IP Security Troubleshooting − Understanding and Using debug

Table of Contents

IP Security Troubleshooting − Understanding and Using debug Commands

Introduction

This document provides an explanation of common **debug** commands used in troubleshooting IPSec issues on both the Cisco IOS® Software and PIX. It is assumed that an attempt to configure IPSec is completed.

Before You Begin

Conventions

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

Prerequisites

There are no specific prerequisites for this document.

Components Used

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

- **Cisco IOS Software**
    - IPSec feature set
    - 56i − Indicates single Data Encryption Standard (DES) feature (on Cisco IOS 11.2 and later)
    - k2 − Indicates triple DES feature (on Cisco IOS 12.0 and later). Triple DES available on the Cisco 2600 series and later
- **PIX** − V5.0 and later. Need single or triple DES license key to activate.

Cisco IOS Software Debugs

The following sections explain the Cisco IOS Software debugs.

show crypto isakmp sa

This command shows the Internet Security Association Management Protocol (ISAKMP) Security Association (SA) built between peers.

```
dst       src         state     conn−id     slot
12.1.1.2  12.1.1.1    QM_IDLE    1           0
```

show crypto ipsec sa

This command shows IPSec SA built between peers. The encrypted tunnel is built between 12.1.1.1 and 12.1.1.2 for traffic going between networks 20.1.1.0 and 10.1.1.0. You can see the two Encapsulating Security Payload (ESP) SAs built inbound and outbound. Authentication Header (AH) is not used since there are no AH SAs. Below is an example of the show **crypto ipsec sa** command.

```
interface: FastEthernet0
  Crypto map tag: test, local addr. 12.1.1.1
 local  ident (addr/mask/prot/port): (20.1.1.0/255.255.255.0/0/0)
 remote ident (addr/mask/prot/port): (10.1.1.0/255.255.255.0/0/0)
 current_peer: 12.1.1.2
   PERMIT, flags={origin_is_acl,}
  #pkts encaps: 7767918, #pkts encrypt: 7767918, #pkts digest 7767918
  #pkts decaps: 7760382, #pkts decrypt: 7760382, #pkts verify 7760382
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0,
  #pkts decompress failed: 0, #send errors 1, #recv errors 0
   local crypto endpt.: 12.1.1.1, remote crypto endpt.: 12.1.1.2
   path mtu 1500, media mtu 1500
   current outbound spi: 3D3
   inbound esp sas:
      spi: 0x136A010F(325714191)
        transform: esp−3des esp−md5−hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 3442, flow_id: 1443, crypto map: test
        sa timing: remaining key lifetime (k/sec): (4608000/52)
        IV size: 8 bytes
        replay detection support: Y
     inbound ah sas:
     inbound pcp sas:
inbound pcp sas:
outbound esp sas:
   spi: 0x3D3(979)
    transform: esp−3des esp−md5−hmac ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 3443, flow_id: 1444, crypto map: test
    sa timing: remaining key lifetime (k/sec): (4608000/52)
    IV size: 8 bytes
    replay detection support: Y
outbound ah sas:
outbound pcp sas:
```

show crypto engine connection active

This command shows each Phase 2 SA built and the amount of traffic sent. Remember that Phase 2 SAs are uni−directional, so each SA will show traffic in one direction only (encryptions are outbound, decryptions are inbound).

debug crypto isakmp

The following is an example of the **debug crypto isakmp** command.

```
processing SA payload. message ID = 0
Checking ISAKMP transform against priority 1 policy
        encryption DES−CBC
        hash SHA
        default group 2
        auth pre−share
        life type in seconds
        life duration (basic) of 240
atts are acceptable. Next payload is 0
processing KE payload. message ID = 0
processing NONCE payload. message ID = 0
processing ID payload. message ID = 0
SKEYID state generated
processing HASH payload. message ID = 0
SA has been authenticated
processing SA payload. message ID = 800032287
```

