

Cisco Secure PIX Firewall 6.x and Cisco VPN Client 3.5 for Windows with Microsoft Windows 2000 and 2003 IAS RADIUS Authentication

Introduction

Before You Begin

- Conventions
- Prerequisites
- Components Used

Configure

- Network Diagram
- Configurations

Verify

Troubleshoot

- Troubleshooting Commands
- Sample debug Output

Related Information

Introduction

This sample configuration shows how to configure Cisco VPN Client version 3.5 for Windows and the Cisco Secure PIX Firewall for use with Microsoft Windows 2000 and 2003 Internet Authentication Service (IAS) RADIUS Server. Please refer to Microsoft – Checklist: Configuring IAS for dial-up and VPN access for further information on IAS.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

Before attempting this configuration, please ensure that you meet the following prerequisites:

The Cisco Secure PIX Firewall Software Release 6.0 supports VPN connections from the Cisco VPN Client 3.5 for Windows.

This sample configuration assumes that the PIX is already operating with the appropriate statics, conduits, or access lists. The current document does not intend to illustrate these basic concepts, but to show connectivity to the PIX from a Cisco VPN Client.

Components Used

The information in this document is based on the software and hardware versions below.

- PIX Firewall Software Release 6.1.1

Note: This was tested on PIX Software Release 6.1.1, but should work on all 6.x releases.

- Cisco VPN Client version 3.5 for Windows
- Windows 2000 and 2003 Server with IAS

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

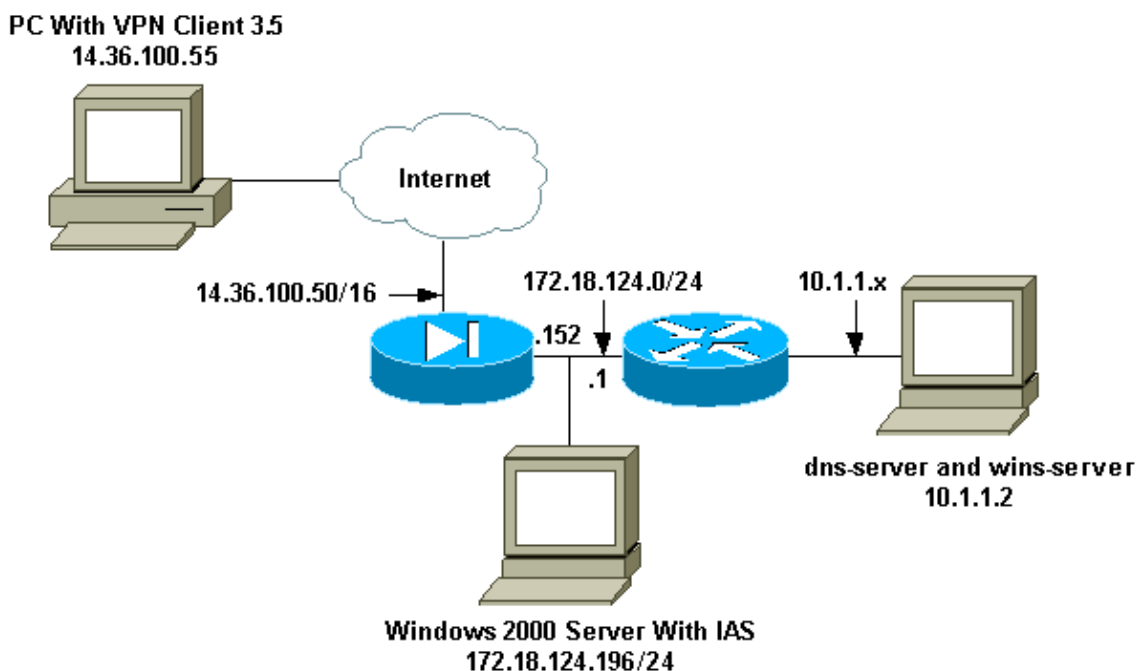
Configure

In this section, you are presented with the information to configure the features described in this document.

Note: To find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only).

Network Diagram

This document uses the network setup shown in the diagram below.



Configurations

This document uses the configurations shown below.

- Configuring the PIX Firewall
- Configuring the Cisco VPN Client 3.5 for Windows
- Configuring the Microsoft Windows 2000 Server with IAS
- Configuring the Microsoft Windows 2003 Server with IAS

Configuring the PIX Firewall

PIX Firewall

```
pixfirewall(config)# write terminal
Building configuration...
: Saved
:
PIX Version 6.1(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd 2KFQnbNIdI.2KYOU encrypted
hostname pixfirewall
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names

!--- Issue the access-list command to avoid
!--- Network Address Translation (NAT) on the IPSec packets.

access-list 101 permit ip 10.1.1.0 255.255.255.0 10.1.2.0
    255.255.255.0
pager lines 24
interface ethernet0 auto
interface ethernet1 auto
mtu outside 1500
mtu inside 1500
ip address outside 14.36.100.50 255.255.0.0
ip address inside 172.18.124.152 255.255.255.0
ip audit info action alarm
ip audit attack action alarm
ip local pool ippool 10.1.2.1-10.1.2.254
pdm history enable
arp timeout 14400
global (outside) 1 14.36.100.51

!--- Binding access list 101 to the NAT statement to avoid
!--- NAT on the IPSec packets.

nat (inside) 0 access-list 101
Nat (inside) 1 0.0.0.0 0.0.0.0 0 0
route outside 0.0.0.0 0.0.0.0 14.36.1.1 1
route inside 10.1.1.0 255.255.255.0 172.18.124.1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00
    rpc 0:10:00 h323 0:05:00 sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute

!--- Enable access to the TACACS+ and RADIUS protocols.

aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius

!--- Associate the partnerauth protocol to RADIUS.

aaa-server partnerauth protocol radius
```

```

aaa-server partnerauth (inside) host 172.18.124.196 cisco123
    timeout 5
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
floodguard enable

!--- Tell PIX to implicitly permit IPsec traffic.

sysopt connection permit-ipsec
no sysopt route dnat

!--- Configure a transform set that defines how the traffic will be protected.

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

!--- Create a dynamic crypto map and specify which
!--- transform sets are allowed for this dynamic crypto map entry.

crypto dynamic-map dynmap 10 set transform-set myset

!--- Add the dynamic crypto map set into a static crypto map set.

crypto map mymap 10 ipsec-isakmp dynamic dynmap

!--- Enable the PIX to launch the Xauth application on the VPN Client.

crypto map mymap client authentication partnerauth

!--- Apply the crypto map to the outside interface.

crypto map mymap interface outside

!--- IKE Policy Configuration.

isakmp enable outside
isakmp identity address
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

!--- IPsec group configuration for VPN Client.

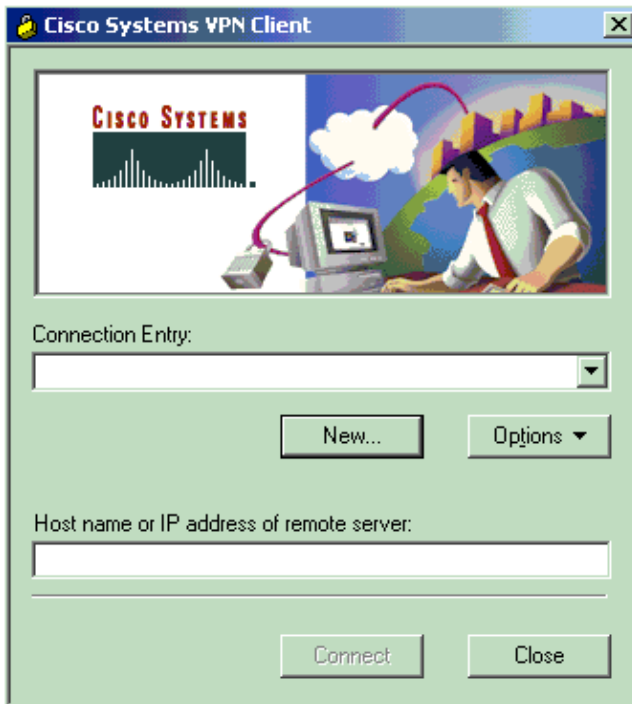
vpngroup vpn3000 address-pool ippool
vpngroup vpn3000 dns-server 10.1.1.2
vpngroup vpn3000 wins-server 10.1.1.2
vpngroup vpn3000 default-domain cisco.com
vpngroup vpn3000 idle-time 1800
vpngroup vpn3000 password *****
telnet timeout 5
ssh timeout 5
terminal width 80
Cryptochecksum:3f9e31533911b8a6bb5c0f06900c2dbc
: end
[OK]
pixfirewall(config)#

