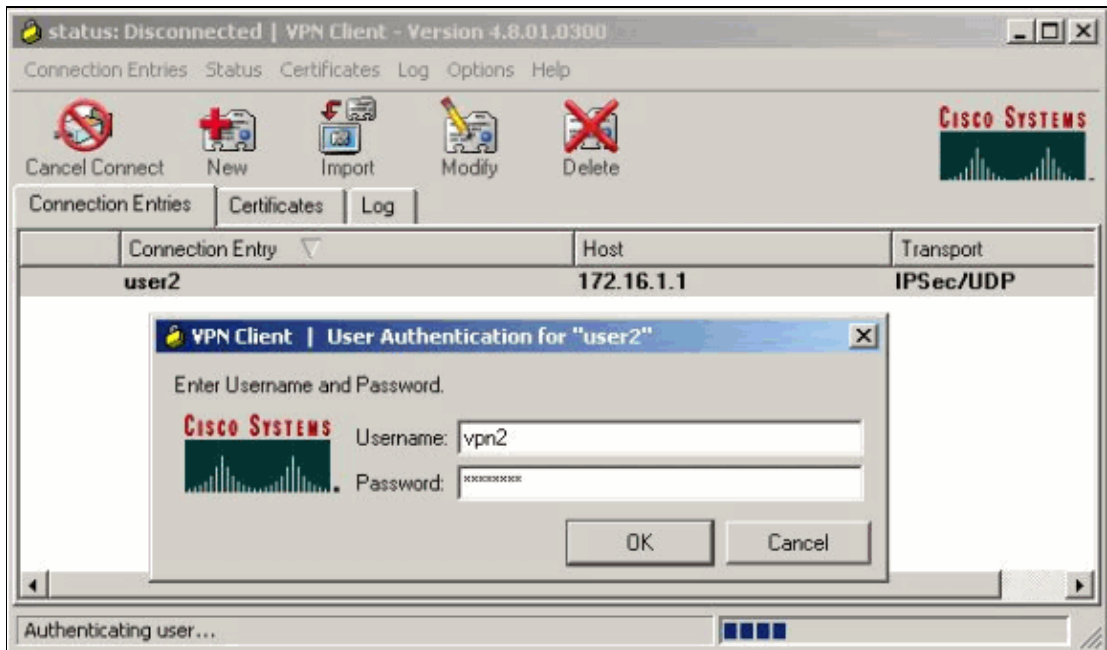
3. Enter the name of the Connection Entry along with a description. Enter the outside IP address of the PIX Firewall in the Host box. Then enter the Tunnel Group name (in this case, vpn2) and the preshared key and click **Save**.



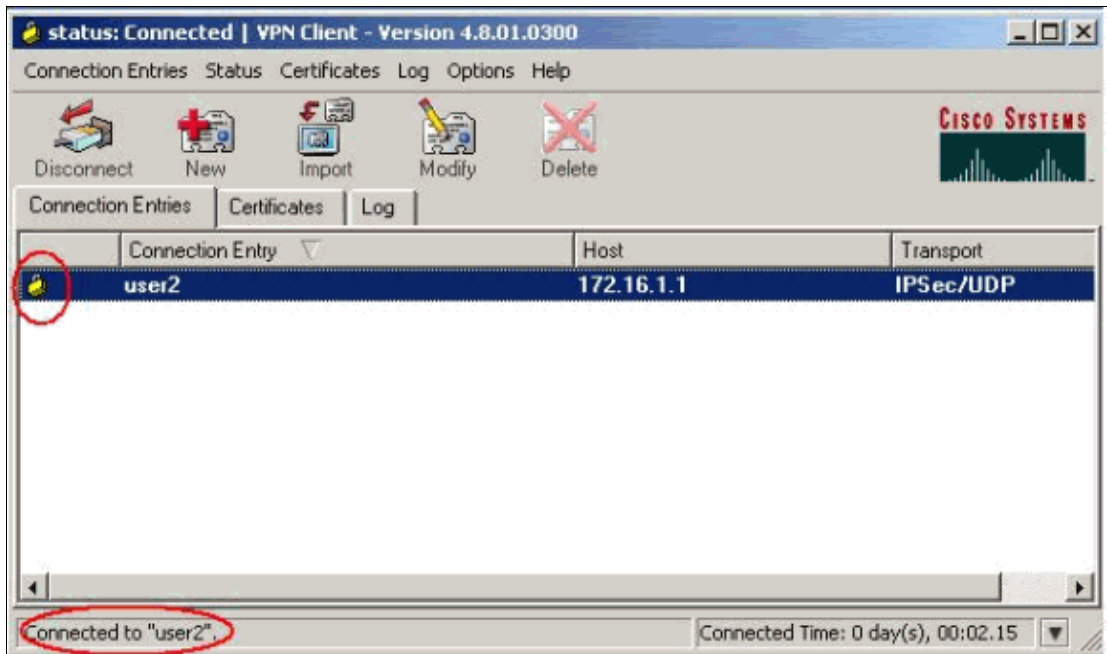
4. Click on the connection you would like to use and click **Connect** from the VPN Client main window.