debug crypto ipsec

This command shows the SRC and Dest IPSec tunnel endpoints. Src_proxy and dest_proxy are the client subnets. Two "sa created" messages should appear, one in each direction (four appear if doing ESP and AH). The following is an example of the **debug crypto ipsec** command.

```
Checking IPSec proposal 1transform 1, ESP_DES
attributes in transform:
        encaps is 1
        SA life type in seconds
        SA life duration (basic) of 3600
        SA life type in kilobytes
        SA life duration (VPI) of 0x0 0x46 0x50 0x0
HMAC algorithm is SHA
atts are acceptable.
Invalid attribute combinations between peers will show up as "atts
  not acceptable".
IPSEC(validate_proposal_request): proposal part #2,
(key eng. msg.) dest= 12.1.1.2, SRC= 12.1.1.1,
      dest_proxy= 10.1.1.0/0.0.0.0/0/0,
      src_proxy= 20.1.1.0/0.0.0.16/0/0,
      protocol= ESP, transform= esp-des esp-sha-hmac
      lifedur= 0s and 0kb,
      spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
IPSEC(key_engine): got a queue event...
IPSEC(spi_response): getting spi 203563166 for SA
      from 12.1.1.2 to 12.1.1.1 for prot 2
IPSEC(spi_response): getting spi 194838793 for SA
      from 12.1.1.2 to 12.1.1.1 for prot 3
IPSEC(key_engine): got a queue event...
IPSEC(initialize_sas): ,
  (key eng. msg.) dest= 12.1.1.2, SRC= 12.1.1.1,
      dest_proxy= 10.1.1.0/255.255.255.0/0/0,
      src_proxy= 20.1.1.0/255.255.255.0/0/0,
      protocol= ESP, transform= esp-des esp-sha-hmac
      lifedur= 3600s and 4608000kb,
      spi= 0xC22209E(203563166), conn_id= 3,
              keysize=0, flags= 0x4
IPSEC(initialize_sas): ,
  (key eng. msg.) SRC= 12.1.1.2, dest= 12.1.1.1,
      src_proxy= 10.1.1.0/255.255.255.0/0/0,
      dest_proxy= 20.1.1.0/255.255.255.0/0/0,
      protocol= ESP, transform= esp-des esp-sha-hmac
      lifedur= 3600s and 4608000kb,
      spi= 0xDED0AB4(233638580), conn_id= 6,
              keysize= 0, flags= 0x4
IPSEC(create_sa): sa created,
      (sa) sa_dest= 12.1.1.2, sa_prot= 50,
      sa_spi= 0xB9D0109(194838793),
      sa_trans= esp-des esp-sha-hmac , sa_conn_id= 5
 IPSEC(create_sa): sa created,
      (sa) sa_dest= 12.1.1.2, sa_prot= 50,
      sa_spi= 0xDED0AB4(233638580),
      sa_trans= esp-des esp-sha-hmac , sa_conn_id= 6
```

Sample Error Messages

The following sample error messages were generated from the debug commands listed below.

- **debug crypto ipsec**
- **debug crypto isakmp**
- **debug crypt engine**

Please see the Error Message Decoder (registered customers only) tool for more information.

Invalid Local Address

Below is an example of the "invalid local address" error message.

```
IPSEC(validate_proposal): invalid local address 12.2.6.2
ISAKMP (0:3): atts not acceptable. Next payload is 0
ISAKMP (0:3): SA not acceptable!
```

This error message is attributed to one of the following two common problems:

- The **crypto map map−name local−address interface−id** command causes the router to use an incorrect address as the identity because it forces the router to use a specified address.
- Crypto map is applied to the wrong interface or, it is not applied at all. Check the configuration to ensure that crypto map is applied to the correct interface.