```

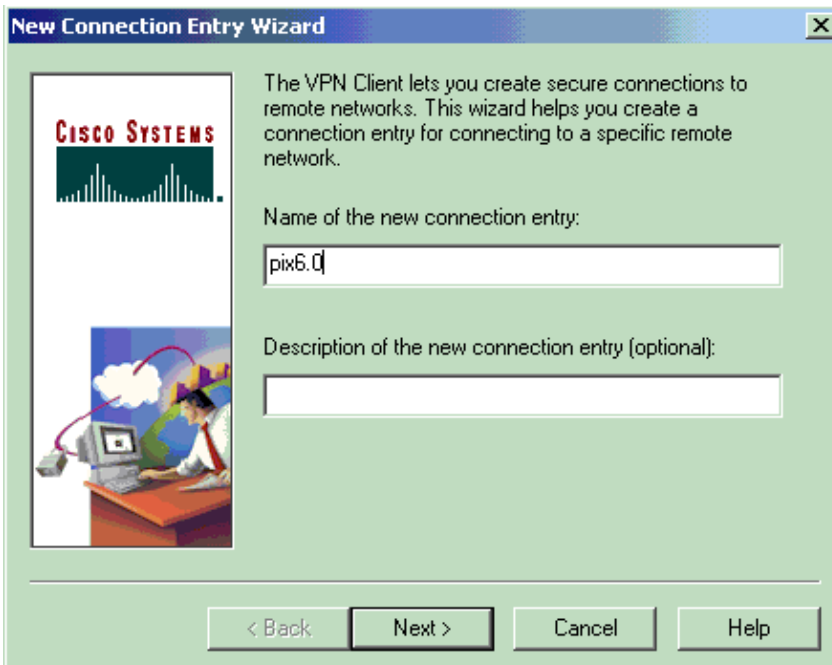
Configuring the Cisco VPN Client 3.5 for Windows

This section explains how to configure the Cisco VPN Client 3.5 for Windows.

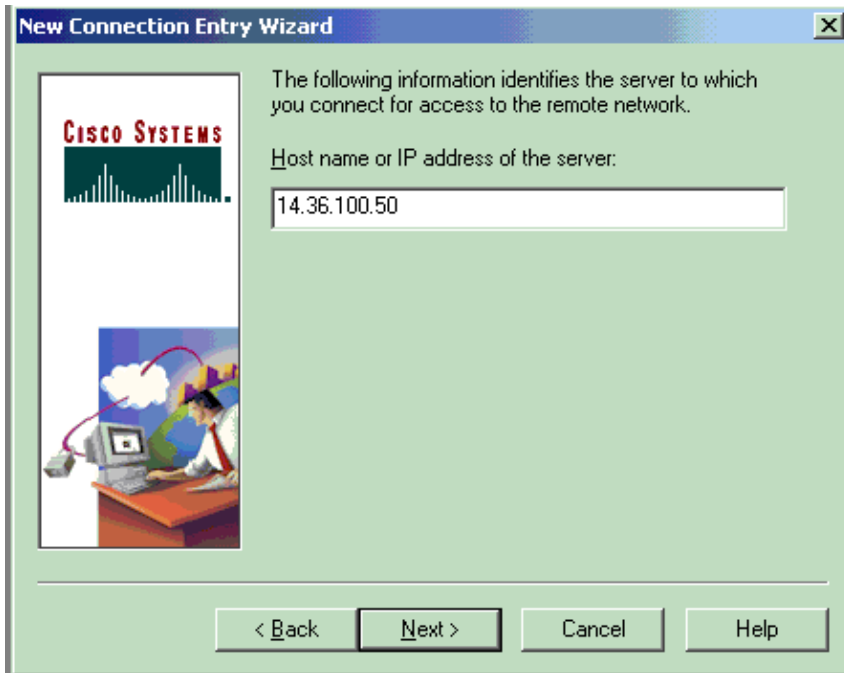
1. Launch the VPN Client and click **New** to create a new connection.



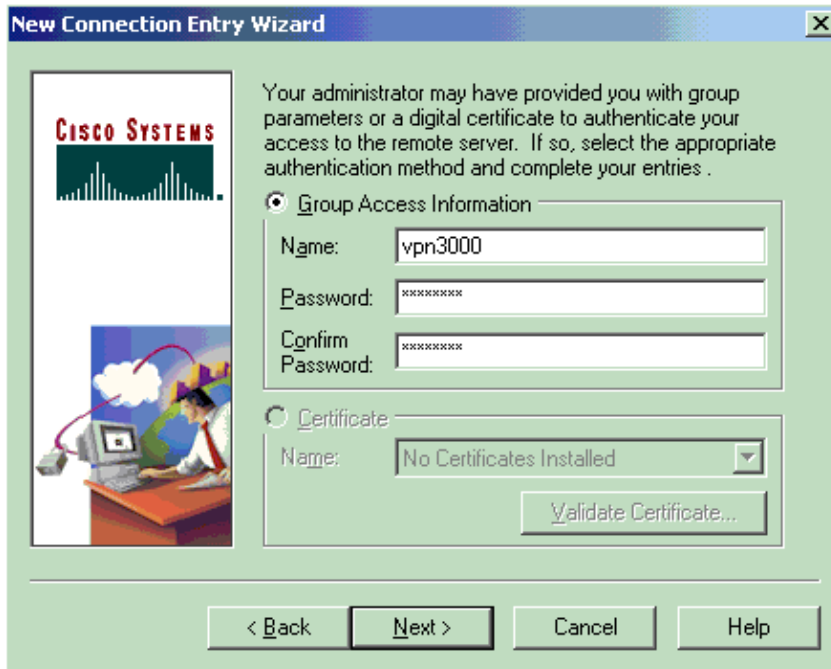
2. In the **Connection Entry** box, assign a name to your entry.



