Cisco Security Troubleshooting: Part II – Virtual Private Networks
Introduction
This white paper is the second in a three-part series, Cisco Security Troubleshooting. Each part in the series examines the challenge of implementing network security on equipment from Cisco Systems® while maintaining the connectivity requirements of the business or enterprise.

Cisco Security Troubleshooting
Part I: Connectivity Through ASA or PIX Firewall
Part II: Virtual Private Networks
Part III: Troubleshooting Intrusion Prevention Systems.

The focus on this second paper will primarily be on the effective use of CLI debug and show commands. To a lesser extent the ASDM Monitoring area will also be shown. We will examine three primary VPN implementation areas: 1) Site-to-Site (LAN-to-LAN) with IPSec; 2) Remote Access with IPSec; and 3) Remote Access with SSL.

Troubleshooting Scenario
The following scenario illustrates the use of the troubleshooting tools outlined above. If the names and IP Addresses look familiar to some of you, this is because the diagram represents a portion of the lab topology used in Cisco® Security classes offered by Global Knowledge.
IPSec VPNs

Before we illustrate the effective use of IPSec debugging tools, we need to discuss some basic principles of operation. These are enumerated below.

1. IPSec VPN sessions undergo two phases of negotiation which result in two types of Security Associations (SAs):
   a) ISAKMP1 SA (IKE2 Phase I result)
   b) 2 @ IPSec SAs - 1 for inbound traffic, 1 for outbound traffic (IKE Phase II result)

2. The negotiations above “begin” when “interesting” traffic arrives at a router or firewall configured for encryption. By “interesting,” we mean a packet matching what is usually specified in permit statements in ACLs associated with a configured crypto map.

3. The device into which the “interesting” packet arrives starts the negotiation process; hence it is commonly known as the “Initiator.” The router or firewall which is the IPSec partner or “peer” is known as the “Responder.”
4. The initiator will send ALL configured Phase I proposals and Phase II proposals configured specifically for the peer device. As a result, when possible, debugs should be run on the device which is the responder. This is the approach that will be taken in this paper.

Phase I Debugging – Policy Mismatch

In the previous scenario, we initiated a VPN from the **IOS Router** to the **ASA Firewall** by starting an HTTP connection from the **Site1-PC (10.20.10.10)** to the **Admin-PC (10.10.10.10)**. The output below shows what happens when there is no match between all of the policies presented by the router (the Initiator, in this example) to the ASA (the Responder). The syntax of the appropriate debug command for this is “**debug crypto isakmp <level>**” where $1 \leq \text{level} \leq 255$, and the default $\text{level} = 1$. Three debugs are shown below for comparison.

```
GKL-ASA(config)# debug crypto isakmp (Note: Level 1 in effect here, the default)
GKL-ASA(config)# Jan 19 21:32:09 [IKEv1]: IP = 200.200.20.2, Removing peer from peer table failed, no match!
GKL-ASA(config)# Jan 19 21:32:09 [IKEv1]: IP = 200.200.20.2, Error: Unable to remove PeerTblEntry

GKL-ASA(config)# debug crypto isakmp 128
GKL-ASA(config)# Jan 19 21:10:11 [IKEv1]: IP = 200.200.20.2, IKE_DECODE RECEIVED Message (msgid=0) with payloads : HDR + SA (1) + VENDOR (13) + VENDOR (13) + VENDOR (13) + NONE (0) total length : 288
Jan 19 21:10:11 [IKEv1 DEBUG]: IP = 200.200.20.2, processing SA payload
Jan 19 21:10:11 [IKEv1]: IP = 200.200.20.2, IKE_DECODE SENDING Message (msgid=0) with payloads : HDR + NOTIFY (11) + NONE (0) total length : 240
Jan 19 21:10:11 [IKEv1 DEBUG]: IP = 200.200.20.2, All SA proposals found unacceptable
Jan 19 21:10:11 [IKEv1]: IP = 200.200.20.2, Error processing payload: Payload ID: 1
Jan 19 21:10:11 [IKEv1 DEBUG]: IP = 200.200.20.2, IKE MM Responder FSM error history (struct &0xc8f6f8f0) <state>, <event>: MM_DONE, EV_ERROR-->MM_START, EV_RCV_MSG-->MM_START, EV_START_MM-->MM_START, EV_START_MM-->MM_START, EV_START_MM-->MM_START, EV_START_MM
Jan 19 21:10:11 [IKEv1 DEBUG]: IP = 200.200.20.2, IKE SA MM:38394b8a terminating: flags 0x01000002, refcnt 0, tuncnt 0
Jan 19 21:10:11 [IKEv1 DEBUG]: IP = 200.200.20.2, sending delete/delete with reason message
Jan 19 21:10:11 [IKEv1]: IP = 200.200.20.2, Removing peer from peer table failed, no match!
```
Jan 19 21:10:11 [IKEv1]: IP = 200.200.20.2, Error: Unable to remove PeerTblEntry