IKE Message From X.X.X.X Failed Its Sanity Check or Is Malformed

The **debug** error below appears if the pre−shared key on the peers do not match. To fix this issue, check the pre−share key on both sides. Please see the Error Message Decoder (registered customers only) tool for more information.

```
1d00H:%CRPTO−4−IKMP_BAD_MESSAGE: IKE message from 150.150.150.1 failed its
 sanity check or is malformed
```

Processing of Main Mode Failed With Peer

Below is an example of the Main Mode error message. The failure of Main Mode suggests that the Phase I policy is not matching on both sides. Please see the Error Message Decoder (registered customers only) tool for more information.

```
1d00h: ISAKMP (0:1): atts are not acceptable. Next payload is 0
1d00h: ISAKMP (0:1); no offers accepted!
1d00h: ISAKMP (0:1): SA not acceptable!
1d00h: %CRYPTO−6−IKMP_MODE_FAILURE: Processing of Main Mode failed with
peer at 150.150.150.1
```

Verify that the Phase I policy is on both peers and ensure that all the attributes match, for example:

```
Encryption DES or 3DES
Hash MD5 or SHA
Diffie−Hellman Group 1 or 2
Authentcation {rsa−sig | rsa−encr | pre−share
```

Proxy Identities Not Supported

The message below appears in debugs if the access list for IPSec traffic does not match.

```
1d00h: IPSec(validate_transform_proposal): proxy identities not supported
1d00h: ISAKMP: IPSec policy invalidated proposal
1d00h: ISAKMP (0:2): SA not acceptable!
```

The access list on each peer should mirror each other (all entries should be reversible). The example below illustrates this point.

```
Peer A
access−list 150 permit ip 172.21.113.0 0.0.0.255 172.21.114.0 0.0.0.255
access−list 150 permit ip host 15.15.15.1 host 172.21.114.123
Peer B
access−list 150 permit ip 172.21.114.0 0.0.0.255 172.21.113.0 0.0.0.255
access−list 150 permit ip host 172.21.114.123 host 15.15.15.1
```

Transform Proposal Not Supported

The message below appears if the Phase II (IPSec) doesn't match on both sides. This most commonly occurs if there is a mismatch in the transform−set.

```
1d00h: IPSec (validate_proposal): transform proposal
   (port 3, trans 2, hmac_alg 2) not supported
1d00h: ISAKMP (0:2) : atts not acceptable. Next payload is 0
1d00h: ISAKMP (0:2) SA not acceptable
```

Verify that the transform−set matches on both sides:

```
crypto ipsec transform−set transform−set−name transform1
[transform2 [transform3]]
? ah−md5−hmac
? ah−sha−hmac
? esp−des
? esp−des and esp−md5−hmac
? esp−des and esp−sha−hmac
? esp−3des and esp−md5−hmac
? esp−3des and esp−sha−hmac
? comp−lzs
```

No Cert and No Keys With Remote Peer

The message below indicates that the peer address configured on the router is wrong or has changed. Verify that the peer address is correct and reachable.

```
1d00h: ISAKMP: No cert, and no keys (public or pre−shared) with
remote peer 150.150.150.2
```

Peer Address X.X.X.X Not Found

The error message below normally appears with the corresponding VPN 3000 Concentrator error message "Message: No proposal chosen(14)". This is a result of the connections being host−to−host. The router configuration had the IPSec proposals in an order such that the proposal chosen for the router matched the access list, but not the peer. The access list had a larger network that included the host that was intersecting traffic. To correct this, make the router proposal for this concentrator−to−router connection first in line, so that it matches the specific host first.

```
20:44:44: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) dest= 194.70.240.150, src= 198.174.236.6,
    dest_proxy= 10.0.0.76/255.255.255.255/0/0 (type=1),
    src_proxy= 198.174.238.203/255.255.255.255/0/0 (type=1),
    protocol= ESP, transform= esp−3des esp−md5−hmac ,
    lifedur= 0s and 0kb,
    spi= 0x0(0), conn_id= 0, keysize= 0, flags= 0x4
20:44:44: IPSEC(validate_transform_proposal):
   peer address 198.174.236.6 not found
```

IPSEC(initialize_sas): Invalid Proxy IDs

The error "21:57:57: IPSEC(initialize_sas): invalid proxy IDs" indicates that the received proxy identity does not match the configured proxy identity as per the access list. Check to ensure that they both match by checking the output from the **debug** command.