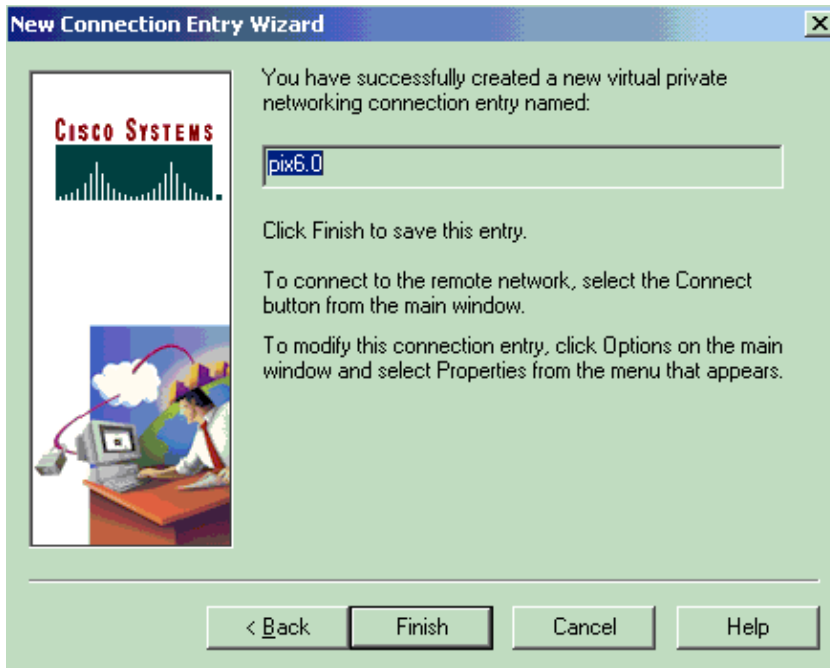
3. Enter the IP address of the public interface of the PIX.



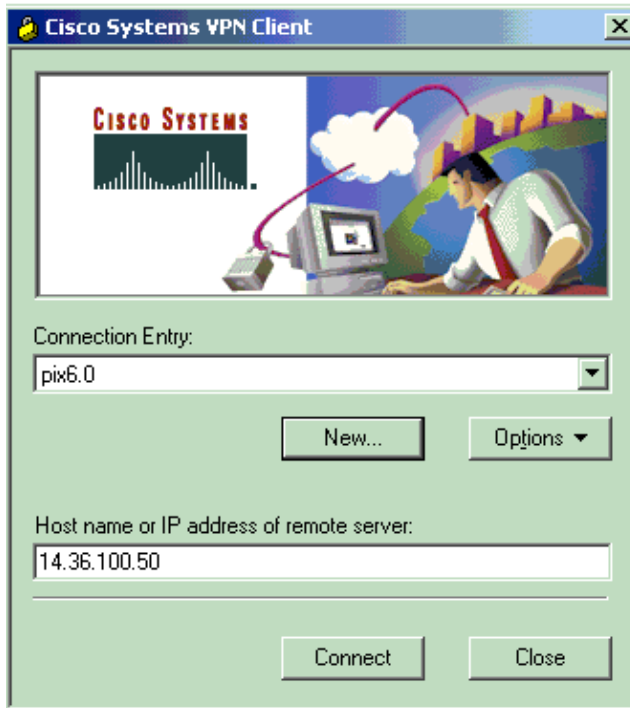
4. Under **Group Access Information**, enter the group name and the group password.



5. Click **Finish** to save the profile in the registry.



6. Click **Connect** to connect to the PIX.



Configuring the Microsoft Windows 2000 Server with IAS

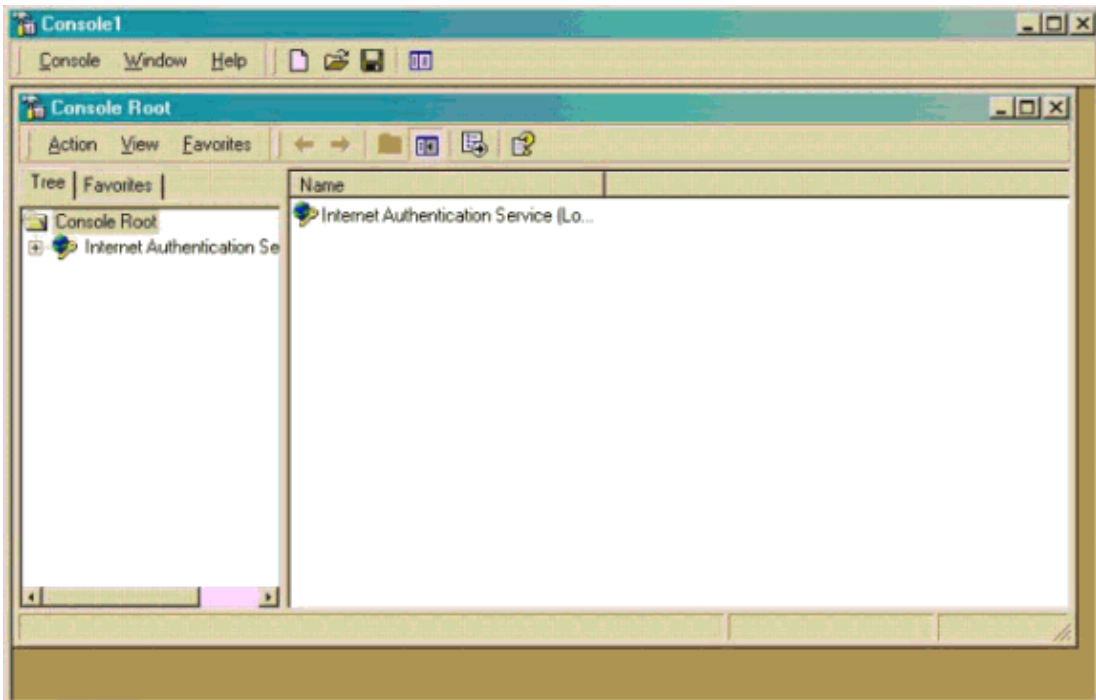
Follow these steps to configure Microsoft Windows 2000 server with IAS. This is a very basic setup to use a Windows 2000 IAS server for RADIUS authentication of VPN users. If you require a more complex design, please contact Microsoft for assistance.