1 - Internet Security Association Key Management Protocol
2 - Internet Key Exchange

Note that even level = 128 does not display which policies are presented by the **IOS Router** during its Phase I negotiations. With the 8.0(3) code used for these traces on the ASA, trial and error shows that a value of **level** = 254 is necessary in order to see these policies.

debug crypto isakmp 254
GKL-ASA(config)#
RECV PACKET from 200.200.20.2
ISAKMP Header
  Initiator COOKIE: b6 db 43 60 18 19 d4 b8
  Responder COOKIE: 00 00 00 00 00 00 00 00
Next Payload: Security Association
  Version: 1.0
  Exchange Type: Identity Protection (Main Mode)
  Flags: (none)
  MessageID: 00000000
  Length: 288
Payload Security Association
  Next Payload: Vendor ID
  Reserved: 00
  Payload Length: 200
  DOI: IPsec
  Situation:(SIT.IDENTITY_ONLY)
Payload Proposal
  Next Payload: None
  Reserved: 00
  Payload Length: 188
  Proposal #: 1
  Protocol-Id: PROTO_ISAKMP
  SPI Size: 0
  # of transforms: 5
Payload Transform
  Next Payload: Transform
  Reserved: 00
  Payload Length: 36
Transform #: 1
Transform-Id: KEY_IKE
Reserved2: 0000
Encryption Algorithm: AES-CBC
Key Length: 256
Hash Algorithm: SHA1
Group Description: Group 2
Authentication Method: RSA Signatures
Life Type: seconds
Life Duration (Hex): a8 c0

Payload Transform
Next Payload: Transform
Reserved: 00
Payload Length: 36
Transform #: 2
Transform-Id: KEY_IKE
Reserved2: 0000
Encryption Algorithm: AES-CBC
Key Length: 128
Hash Algorithm: SHA1
Group Description: Group 2
Authentication Method: RSA Signatures
Life Type: seconds
Life Duration (Hex): a8 c0

Payload Transform
Next Payload: Transform
Reserved: 00
Payload Length: 36
Transform #: 3
Transform-Id: KEY_IKE
Reserved2: 0000
Encryption Algorithm: AES-CBC
Key Length: 128
Hash Algorithm: SHA1
Group Description: Group 2
Authentication Method: Preshared key
Life Type: seconds
Life Duration (Hex): a8 c0
Payload Transform
   Next Payload: Transform
   Reserved: 00
   Payload Length: 36
   Transform #: 4
   Transform-Id: KEY_IKE
   Reserved2: 0000
   Encryption Algorithm: 3DES-CBC
   Hash Algorithm: SHA1
   Group Description: Group 2
   Authentication Method: Preshared key
   Life Type: seconds
   Life Duration (Hex): 00 01 51 80

Payload Transform
   Next Payload: None
   Reserved: 00
   Payload Length: 36
   Transform #: 5
   Transform-Id: KEY_IKE
   Reserved2: 0000
   Encryption Algorithm: DES-CBC
   Hash Algorithm: SHA1
   Group Description: Group 1
   Authentication Method: RSA Signatures
   Life Type: seconds
   Life Duration (Hex): 00 01 51 80

Payload Vendor ID
   Next Payload: Vendor ID
   Reserved: 00
   Payload Length: 20
   Data (In Hex):
      43 9b 59 f8 ba 67 6c 4c 77 37 ae 22 ea b8 f5 82