In the **debug** command output of the proposal request below, the corresponding access−list 103 permit ip 10.1.1.0 0.0.0.255 20.1.1.0 0.0.0.255 does not match. The access list is network−specific on one end and host−specific on the other.

```
21:57:57: IPSEC(validate_proposal_request): proposal part #1,
  (key eng. msg.) dest= 192.1.1.1, src= 192.1.1.2,
    dest_proxy= 10.1.1.1/255.255.255.0/0/0 (type=4),
    src_proxy= 20.1.1.1/255.255.255.0/0/0 (type=4)
```

Reserved Not Zero on Payload 5

This means that the ISAKMP keys do not match. You should rekey/reset in order to ensure accuracy.

PIX Debugs

show crypto isakmp sa

This command shows the ISAKMP SA built between peers.

```
dst       src        state     conn−id     slot
12.1.1.2  12.1.1.1   QM_IDLE    1           0
```

show crypto ipsec sa

This command shows IPSec SA built between peers. An encrypted tunnel is built between 12.1.1.1 and 12.1.1.2 for traffic going between networks 20.1.1.0 and 10.1.1.0. You can see the two ESP SAs built inbound and outbound. AH is not used since there are no AH SAs. The following is an example of the **show crypto ipsec sa** command.

```
interface: outside
    Crypto map tag: vpn, local addr. 12.1.1.1
   local  ident (addr/mask/prot/port): (20.1.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (12.1.1.2/255.255.255.255/0/0)
   current_peer: 10.2.1.1
   dynamic allocated peer ip: 12.1.1.2
     PERMIT, flags={}
    #pkts encaps: 345, #pkts encrypt: 345, #pkts digest 0
    #pkts decaps: 366, #pkts decrypt: 366, #pkts verify 0
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0,
    #pkts decompress failed: 0, #send errors 0, #recv errors 0
     local crypto endpt.: 12.1.1.1, remote crypto endpt.: 12.1.1.2
     path mtu 1500, ipsec overhead 56, media mtu 1500
     current outbound spi: 9a46ecae
     inbound esp sas:
      spi: 0x50b98b5(84646069)
        transform: esp−3des esp−md5−hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 1, crypto map: vpn
        sa timing: remaining key lifetime (k/sec): (460800/21)
        IV size: 8 bytes
        replay detection support: Y
     inbound ah sas:

     inbound pcp sas:

     outbound esp sas:
      spi: 0x9a46ecae(2588339374)
```

```
        transform: esp-3des esp-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2, crypto map: vpn
        sa timing: remaining key lifetime (k/sec): (460800/21)
        IV size: 8 bytes
        replay detection support: Y
    outbound ah sas:
```

debug crypto isakmp

This command displays debug information about IPSec connections and shows the first set of attributes being denied due to incompatibilities on both ends. The second attempt at matching (trying 3DES instead of DES and the Secure Hash Algorithm (SHA)) is acceptable, and the ISAKMP SA is built. This debug is also from a dial−up client which accepts an IP address (10.32.8.1) out of a local pool. Once the ISAKMP SA is built, the IPSec attributes are negotiated and are eventually found acceptable. The PIX then sets up the IPSec SAs as seen below. The following is an example of the **debug crypto isakmp** command.

```
crypto_isakmp_process_block: src 12.1.1.1, dest 12.1.1.2
OAK_AG exchange
ISAKMP (0): processing SA payload. message ID = 0
ISAKMP (0): Checking ISAKMP transform 1 against priority 1 policy
ISAKMP:      encryption DES−CBC
ISAKMP:      hash MD5
ISAKMP:      default group 1
ISAKMP:      auth pre−share
ISAKMP (0): atts are not acceptable. Next payload is 3
ISAKMP (0): Checking ISAKMP transform 3 against priority 1 policy
ISAKMP:      encryption 3DES−CBC
ISAKMP:      hash SHA
ISAKMP:      default group 1
ISAKMP:      auth pre−share
ISAKMP (0): atts are acceptable. Next payload is 3
ISAKMP (0): processing KE payload. message ID = 0
ISAKMP: Created a peer node for 12.1.1.2
OAK_QM exchange
ISAKMP (0:0): Need config/address
ISAKMP (0:0): initiating peer config to 12.1.1.2. ID = 2607270170 (0x9b67c91a)
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 12.1.1.2, dest 12.1.1.1
ISAKMP_TRANSACTION exchange
ISAKMP (0:0): processing transaction payload from 12.1.1.2.
   message ID = 2156506360
ISAKMP: Config payload CFG_ACK
ISAKMP (0:0): peer accepted the address!
ISAKMP (0:0): processing saved QM.
oakley_process_quick_mode:
OAK_QM_IDLE
ISAKMP (0): processing SA payload. message ID = 818324052
ISAKMP : Checking IPSec proposal 1
ISAKMP: transform 1, ESP_DES
ISAKMP:   attributes in transform:
ISAKMP:      authenticator is HMAC−MD5
ISAKMP:      encaps is 1
IPSEC(validate_proposal): transform proposal
   (prot 3, trans 2, hmac_alg 1) not supported
ISAKMP (0): atts not acceptable. Next payload is 0
ISAKMP : Checking IPSec proposal 2
ISAKMP: transform 1, ESP_3DES
ISAKMP:   attributes in transform:
ISAKMP:      authenticator is HMAC−MD5
ISAKMP:      encaps is 1
ISAKMP (0): atts are acceptable.
ISAKMP (0): processing NONCE payload. message ID = 818324052
ISAKMP (0): processing ID payload. message ID = 81
ISAKMP (0): ID_IPV4_ADDR src 10.32.8.1 prot 0 port 0
ISAKMP (0): processing ID payload. message ID = 81
ISAKMP (0): ID_IPV4_ADDR dst 12.1.1.1 prot 0 port 0
INITIAL_CONTACTIPSEC(key_engine): got a queue event...
```