Note: These steps assume that IAS has already been installed on the local machine. If not, please add this through **Control Panel > Add/Remove Programs**.

1. Launch the Microsoft Management Console by going to **Start > Run** and typing **mmc** and then

clicking **OK**.

2. To add the IAS service to this console, go to **Console > Add Remove Snap-In...**
3. Click **Add**. This will launch a new window with all of the available standalone snap-ins. Click **Internet Authentication Service (IAS)** and click **Add**.
4. Make sure **Local Computer** is selected and click **Finish**. Then click **Close**.
5. Notice that IAS is now added. Click **OK** to see that it has been added to the Console Root.



6. Expand the **Internet Authentication Service** and right-click on **Clients**. Click **New Client** and input a name. The choice of name really does not matter; it will be what you see in this view. Make sure to select **RADIUS** and click **Next**.
7. Fill in the **Client address** with the PIX interface address that the IAS server is connected to. Make sure to select **RADIUS Standard** and add the shared secret to match the command you entered on the PIX:

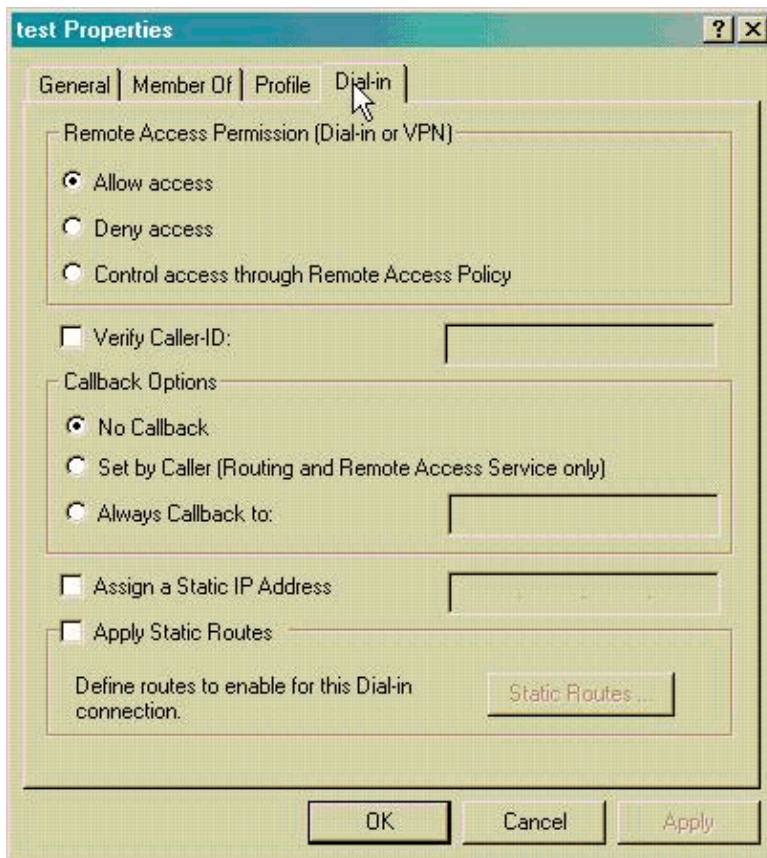
```
aaa-server partnerauth (inside) host 172.18.124.196 cisco123 timeout 5
```

Note: In this example, "cisco123" is the shared secret.

8. Click **Finish** to return to the Console Root.
9. Click **Remote Access Policies** in the left pane and double-click the policy labeled **Allow access if dial-in permission is enabled**.
10. Click **Edit Profile** and go to the Authentication tab. Under **Authentication Methods**, make sure only **Unencrypted Authentication (PAP, SPAP)** is checked.

Note: The VPN Client can only use this method for authentication.

11. Click **Apply** and then **OK** twice.
12. To modify the users to allow connection, go to **Console > Add/Remove Snap-in**. Click **Add** and then select the **Local Users and Groups snap-in**. Click **Add**. Make sure to select **Local Computer** and click **Finish**. Click **OK**.
13. Expand **Local User and Groups** and click the **Users** folder in the left pane. In the right pane, double-click the user you want to allow access.
14. Click the Dial-in tab and select **Allow Access** under **Remote Access Permission (Dial-in or VPN)**.



15. Click **Apply** and **OK** to complete the action. You can close the **Console Management** screen and save the session, if desired.
16. The users that you modified should now be able to access the PIX with the VPN Client 3.5. Please keep in mind that the IAS server only authenticates the user information. The PIX still does the group authentication.

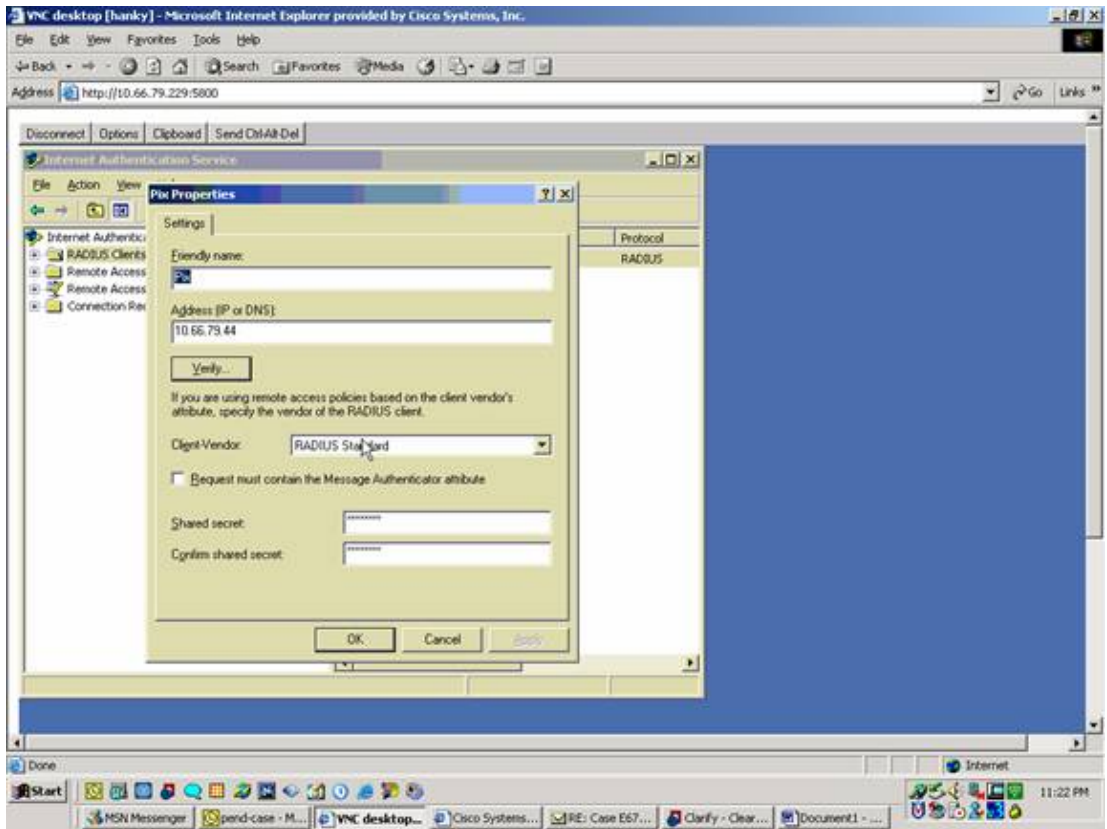
Configuring the Microsoft Windows 2003 Server with IAS

Follow these steps to configure Microsoft Windows 2003 server with IAS.