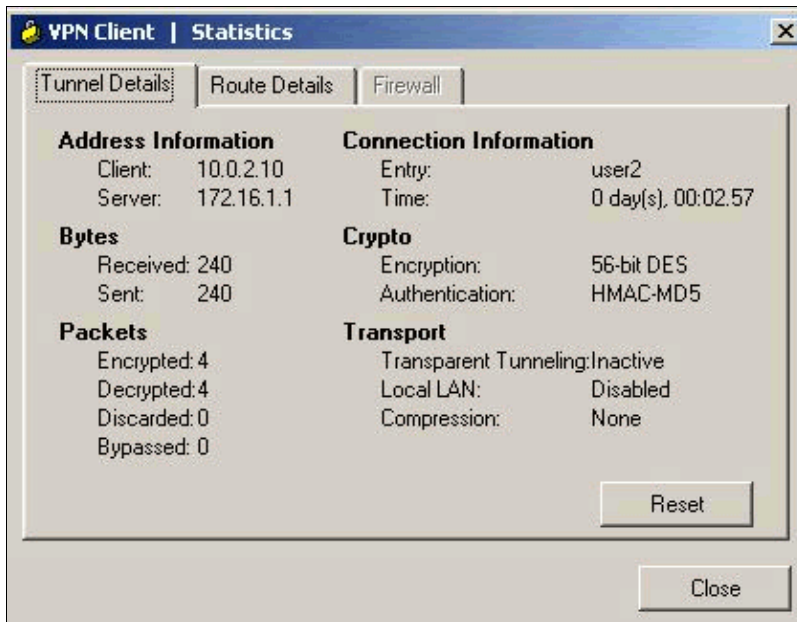
5. When prompted, enter the Username and Password information configured in the PIX and click **OK** to connect to the remote network.