debug crypto ipsec

This command displays **debug** information about IPSec connections.

```
IPSEC(key_engine): got a queue event...
IPSEC(spi_response): getting spi 0xd532efbd(3576885181) for SA
        from  12.1.1.2  to  12.1.1.1  for prot 3
return status is IKMP_NO_ERROR
crypto_isakmp_process_block: src 12.1.1.2, dest 12.1.1.1
OAK_QM exchange
oakley_process_quick_mode:
OAK_QM_AUTH_AWAIT
ISAKMP (0): Creating IPSec SAs
        inbound SA from  12.1.1.2  to  12.1.1.1
           (proxy 10.32.8.1 to  12.1.1.1.)
        has spi 3576885181 and conn_id 2 and flags 4
        outbound SA from  12.1.1.1  to  12.1.1.2
           (proxy 12.1.1.1 to 10.32.8.1)
        has spi 2749108168 and conn_id 1 and flags 4IPSEC(key_engine):
           got a queue event...
IPSEC(initialize_sas): ,
  (key eng. msg.) dest= 12.1.1.1, src= 12.1.1.2,
    dest_proxy= 12.1.1.1/0.0.0.0/0/0 (type=1),
    src_proxy= 10.32.8.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp−3des esp−md5−hmac ,
    lifedur= 0s and 0kb,
    spi= 0xd532efbd(3576885181), conn_id= 2, keysize= 0, flags= 0x4
IPSEC(initialize_sas): ,
  (key eng. msg.) src= 12.1.1.1, dest= 12.1.1.2,
    src_proxy= 12.1.1.1/0.0.0.0/0/0 (type=1),
    dest_proxy= 10.32.8.1/0.0.0.0/0/0 (type=1),
    protocol= ESP, transform= esp−3des esp−md5−hmac ,
    lifedur= 0s and 0kb,
    spi= 0xa3dc0fc8(2749108168), conn_id= 1, keysize= 0, flags= 0x4
return status is IKMP_NO_ERROR
```

Common Router−to−VPN Client Issues

Inability to Access Subnets Outside the VPN Tunnel − Split Tunneling

The following router configuration excerpt shows how to enable split tunneling for the VPN connections. The **access list 150** command is associated with the group as configured in the **crypto isakmp client configuration group hw−client−groupname** command. This allows the Cisco VPN Client to use the router to access an additional subnet that is not part of the VPN tunnel, without compromising the security of the IPSec connection. The tunnel is formed on the 172.168.0.128 network. Traffic flows unencrypted to devices not defined in the **access list 150** command, for example the Internet.

```
!
crypto isakmp client configuration group hw−client−groupname
 key hw−client−password
 dns 172.168.0.250 172.168.0.251
 wins 172.168.0.252 172.168.0.253
 domain cisco.com
 pool dynpool
 acl 150
!
!
access-list 150 permit ip 172.168.0.128 0.0.0.127 any
!
```

Common PIX−to−VPN Client Issues

This following sections address common problems encountered with configuring PIX to IPSec using VPN Client 3.x. The sample configurations for the PIX are based on version 6.x.