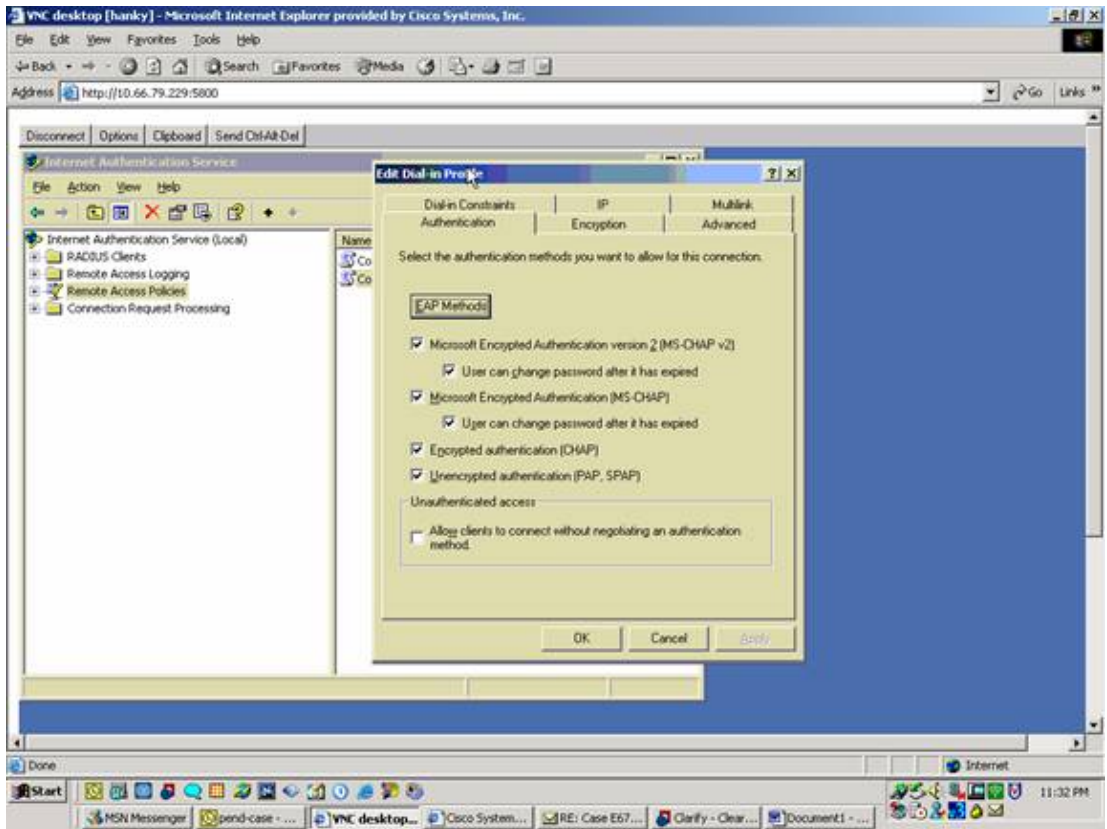
Note: These steps assume that IAS has already been installed on the local machine. If not, please add this through **Control Panel > Add/Remove Programs**.

1. Go to **Administrative Tools > Internet Authentication Service** and right-click on **RADIUS Client** to add a new RADIUS client. When you have typed the client information, click **OK**.

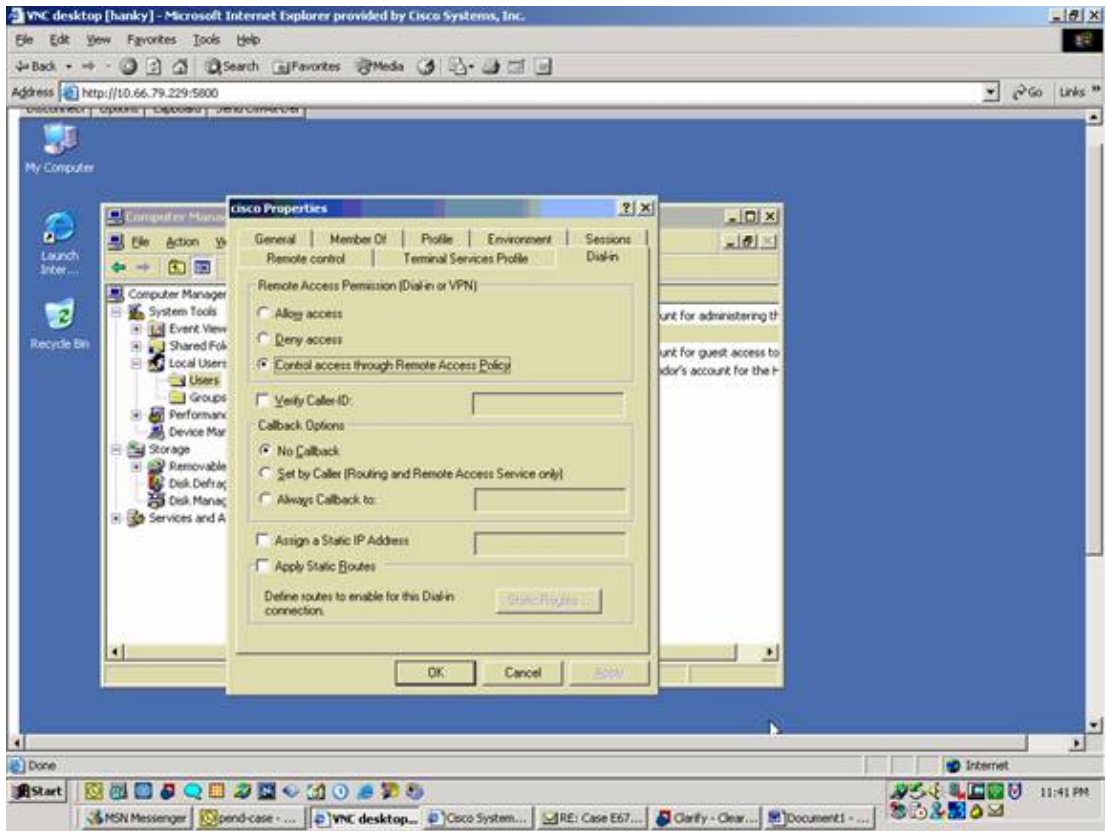
The example below shows a client named "Pix" with IP address 10.66.79.44. Client-Vendor is set to RADIUS Standard, and the shared secret is "cisco123."