Payload Vendor ID
   Next Payload: Vendor ID
   Reserved: 00
   Payload Length: 20
   Data (In Hex):
      7d 94 19 a6 53 10 ca 6f 2c 17 9d 92 15 52 9d 56

Payload Vendor ID
Next Payload: None
Reserved: 00
Payload Length: 20
Data (In Hex):

Data (In Hex):

90 cb 80 91 3e bb 69 6e 08 63 81 b5 ec 42 7b 1f

Jan 19 21:21:34 [IKEv1]: IP = 200.200.20.2, IKE_DECODE RECEIVED Message (msgid=0) with payloads : HDR + SA (1) + VENDOR (13) + VENDOR (13) + VENDOR (13) + NONE (0) total length : 288
Jan 19 21:21:34 [IKEv1 DEBUG]: IP = 200.200.20.2, processing SA payload
Jan 19 21:21:34 [IKEv1]: IP = 200.200.20.2, IKE_DECODE SENDING Message (msgid=0) with payloads : HDR + NOTIFY (11) + NONE (0) to- tal length : 240

ISAKMP Header

Initiator COOKIE: b6 db 43 60 18 19 d4 b8
Responder COOKIE: 08 69 12 43 2c c4 48 0e
Next Payload: Notification
Version: 1.0
Exchange Type: Informational
Flags: (none)
MessageID: 00000000
Length: 240
Payload Notification

Next Payload: None
Reserved: 00
Payload Length: 212
DOI: IPsec
Protocol-ID: Reserved
Spi Size: 0
Notify Type: NO_PROPOSAL_CHOSEN
Data:

Data:

0d 00 00 c8 00 00 00 01 00 00 00 01 00 00 00 bc
01 01 00 05 03 00 00 24 01 01 00 00 80 01 00 07
80 0e 01 00 80 02 00 02 80 04 00 02 80 03 00 03
80 0b 00 01 80 0c a8 c0 03 00 00 24 02 01 00 00
80 01 00 07 80 0e 00 80 80 02 00 02 80 04 00 02
80 03 00 03 80 0b 00 01 80 0c a8 c0 03 00 00 24
03 01 00 00 80 01 00 07 80 0e 00 80 80 02 00 02
80 04 00 02 80 03 00 01 80 0b 00 01 80 0c a8 c0
03 00 00 24 04 01 00 00 80 01 00 05 80 02 00 02
As seen above, the 5 proposals are seen in the debug (shown in **bold** font for emphasis) as well as the result of no match having been found. With this detailed level of debugging, the “capture to disk” feature of many terminal emulators would be highly recommended.

**Phase I Debugging – Preshared Key Mismatch**

Although the previous section would seem to suggest that debug crypto isakmp is not useful at its default level of 1, that level of debugging, as shown below, is able to pick up a mismatch between the preshared keys between the **IOS Router** and the **ASA Firewall**.

```
GKL-ASA(config)# debug crypto isakmp
GKL-ASA(config)# show debug
debug crypto isakmp enabled at level 1
2, Received encrypted Oakley Main Mode packet with invalid pay-
loads, MessID = 0
ERROR, had pro-
blems decrypting packet, probably due to mismatched pre-shared
key. Aborting
Jan 20 19:33:30 [IKEv1]: Group = 200.200.20.2, IP = 200.200.20.2,
Removing peer
```
from peer table failed, no match!

Jan 20 19:33:30 [IKEv1]: Group = 200.200.20.2, IP = 200.200.20.2,
Error: Unable
to remove PeerTblEntry

Phase II Debugging – Transform Set Mismatch

Unlike ISAKMP Phase I where ALL configured proposals are presented by the Initiator to the Responder, regardless of peer, the Phase II transform sets are specifically referenced in a set of crypto map entries for a particular peer. As with the Phase I debug traces, level 254 must be chosen in order to see the presented alternative transform sets. The output from the ASA CLI `debug crypto isakmp 254` command is shown on the next page. For the sake of clarity, the output for IKE Phase I has been omitted.