Traffic Does Not Flow After the Tunnel Is Established − Cannot Ping Inside the Network Behind PIX

This is a common problem associated with routing. Ensure that the PIX has a route for networks which are on the inside and not directly connected to the same subnet. Also, the inside network should have a route back to the PIX for the addresses in the client address pool.

An example is shown below.

```
!−−− Address of PIX inside interface.

ip address inside 10.1.1.1 255.255.255.240

!−−−Route to the networks which are on the inside segment,
!−−− the next hop is the router on the inside.

route inside 172.16.0.0 255.255.0.0 10.1.1.2 1

!−−−Pool of address defined on PIX from which it assigns addresses
!−−− to the VPN Client for the IPSec session

ip local pool mypool 10.1.2.1−10.1.2.254

!−−− On the internal router if the default gateway is not
!−−− the PIX inside interface, then the router should have route
!−−− for 10.1.2.0/24 network with next hop as the PIX inside interface
!−−− (as in Cisco IOS routers).

ip route 10.1.2.0 255.255.255.0 10.1.1.1
```

After the Tunnel Is Up, User Is Unable to Browse the Internet − Split Tunneling

The most common reason for this problem is that with the IPSec tunnel from the VPN Client to PIX, all the traffic is sent through the tunnel to the PIX firewall. The PIX functionality is such that it does not allow sending the traffic back to the interface where it was received, therefore the traffic destined to the Internet does not work. To fix this problem, use the **split tunneling** command. The idea behind this fix is that one only sends specific traffic through the tunnel and rest of the traffic goes directly to the Internet, not through the tunnel.

```
vpngroup vpn3000 split-tunnel 90
access−list 90 permit ip 10.1.1.0 255.255.255.0 10.1.2.0 255.255.255.0
access−list 90 permit ip 172.16.0.0 255.255.0.0 10.1.2.0 255.255.255.0
```

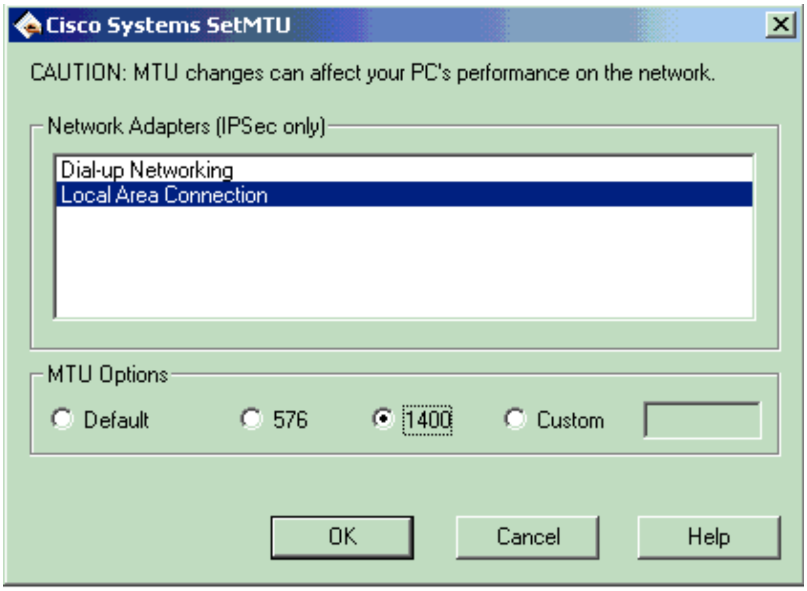
Note: The **vpngroup vpn3000 split−tunnel 90** command enables the split tunneling with **access−list number 90**. The **access−list 90** command defines which traffic flows through the tunnel, the rest of which is denied at the end of the access list. The access list should be the same for denying NAT on PIX.