6. The Cisco VPN Client gets connected with the PIX at the central site.

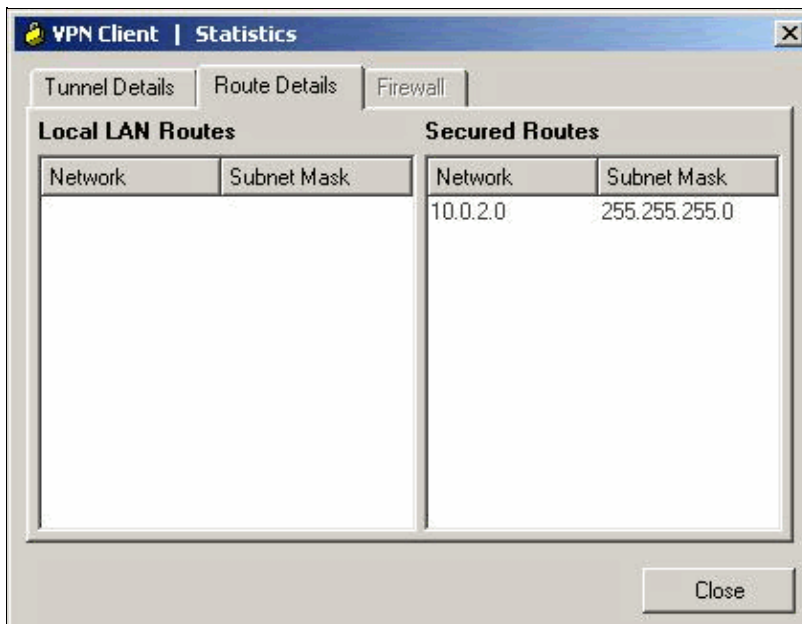


7. Select **Status** > **Statistics** to check the tunnel statistics of the Cisco VPN Client.



8. Select **Status > Statistics** and click **Route Details** to check the route details of the Cisco VPN Client.

The access list gets downloaded from the PIX to form the secured network connection for the network specified in the access list. The rest of the traffic directly enters into the Internet without encrypting into the tunnel.



Verify

This section provides information you can use to confirm your configuration is working properly.

The Output Interpreter Tool (registered customers only) (OIT) supports certain **show** commands. Use the OIT to view an analysis of **show** command output.

- **show crypto isakmp sa** Displays all current IKE Security Associations (SAs) at a peer.
- **show crypto ipsec sa** Displays the settings used by current SAs.

```
pix#show crypto ipsec sa
interface: outside
```

```

Crypto map tag: dynmap, seq num: 10, local addr: 172.16.1.1

local ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
remote ident (addr/mask/prot/port): (10.0.2.10/255.255.255.255/0/0)
current_peer: 10.0.0.2, username: vpn2
dynamic allocated peer ip: 10.0.2.10

#pkts encaps: 200, #pkts encrypt: 200, #pkts digest: 200
#pkts decaps: 201, #pkts decrypt: 201, #pkts verify: 201
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 200, #pkts comp failed: 0, #pkts decomp failed: 0
#send errors: 0, #recv errors: 0

local crypto endpt.: 172.16.1.1, remote crypto endpt.: 10.0.0.2

path mtu 1500, ipsec overhead 60, media mtu 1500
current outbound spi: 7233CD22

inbound esp sas:
  spi: 0x2F8C6D57 (797732183)
    transform: esp-des esp-md5-hmac
    in use settings ={RA, Tunnel, }
    slot: 0, conn_id: 1, crypto-map: dynmap
    sa timing: remaining key lifetime (sec): 28703
    IV size: 8 bytes
    replay detection support: Y
outbound esp sas:
  spi: 0x7233CD22 (1915997474)
    transform: esp-des esp-md5-hmac
    in use settings ={RA, Tunnel, }
    slot: 0, conn_id: 1, crypto-map: dynmap
    sa timing: remaining key lifetime (sec): 28701
    IV size: 8 bytes
    replay detection support: Y

```

Troubleshoot

This section provides information you can use to troubleshoot your configuration. Sample debug output is also shown.

Troubleshooting Commands

The Output Interpreter Tool (registered customers only) (OIT) supports certain **show** commands. Use the OIT to view an analysis of **show** command output.

Note: Refer to Important Information on Debug Commands and IP Security Troubleshooting – Understanding and Using debug Commands before you use **debug** commands.

- **debug crypto ipsec** Displays the IPsec negotiations of Phase 2.
- **debug crypto isakmp** Displays the ISAKMP negotiations of Phase 1.

Clear SAs

When a change is made to the tunnel configuration, be sure to clear the SAs. Use these commands in the privileged mode of the PIX:

- **clear [crypto] ipsec sa** Deletes the active IPsec SAs. The keyword 'crypto' is optional.
- **clear [crypto] isakmp sa** Deletes the active IKE SAs. The keyword 'crypto' is optional.