2. Go to **Remote Access Policies**, right-click on **Connections to Other Access Servers**, and select **Properties**.
3. Ensure that the option for Grant Remote Access Permissions is selected.
4. Click **Edit Profile** and check the following settings.
 - ◆ On the Authentication tab, check **Unencrypted authentication (PAP, SPAP)**.
 - ◆ On the Encryption tab, ensure that the option for No Encryption is selected.Click **OK** when you are finished.



5. Add a user into the local computer account by going to **Administrative Tools > Computer Management > System Tools > Local Users and Groups**.. Right-click on **Users** and select **New Users**.
 6. Add user with Cisco password "cisco123" and check the following profile information.
 - ◆ On the **General** tab, ensure that the option for **Password Never Expired** is selected instead of the option for **User Must Change Password**.
 - ◆ On the **Dial-in** tab, select the option for **Allow access** (or leave default setting of **Control access through Remote Access Policy**).
- Click **OK** when you are finished.



Verify

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

Certain **show** commands are supported by the Output Interpreter Tool [🔗](#) (registered customers only), which allows you to view an analysis of **show** command output.

- **show crypto isakmp sa** – View all current IKE security associations (SAs) at a peer.
- **show crypto ipsec sa** – View the settings used by current security associations.

Troubleshoot

This section provides information you can use to troubleshoot your configuration. For additional information, refer to Troubleshooting the PIX to Pass Data Traffic on an Established IPsec Tunnel.

Troubleshooting Commands

Certain commands are supported by the Output Interpreter Tool [🔗](#) (registered customers only), which allows you to view an analysis of **show** command output.

Note: Before issuing **debug** commands, please see Important Information on Debug Commands and IP Security Troubleshooting – Understanding and Using debug Commands.

- **debug crypto ipsec** – View the IPsec negotiations of phase 2.
- **debug crypto isakmp** – View the ISAKMP negotiations of phase 1.
- **debug crypto engine** – View the traffic that is encrypted.

Sample debug Output

- PIX Firewall
- VPN Client 3.5 for Windows

PIX Firewall

```
pixfirewall(config)#
crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
VPN Peer: ISAKMP: Added new peer: ip:14.36.100.55 Total VPN Peers:1
VPN Peer: ISAKMP: Peer ip:14.36.100.55 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-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 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
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: Created a peer node for 14.36.100.55
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 14.36.100.55, dest 14.36.100.50
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_CONTACTIPSEC(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 14.36.100.55

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 14.36.100.55. ID = 3870616596
    (0xe6b4ec14)
crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
ISAKMP_TRANSACTION exchange
ISAKMP (0:0): processing transaction payload from 14.36.100.55.
    message ID = 84
ISAKMP: Config payload CFG_REPLY
return status is IKMP_ERR_NO_RETRANS
ISAKMP (0:0): initiating peer config to 14.36.100.55. ID = 3612718114
    (0xd755b422)
crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
ISAKMP_TRANSACTION exchange
ISAKMP (0:0): processing transaction payload from 14.36.100.55.
    message ID = 60
ISAKMP: Config payload CFG_ACK
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
ISAKMP_TRANSACTION exchange
ISAKMP (0:0): processing transaction payload from 14.36.100.55.
    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 14.36.100.55.
                ID = 3979868003
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_IDLE
ISAKMP (0): processing SA payload. message ID = 1527320241

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= 14.36.100.50, src= 14.36.100.55,
dest_proxy= 14.36.100.50/255.255.255.255/0/0 (type=1),
src_proxy= 10.1.2.1/255.255.255.255/0/0 (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 = 1527320241

ISAKMP (0): processing ID payload. message ID = 1527320241
ISAKMP (0): ID_IPV4_ADDR src 10.1.2.1 prot 0 port 0
ISAKMP (0): processing ID payload. message ID = 1527320241
ISAKMP (0): ID_IPV4_ADDR dst 14.36.100.50 prot 0 port
0IPSEC(key_engine): got a queue event...
IPSEC(spi_response): getting spi 0xf39c2217(4087095831) for SA
from 14.36.100.55 to 14.36.100.50 for prot 3

return status is IKMP_NO_ERROR

```

```

crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_IDLE
ISAKMP (0): processing SA payload. message ID = 3487980779

ISAKMP : Checking IPsec proposal 1

ISAKMP: transform 1, ESP_3DES
ISAKMP:  attributes in transform:
ISAKMP:      authenticator is HMAC-MD5
crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_AUTH_AWAIT
ISAKMP (0): Creating IPsec SAs
      inbound SA from 14.36.100.55 to 14.36.100.50
                (proxy 10.1.2.1 to 14.36.100.50)
      has spi 4087095831 and conn_id 1 and flags 4
      lifetime of 2147483 seconds
      outbound SA from 14.36.100.50 to 14.36.100.55
                (proxy 14.36.100.50 to 10.1.2.1)
      has spi 1929305241 and conn_id 2 and flags 4
      lifetime of 2147483 secondsIPSEC(key_engine): got a queue event...
IPSEC(initialize_sas): ,
      (key eng. msg.) dest= 14.36.100.50, src= 14.36.100.55,
      dest_proxy= 14.36.100.50/0.0.0.0/0/0 (type=1),
      src_proxy= 10.1.2.1/0.0.0.0/0/0 (type=1),
      protocol= ESP, transform= esp-des esp-md5-hmac ,
      lifedur= 2147483s and 0kb,
      spi= 0xf39c2217(4087095831), conn_id= 1, keysize= 0, flags= 0x4
IPSEC(initialize_sas): ,
      (key eng. msg.) src= 14.36.100.50, dest= 14.36.100.55,
      src_proxy= 14.36.100.50/0.0.0.0/0/0 (type=1),
      dest_proxy= 10.1.2.1/0.0.0.0/0/0 (type=1),
      protocol= ESP, transform= esp-des esp-md5-hmac ,
      lifedur= 2147483s and 0kb,
      spi= 0x72fedc99(1929305241), conn_id= 2, keysize= 0, flags= 0x4

VPN Peer: IPSEC: Peer ip:14.36.100.55 Ref cnt incremented to:2
      Total VPN Peers:1
VPN Peer: IPSEC: Peer ip:14.36.100.55 Ref cnt incremented to:3
      Total VPN Peers:1
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_AUTH_AWAIT
ISAKMP (0): Creating IPsec SAs
      inbound SA from 14.36.100.55 to 14.36.100.50
                (proxy 10.1.2.1 to 0.0.0.0)
      has spi 1791135440 and conn_id 3 and flags 4
      lifetime of 2147483 seconds
      outbound SA from 14.36.100.50 to 14.36.100.55
                (proxy 0.0.0.0 to 10.1.2.1)
      has spi 173725574 and conn_id 4 and flags 4
      lifetime of 2147483 secondsIPSEC(key_engine): got a queue event...
IPSEC(initialize_sas): ,
      (key eng. msg.) dest= 14.36.100.50, src= 14.36.100.55,
      dest_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
      src_proxy= 10.1.2.1/0.0.0.0/0/0 (type=1),
      protocol= ESP, transform= esp-des esp-md5-hmac ,
      lifedur= 2147483s and 0kb,
      spi= 0x6ac28ed0(1791135440), conn_id= 3, keysize= 0, flags= 0x4