After the Tunnel Is Up, Certain Applications Do Not Work −−− MTU Adjustment on Client

Sometimes after the tunnel is established, a user finds they can ping the machines on the network behind the PIX firewall, but are unable to use certain applications like Microsoft Outlook. A common problem is the maximum transfer unit (MTU) size of the packets. The IPSec header can be up to 50 to 60 bytes, which is added to the original packet. If the size of the packet becomes more than 1500, the default for Internet, the devices need to fragment it. This way, after adding the IPSec header, it is still under 1496, which is the maximum for IPSec.

Note: The VPN Client comes with an MTU adjust utility that allows the user to adjust MTU for the Cisco VPN Client. In the case of PPPoE client users, adjust MTU for the PPPoE adapter. Use the following procedure to adjust the MTU utility for the VPN Client:

1. Select **Start** > **Programs** > **Cisco System VPN Client** > **Set MTU**.
2. Select **Local Area Connection** and choose **1400**. Click **OK**.



Cisco Systems SetMTU

CAUTION: MTU changes can affect your PC's performance on the network.

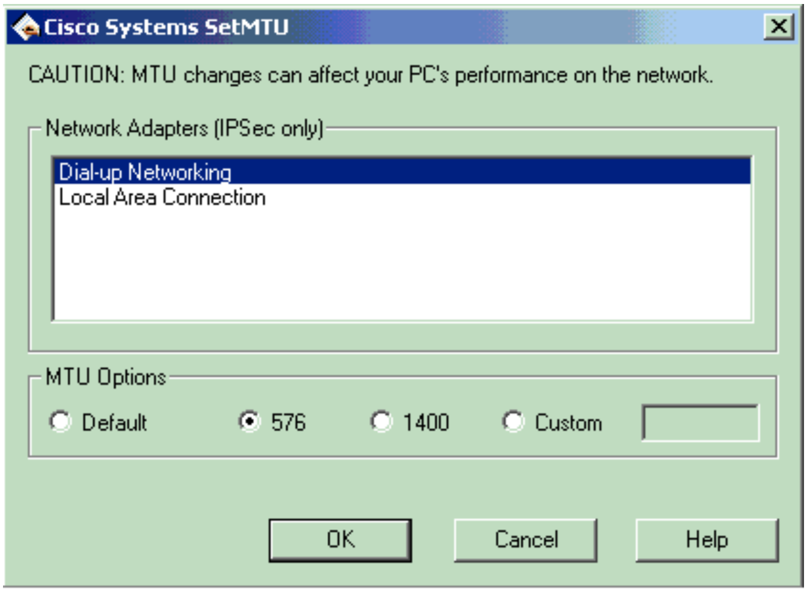
Network Adapters (IPSec only)

Dial-up Networking
Local Area Connection

MTU Options

○ Default ○ 576 ● 1400 ○ Custom

3. Repeat step 1 and select **Dial−up Networking** and choose **576**. Click **OK**.



Cisco Systems SetMTU

CAUTION: MTU changes can affect your PC's performance on the network.

Network Adapters (IPSec only)

Dial-up Networking
Local Area Connection

MTU Options

○ Default ● 576 ○ 1400 ○ Custom

Missing the sysopt Command

Use the **sysopt connection permit−ipsec** command in IPSec configurations on the PIX to permit IPSec traffic to pass through the PIX firewall without a check of **conduit** or **access−list** command statements. By default, any inbound session must be explicitly permitted by a **conduit** or **access−list** command statement. With IPSec protected traffic, the secondary access list check could be redundant. To enable IPSec authenticated/cipher inbound sessions to always be permitted, use the **sysopt connection permit−ipsec** command.

Related Information

- **IPSec Support Page**
- **An Introduction to IP Security (IPSec) Encryption**
- **PIX Support Page**
- **Documentation for PIX Firewall**
- **PIX Command Reference**
- **Error Message Decoder (registered customers only)**
- **Requests for Comments (RFCs)**
- **Technical Support − Cisco Systems**