Sample debug Output

- PIX Firewall
- VPN Client 3.5 for Windows

PIX Firewall

```
PIX#debug crypto isakmp 7
pix# May 31 02:39:55 [IKEv1]: IP = 10.0.0.2, IKE_DECODE RECEIVED Message (msgid=
0) with payloads : HDR + SA (1) + KE (4) + NONCE (10) + ID (5) + VENDOR (13) + V
ENDOR (13) + VENDOR (13) + VENDOR (13) + NONE (0) total length : 8
48
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing SA payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing ke payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing ISA_KE payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing nonce payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing ID payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing VID payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, Received xauth V6 VID
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing VID payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, Received DPD VID
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing VID payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, Received Fragmentation VID
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, IKE Peer included IKE fragmentatio
n capability flags: Main Mode: True Aggressive Mode: False
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing VID payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, Received NAT-Traversal ver 02 VID
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, processing VID payload
May 31 02:39:55 [IKEv1 DEBUG]: IP = 10.0.0.2, Received Cisco Unity client VID
May 31 02:39:55 [IKEv1]: IP = 10.0.0.2, Connection landed on tunnel_group vpn2
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, processing IKE SA pa
yload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, IKE SA Proposal # 1,
Transform # 9 acceptable Matches global IKE entry # 2
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing ISAKMP
SA payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing ke payl
oad
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing nonce p
ayload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, Generating keys for
Responder...
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing ID payl
oad
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing hash pa
yload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, Computing hash for I
SAKMP
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing Cisco U
nity VID payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing xauth V
6 VID payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing dpd vid
payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing Fragmen
tation VID + extended capabilities payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing VID pay
load
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, Send Altiga/Cisco VP
N3000/Cisco ASA GW VID
May 31 02:39:55 [IKEv1]: IP = 10.0.0.2, IKE_DECODE SENDING Message (msgid=0) wit
h payloads : HDR + SA (1) + KE (4) + NONCE (10) + ID (5) + HASH (8) + VENDOR (13
) + VENDOR (13) + VENDOR (13) + VENDOR (13) + NONE (0) total lengt
h : 371
```

May 31 02:39:55 [IKEv1]: IP = 10.0.0.2, IKE_DECODE RECEIVED Message (msgid=0) with payloads : HDR + HASH (8) + NOTIFY (11) + VENDOR (13) + VENDOR (13) + NONE (0) total length : 120
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, processing hash payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, Computing hash for I SAKMP
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, processing notify payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, processing VID payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, Processing IOS/PIX Vendor ID payload (version: 1.0.0, capabilities: 00000408)
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, processing VID payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, Received Cisco Unity client VID
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing blank hash payload
May 31 02:39:55 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, constructing qm hash payload
May 31 02:39:55 [IKEv1]: IP = 10.0.0.2, IKE_DECODE SENDING Message (msgid=732d96ba) with payloads : HDR + HASH (8) + ATTR (14) + NONE (0) total length : 104
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, IKE_DECODE RECEIVED Message (msgid=732d96ba) with payloads : HDR + HASH (8) + ATTR (14) + NONE (0) total length : 84
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, process_attr(): Enter!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, IP = 10.0.0.2, Processing MODE_CFG Reply attributes.

*!--- User (vpn2) attributes from the tunnel group (vpn2) are downloaded
!--- to the VPN Client.*

May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE GetUserAttributes: primary DNS = cleared
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE GetUserAttributes: secondary DNS = cleared
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE GetUserAttributes: primary WINS = cleared
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE GetUserAttributes: secondary WINS = cleared
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE GetUserAttributes: split tunneling list = SPLIT-Tunnel-vpn2group
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE GetUserAttributes: IP Compression = disabled

!--- Split tunnel policy attributes are downloaded to the VPN Client (user2).