```

```

IPSEC(initialize_sas): ,
  (key eng. msg.) src= 14.36.100.50, dest= 14.36.100.55,
  src_proxy= 0.0.0.0/0.0.0.0/0/0 (type=4),
  dest_proxy= 10.1.2.1/0.0.0.0/0/0 (type=1),
  protocol= ESP, transform= esp-des esp-md5-hmac ,
  lifedur= 2147483s and 0kb,
  spi= 0xa5ad786(173725574), conn_id= 4, keysize= 0, flags= 0x4

VPN Peer: IPSEC: Peer ip:14.36.100.55 Ref cnt incremented to:4
  Total VPN Peers:1
VPN Peer: IPSEC: Peer ip:14.36.100.55 Ref cnt incremented to:5
  Total VPN Peers:1
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 14.36.100.55, dest 14.36.100.50
ISAKMP (0): processing NOTIFY payload 36136 protocol 1
  spi 0, message ID = 3443334051
ISAKMP (0): received DPD_R_U_THERE from peer 14.36.100.55
ISAKMP (0): sending NOTIFY message 36137 protocol 1
return status is IKMP_NO_ERR_NO_TRANS

```

VPN Client 3.5 for Windows

```

193    19:00:56.073  01/24/02  Sev=Info/6      DIALER/0x63300002
Initiating connection.

194    19:00:56.073  01/24/02  Sev=Info/4      CM/0x63100002
Begin connection process

195    19:00:56.083  01/24/02  Sev=Info/4      CM/0x63100004
Establish secure connection using Ethernet

196    19:00:56.083  01/24/02  Sev=Info/4      CM/0x63100026
Attempt connection with server "14.36.100.50"

197    19:00:56.083  01/24/02  Sev=Info/6      IKE/0x6300003B
Attempting to establish a connection with 14.36.100.50.

198    19:00:56.124  01/24/02  Sev=Info/4      IKE/0x63000013
SENDING >>> ISAKMP OAK AG (SA, KE, NON, ID, VID, VID, VID)
to 14.36.100.50

199    19:00:56.774  01/24/02  Sev=Info/4      IPSEC/0x63700014
Deleted all keys

200    19:00:59.539  01/24/02  Sev=Info/5      IKE/0x6300002F
Received ISAKMP packet: peer = 14.36.100.50

201    19:00:59.539  01/24/02  Sev=Info/4      IKE/0x63000014
RECEIVING <<< ISAKMP OAK AG (SA, VID, VID, VID, KE, ID, NON, HASH)
from 14.36.100.50

202    19:00:59.539  01/24/02  Sev=Info/5      IKE/0x63000059
Vendor ID payload = 12F5F28C457168A9702D9FE274CC0100

203    19:00:59.539  01/24/02  Sev=Info/5      IKE/0x63000001
Peer is a Cisco-Unity compliant peer

204    19:00:59.539  01/24/02  Sev=Info/5      IKE/0x63000059
Vendor ID payload = AFCAD71368A1F1C96B8696FC77570100

205    19:00:59.539  01/24/02  Sev=Info/5      IKE/0x63000001
Peer supports DPD

206    19:00:59.539  01/24/02  Sev=Info/5      IKE/0x63000059

```

Vendor ID payload = 6D761DDC26ACECA1B0ED11FABBB860C4

207 19:00:59.569 01/24/02 Sev=Info/4 IKE/0x63000013
SENDING >>> ISAKMP OAK AG *(HASH, NOTIFY:STATUS_INITIAL_CONTACT)
to 14.36.100.50

208 19:00:59.569 01/24/02 Sev=Info/5 IKE/0x6300002F
Received ISAKMP packet: peer = 14.36.100.50

209 19:00:59.569 01/24/02 Sev=Info/4 IKE/0x63000014
RECEIVING <<< ISAKMP OAK TRANS *(HASH, ATTR) from 14.36.100.50

210 19:00:59.569 01/24/02 Sev=Info/4 CM/0x63100015
Launch xAuth application

211 19:01:04.236 01/24/02 Sev=Info/4 CM/0x63100017
xAuth application returned

212 19:01:04.236 01/24/02 Sev=Info/4 IKE/0x63000013
SENDING >>> ISAKMP OAK TRANS *(HASH, ATTR) to 14.36.100.50

213 19:01:04.496 01/24/02 Sev=Info/5 IKE/0x6300002F
Received ISAKMP packet: peer = 14.36.100.50

214 19:01:04.496 01/24/02 Sev=Info/4 IKE/0x63000014
RECEIVING <<< ISAKMP OAK TRANS *(HASH, ATTR) from 14.36.100.50

215 19:01:04.496 01/24/02 Sev=Info/4 CM/0x6310000E
Established Phase 1 SA. 1 Phase 1 SA in the system

216 19:01:04.506 01/24/02 Sev=Info/4 IKE/0x63000013
SENDING >>> ISAKMP OAK TRANS *(HASH, ATTR) to 14.36.100.50

217 19:01:04.516 01/24/02 Sev=Info/5 IKE/0x6300005D
Client sending a firewall request to concentrator

218 19:01:04.516 01/24/02 Sev=Info/5 IKE/0x6300005C
Firewall Policy: Product=Cisco Integrated Client, Capability=
(Centralized Policy Push).

219 19:01:04.516 01/24/02 Sev=Info/4 IKE/0x63000013
SENDING >>> ISAKMP OAK TRANS *(HASH, ATTR) to 14.36.100.50

220 19:01:04.586 01/24/02 Sev=Info/5 IKE/0x6300002F
Received ISAKMP packet: peer = 14.36.100.50

221 19:01:04.586 01/24/02 Sev=Info/4 IKE/0x63000014
RECEIVING <<< ISAKMP OAK TRANS *(HASH, ATTR) from 14.36.100.50

222 19:01:04.586 01/24/02 Sev=Info/5 IKE/0x63000010
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_ADDRESS: ,
value = 10.1.2.1

223 19:01:04.586 01/24/02 Sev=Info/5 IKE/0x63000010
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_DNS(1): ,
value = 10.1.1.2

224 19:01:04.586 01/24/02 Sev=Info/5 IKE/0x63000010
MODE_CFG_REPLY: Attribute = INTERNAL_IPV4_NBNS(1) (a.k.a. WINS)
: , value = 10.1.1.2