May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE GetUserAttributes: Split Tunneling Policy = Split Network
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, User (vpn2) authenticated.
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing blank hash payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing qm hash payload
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, IKE_DECODE SENDING Message (msgid=2b0b306) with payloads : HDR + HASH (8) + ATTR (14) + NONE (0) total length : 64
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, IKE_DECODE RECEIVED Message (msgid=2b0b306) with payloads : HDR + HASH (8) + ATTR (14) + NONE (0) total length : 60
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, process_attr(): Enter!

May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Processing cfg ACK attributes
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, IKE_DECODE RECEIVED Message (msgid=b983e913) with payloads : HDR + HASH (8) + ATTR (14) + NONE (0) total length : 194
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, process_attr(): Enter!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Processing cfg Request attributes
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for IPV4 address!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for IPV4 net mask!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for DNS server address!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for WINS server address!
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Received unsupported transaction mode attribute: 5
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for Banner!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for Save PW setting!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for Default Domain Name!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for Split Tunnel List!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for Split DNS!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for PFS setting!
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Received unknown transaction mode attribute: 28683
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for backup ip-sec peer list!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for Application Version!
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Client Type: WinNT Client Application Version: 4.8.01.0300
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for FWTYPE!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for DHCP hostname for DDNS is: tsweb-laptop!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, MODE_CFG: Received request for UDP Port!

!--- Assigns the private address to the remote user.

May 26 01:43:19 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Assigned private IP address 10.0.4.1 to remote user
May 26 01:43:19 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing blank hash payload
May 26 01:43:19 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing qm hash payload
May 26 01:43:19 [IKEv1]: IP = 10.0.0.2, IKE_DECODE SENDING Message (msgid=751f677d) with payloads : HDR + HASH (8) + ATTR (14) + NONE (0) total length : 189
May 26 01:43:19 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Delay Quick Mode processing, Cert/Trans Exch/RM DSID in progress
May 26 01:43:19 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Resume Quick Mode processing, Cert/Trans Exch/RM DSID completed

!--- ISAKMP (Phase 1) process is complete.

May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, PHASE 1 COMPLETED
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, Keep-alive type for this connection: DPD
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Starting phase 1 rekey timer: 82080000 (ms)
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, sending notify message
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing blank hash payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing qm hash payload
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, IKE_DECODE SENDING Message (msgid=1a3238c3) with payloads : HDR + HASH (8) + NOTIFY (11) + NONE (0) total length : 92
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, IKE_DECODE RECEIVED Message (msgid=a8bc0892) with payloads : HDR + HASH (8) + SA (1) + NONCE (10) + ID (5) + ID (5) + NONE (0) total length : 1026
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, processing hash payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, processing SA payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, processing nonce payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, processing ID payload
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Received remote Proxy Host data in ID Payload: Address 10.0.2.10, Protocol 0, Port 0
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, processing ID payload
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Received local IP Proxy Subnet data in ID Payload: Address 0.0.0.0, Mask 0.0.0.0, Protocol 0, Port 0
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, QM IsRekeyed old sa not found by addr
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE Remote Peer configured for crypto map: dynmap
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, processing IPsec SA payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IPsec SA Proposal # 14, Transform # 1 acceptable Matches global IPsec SA entry # 10
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE: requesting SPI!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE got SPI from key engine: SPI = 0xb9b5c50a
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, oakley constructing quick mode
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing blank hash payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing IPsec SA payload
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Overriding Initiator's IPsec rekeying duration from 2147483 to 28800 seconds
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing IPsec nonce payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing proxy ID
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Transmitting Proxy Id:
Remote host: 10.0.2.10 Protocol 0 Port 0
Local subnet: 0.0.0.0 mask 0.0.0.0 Protocol 0 Port 0
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Sending RESPONDER LIFETIME notification to Initiator
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, constructing qm hash payload
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, IKE_DECODE SENDING Message (msgid=a8bc0892) with payloads : HDR + HASH (8) + SA (1) + NONCE (10) + ID (5) + ID (5) + NOTIFY (11) + NONE (0) total length : 180
May 31 02:39:59 [IKEv1]: IP = 10.0.0.2, IKE_DECODE RECEIVED Message (msgid=a8bc0

892) with payloads : HDR + HASH (8) + NONE (0) total length : 52
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, processing hash payload
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, loading all IPSEC SAs
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Generating Quick Mode Key!
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Generating Quick Mode Key!
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Security negotiation complete for User (vpn2) Responder, Inbound SPI = 0xb9b5c50a, Outbound SPI = 0x691a0f90
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, IKE got a KEY_ADD msg for SA: SPI = 0x691a0f90
May 31 02:39:59 [IKEv1 DEBUG]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Pitcher: received KEY_UPDATE, spi 0xb9b5c50a
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Starting P2 Rekey timer to expire in 27360 seconds

*!--- Adds a static route for the client IP address in the PIX and
!--- the Phase 2 completed notification.*

May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, Adding static route for client address: 10.0.2.10
May 31 02:39:59 [IKEv1]: Group = vpn2, Username = vpn2, IP = 10.0.0.2, PHASE 2 C COMPLETED (msgid=a8bc0892)

PIX#debug crypto ipsec 7

pix# IPSEC: New embryonic SA created @ 0x02501E38,
SCB: 0x02501DA8,
Direction: inbound
SPI : 0x2F8C6D57
Session ID: 0x00000001
VPIF num : 0x00000001
Tunnel type: ra
Protocol : esp
Lifetime : 240 seconds
IPSEC: New embryonic SA created @ 0x02483448,
SCB: 0x02507930,
Direction: outbound
SPI : 0x7233CD22
Session ID: 0x00000001
VPIF num : 0x00000001
Tunnel type: ra
Protocol : esp
Lifetime : 240 seconds
IPSEC: Completed host ODSA update, SPI 0x7233CD22
IPSEC: Creating outbound VPN context, SPI 0x7233CD22
Flags: 0x00000005
SA : 0x02483448
SPI : 0x7233CD22
MTU : 1500 bytes
VCID : 0x00000000
Peer : 0x00000000
SCB : 0x02507930
Channel: 0x014A42F0
IPSEC: Completed outbound VPN context, SPI 0x7233CD22
VPN handle: 0x0245DBE8
IPSEC: New outbound encrypt rule, SPI 0x7233CD22
Src addr: 0.0.0.0
Src mask: 0.0.0.0
Dst addr: 10.0.2.10

```
Dst mask: 255.255.255.255
Src ports
  Upper: 0
  Lower: 0
  Op   : ignore
Dst ports
  Upper: 0
  Lower: 0
  Op   : ignore
Protocol: 0
Use protocol: false
SPI: 0x00000000
Use SPI: false
IPSEC: Completed outbound encrypt rule, SPI 0x7233CD22
  Rule ID: 0x025077F8
IPSEC: New outbound permit rule, SPI 0x7233CD22
  Src addr: 172.16.1.1
  Src mask: 255.255.255.255
  Dst addr: 10.0.0.2
  Dst mask: 255.255.255.255
  Src ports
    Upper: 0
    Lower: 0
    Op   : ignore
  Dst ports
    Upper: 0
    Lower: 0
    Op   : ignore
  Protocol: 50
  Use protocol: true
  SPI: 0x7233CD22
  Use SPI: true
IPSEC: Completed outbound permit rule, SPI 0x7233CD22
  Rule ID: 0x0245DC98
IPSEC: Completed host IBSA update, SPI 0x2F8C6D57
IPSEC: Creating inbound VPN context, SPI 0x2F8C6D57
  Flags: 0x00000006
  SA   : 0x02501E38
  SPI  : 0x2F8C6D57
  MTU  : 0 bytes
  VCID : 0x00000000
  Peer : 0x0245DBE8
  SCB  : 0x02501DA8
  Channel: 0x014A42F0
IPSEC: Completed inbound VPN context, SPI 0x2F8C6D57
  VPN handle: 0x024736F0
IPSEC: Updating outbound VPN context 0x0245DBE8, SPI 0x7233CD22
  Flags: 0x00000005
  SA   : 0x02483448
  SPI  : 0x7233CD22
  MTU  : 1500 bytes
  VCID : 0x00000000
  Peer : 0x024736F0
  SCB  : 0x02507930
  Channel: 0x014A42F0
IPSEC: Completed outbound VPN context, SPI 0x7233CD22
  VPN handle: 0x0245DBE8
IPSEC: Completed outbound inner rule, SPI 0x7233CD22
  Rule ID: 0x025077F8
IPSEC: Completed outbound outer SPD rule, SPI 0x7233CD22
  Rule ID: 0x0245DC98
```

```
!--- The IP address is assigned to the VPN Client
!--- from the pool (user2) of the PIX.
```