225 19:01:04.586 01/24/02 Sev=Info/5 IKE/0x6300000E
MODE_CFG_REPLY: Attribute = MODECFG_UNITY_DEFDOMAIN: ,
value = cisco.com

```

226    19:01:04.586  01/24/02  Sev=Info/4      CM/0x63100019
Mode Config data received

227    19:01:04.606  01/24/02  Sev=Info/5      IKE/0x63000055
Received a key request from Driver for IP address 14.36.100.50,
GW IP = 14.36.100.50

228    19:01:04.606  01/24/02  Sev=Info/4      IKE/0x63000013
SENDING >>> ISAKMP OAK QM *(HASH, SA, NON, ID, ID) to 14.36.100.50

229    19:01:04.606  01/24/02  Sev=Info/5      IKE/0x63000055
Received a key request from Driver for IP address 10.10.10.255,
GW IP = 14.36.100.50

230    19:01:04.606  01/24/02  Sev=Info/4      IKE/0x63000013
SENDING >>> ISAKMP OAK QM *(HASH, SA, NON, ID, ID) to 14.36.100.50

231    19:01:04.786  01/24/02  Sev=Info/4      IPSEC/0x63700014
Deleted all keys

232    19:01:05.948  01/24/02  Sev=Info/5      IKE/0x6300002F
Received ISAKMP packet: peer = 14.36.100.50

233    19:01:05.948  01/24/02  Sev=Info/4      IKE/0x63000014
RECEIVING <<< ISAKMP OAK QM *(HASH, SA, NON, ID, ID,
NOTIFY:STATUS_RESP_LIFETIME) from 14.36.100.50

234    19:01:05.948  01/24/02  Sev=Info/5      IKE/0x63000044
RESPONDER-LIFETIME notify has value of 28800 seconds

235    19:01:05.948  01/24/02  Sev=Info/5      IKE/0x63000045
RESPONDER-LIFETIME notify has value of 4608000 kb

236    19:01:05.948  01/24/02  Sev=Info/4      IKE/0x63000013
SENDING >>> ISAKMP OAK QM *(HASH) to 14.36.100.50

237    19:01:05.948  01/24/02  Sev=Info/5      IKE/0x63000058
Loading IPsec SA (Message ID = 0x5B090EB1 OUTBOUND SPI =
0xF39C2217 INBOUND SPI = 0x72FEDC99)

238    19:01:05.948  01/24/02  Sev=Info/5      IKE/0x63000025
Loaded OUTBOUND ESP SPI: 0xF39C2217

239    19:01:05.948  01/24/02  Sev=Info/5      IKE/0x63000026
Loaded INBOUND ESP SPI: 0x72FEDC99

240    19:01:05.948  01/24/02  Sev=Info/4      CM/0x6310001A
One secure connection established

241    19:01:05.988  01/24/02  Sev=Info/6      DIALER/0x63300003
Connection established.

242    19:01:06.078  01/24/02  Sev=Info/6      DIALER/0x63300008
MAPI32 Information - Outlook not default mail client

243    19:01:06.118  01/24/02  Sev=Info/5      IKE/0x6300002F
Received ISAKMP packet: peer = 14.36.100.50

244    19:01:06.118  01/24/02  Sev=Info/4      IKE/0x63000014
RECEIVING <<< ISAKMP OAK QM *(HASH, SA, NON, ID, ID,
NOTIFY:STATUS_RESP_LIFETIME) from 14.36.100.50

245    19:01:06.118  01/24/02  Sev=Info/5      IKE/0x63000044

```

```

RESPONDER-LIFETIME notify has value of 28800 seconds

246    19:01:06.118  01/24/02  Sev=Info/5      IKE/0x63000045
RESPONDER-LIFETIME notify has value of 4608000 kb

247    19:01:06.118  01/24/02  Sev=Info/4      IKE/0x63000013
SENDING >>> ISAKMP OAK QM *(HASH) to 14.36.100.50

248    19:01:06.118  01/24/02  Sev=Info/5      IKE/0x63000058
Loading IPsec SA (Message ID = 0xCFE65CEB OUTBOUND SPI =
0x6AC28ED0 INBOUND SPI = 0x0A5AD786)

249    19:01:06.118  01/24/02  Sev=Info/5      IKE/0x63000025
Loaded OUTBOUND ESP SPI: 0x6AC28ED0

250    19:01:06.118  01/24/02  Sev=Info/5      IKE/0x63000026
Loaded INBOUND ESP SPI: 0x0A5AD786

251    19:01:06.118  01/24/02  Sev=Info/4      CM/0x63100022
Additional Phase 2 SA established.

252    19:01:07.020  01/24/02  Sev=Info/4      IPSEC/0x63700010
Created a new key structure

253    19:01:07.020  01/24/02  Sev=Info/4      IPSEC/0x6370000F
Added key with SPI=0x17229cf3 into key list

254    19:01:07.020  01/24/02  Sev=Info/4      IPSEC/0x63700010
Created a new key structure

255    19:01:07.020  01/24/02  Sev=Info/4      IPSEC/0x6370000F
Added key with SPI=0x99dcfe72 into key list

256    19:01:07.020  01/24/02  Sev=Info/4      IPSEC/0x63700010
Created a new key structure

257    19:01:07.020  01/24/02  Sev=Info/4      IPSEC/0x6370000F
Added key with SPI=0xd08ec26a into key list

258    19:01:07.020  01/24/02  Sev=Info/4      IPSEC/0x63700010
Created a new key structure

259    19:01:07.020  01/24/02  Sev=Info/4      IPSEC/0x6370000F
Added key with SPI=0x86d75a0a into key list

260    19:01:15.032  01/24/02  Sev=Info/6      IKE/0x6300003D
Sending DPD request to 14.36.100.50, seq# = 152233542

261    19:01:15.032  01/24/02  Sev=Info/4      IKE/0x63000013
SENDING >>> ISAKMP OAK INFO *(HASH, NOTIFY:DPD_REQUEST)
to 14.36.100.50

262    19:01:15.032  01/24/02  Sev=Info/5      IKE/0x6300002F
Received ISAKMP packet: peer = 14.36.100.50

263    19:01:15.032  01/24/02  Sev=Info/4      IKE/0x63000014
RECEIVING <<< ISAKMP OAK INFO *(HASH, NOTIFY:DPD_ACK)
from 14.36.100.50

264    19:01:15.032  01/24/02  Sev=Info/5      IKE/0x6300003F
Received DPD ACK from 14.36.100.50, seq# received = 152233542,
seq# expected = 152233542

```

Related Information

- [PIX Support Page](#)
 - [Documentation for PIX Firewall](#)
 - [PIX Command References](#)
 - [RADIUS Support Page](#)
 - [RADIUS in IOS Documentation](#)
 - [Cisco VPN 3000 Series Concentrator Support Page](#)
 - [Cisco VPN 3000 Series Client Support Page](#)
 - [IPSec Support Page](#)
 - [Requests for Comments \(RFCs\)](#)
 - [Technical Support – Cisco Systems](#)
-

All contents are Copyright © 1992–2004 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.