IPSEC: New inbound tunnel flow rule, SPI 0x2F8C6D57

Src addr: 10.0.2.10
Src mask: 255.255.255.255
Dst addr: 0.0.0.0
Dst mask: 0.0.0.0
Src ports
 Upper: 0
 Lower: 0
 Op : ignore
Dst ports
 Upper: 0
 Lower: 0
 Op : ignore
Protocol: 0
Use protocol: false
SPI: 0x00000000
Use SPI: false

IPSEC: Completed inbound tunnel flow rule, SPI 0x2F8C6D57

Rule ID: 0x02515C88

IPSEC: New inbound decrypt rule, SPI 0x2F8C6D57

Src addr: 10.0.0.2
Src mask: 255.255.255.255
Dst addr: 172.16.1.1
Dst mask: 255.255.255.255
Src ports
 Upper: 0
 Lower: 0
 Op : ignore
Dst ports
 Upper: 0
 Lower: 0
 Op : ignore
Protocol: 50
Use protocol: true
SPI: 0x2F8C6D57
Use SPI: true

IPSEC: Completed inbound decrypt rule, SPI 0x2F8C6D57

Rule ID: 0x022A7D10

*!--- Inbound rule for the VPN Client is downloaded from
!--- the split tunnel access list of the PIX.*

IPSEC: New inbound permit rule, SPI 0x2F8C6D57

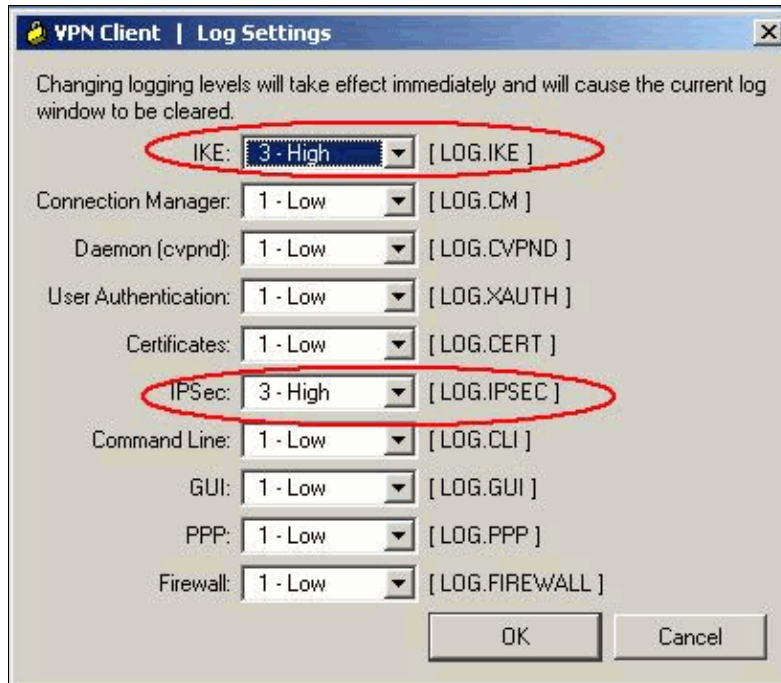
Src addr: 10.0.0.2
Src mask: 255.255.255.255
Dst addr: 172.16.1.1
Dst mask: 255.255.255.255
Src ports
 Upper: 0
 Lower: 0
 Op : ignore
Dst ports
 Upper: 0
 Lower: 0
 Op : ignore
Protocol: 50
Use protocol: true
SPI: 0x2F8C6D57
Use SPI: true

IPSEC: Completed inbound permit rule, SPI 0x2F8C6D57

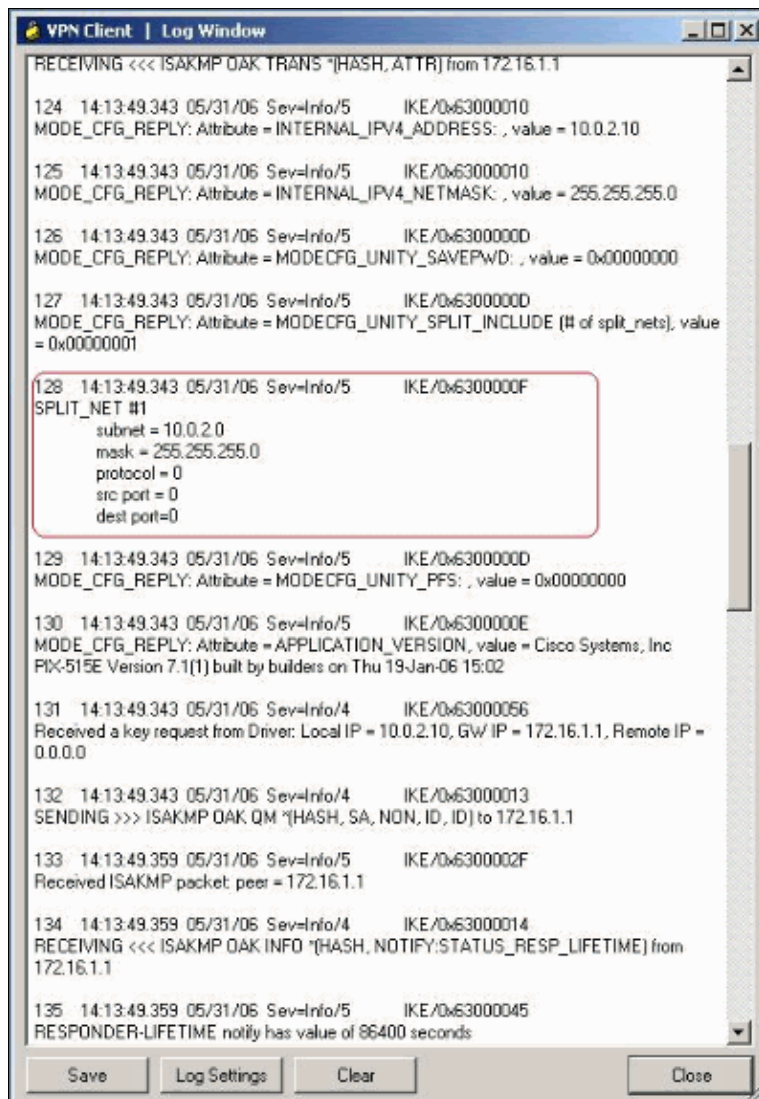
Rule ID: 0x02507788

VPN Client 4.8 for Windows

Select **Log > Log Settings** to enable the log levels in the Cisco VPN Client.



Select **Log > Log Windows** to view the log entries in the Cisco VPN Client. The split tunnel access lists are downloaded from the PIX for the vpn2 tunnel group user.



Related Information

- [Cisco PIX 500 Series Security Appliances](#)
- [Documentation for Cisco PIX Security Appliance OS Software](#)
- [Cisco Secure PIX Firewall Command References](#)
- [IPSec Negotiation/IKE Protocols Support Page](#)
- [Cisco VPN Client Support Page](#)
- [Requests for Comments \(RFCs\)](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Oct 02, 2009

Document ID: 69393