**ISAKMP Header**

- Initiator COOKIE: b6 db 43 60 ad cd dc 5d
- Responder COOKIE: 92 54 48 45 22 8f 86 ea
- Next Payload: Hash
- Version: 1.0
- Exchange Type: **Quick Mode**
- Flags: (Encryption)
- MessageID: 7AD1A886
- Length: 428
- Payload Hash
  - Next Payload: Security Association
    - Reserved: 00
    - Payload Length: 24
    - Data:
      - 1b 37 31 38 38 37 d6 eb 2d 78 5a b5 58 85 d2 fc
      - 66 d5 4d 01
- Payload Security Association
  - Next Payload: Nonce
    - Reserved: 00
    - Payload Length: 176
    - DOI: IPsec
    - Situation:(SIT_IDENTITY_ONLY)
- Payload Proposal
  - Next Payload: Proposal
    - Reserved: 00
    - Payload Length: 56
    - Proposal #: 1
    - Protocol-Id: PROTO_IPSEC_ESP
SPI Size: 4
# of transforms: 1
SPI: 3a 66 a7 5c
Payload Transform
   Next Payload: None
   Reserved: 00
   Payload Length: 44
   Transform #: 1
   Transform-Id: ESP_AES
   Reserved2: 0000
   Encapsulation Mode: Tunnel
   Life Type: Seconds
   Life Duration (Hex): 0e 10
   Life Type: Kilobytes
   Life Duration (Hex): 00 46 50 00
   Authentication Algorithm: SHA1
   Key Length: 256
   Group Description: Group 2
Payload Proposal
   Next Payload: Proposal
   Reserved: 00
   Payload Length: 56
   Proposal #: 2
   Protocol-Id: PROTO_IPSEC_ESP
SPI Size: 4
# of transforms: 1
SPI: 6a 26 4b 31
Payload Transform
   Next Payload: None
   Reserved: 00
   Payload Length: 44
   Transform #: 1
   Transform-Id: ESP_AES
   Reserved2: 0000
   Encapsulation Mode: Tunnel
   Life Type: Seconds
   Life Duration (Hex): 0e 10
   Life Type: Kilobytes
   Life Duration (Hex): 00 46 50 00
Authentication Algorithm: SHA1
Key Length: 128
Group Description: Group 2

Payload Proposal
Next Payload: None
Reserved: 00
Payload Length: 52
Proposal #: 3
Protocol-Id: PROTO_IPSEC_ESP
SPI Size: 4
# of transforms: 1
SPI: 94 c6 42 3c

Payload Transform
Next Payload: None
Reserved: 00
Payload Length: 40
Transform #: 1

Transform-Id: ESP_3DES
Reserved2: 0000
Encapsulation Mode: Tunnel
Life Type: Seconds
Life Duration (Hex): 0e 10
Life Type: Kilobytes
Life Duration (Hex): 00 46 50 00

Authentication Algorithm: SHA1
Group Description: Group 2

Payload Nonce
Next Payload: Key Exchange
Reserved: 00
Payload Length: 24
Data:
13 7e 94 d8 e9 45 06 f0 4b 43 ef 17 04 74 e1 de
91 39 92 ec

Payload Key Exchange
Next Payload: Identification
Reserved: 00
Payload Length: 132
Data:
9f f0 a6 5f 60 c4 e8 f1 cd c5 85 b9 95 dc 9a a4
Payload Identification
Next Payload: Identification
Reserved: 00
Payload Length: 16
ID Type: IPv4 Subnet (4)
Protocol ID (UDP/TCP, etc...): 0
Port: 0
ID Data: 10.20.10.0/255.255.255.0

Payload Identification
Next Payload: None
Reserved: 00
Payload Length: 16
ID Type: IPv4 Subnet (4)
Protocol ID (UDP/TCP, etc...): 0
Port: 0
ID Data: 10.10.0.0/255.255.0.0

Jan 20 20:02:26 [IKEv1]: IP = 200.200.20.2, IKE_DECODE RECEIVED Message (msgid=7ad1a886) with payloads : HDR + HASH (8) + SA (1) + NONCE (10) + KE (4) + ID (5) + ID (5) + NONE (0) total length : 416
Jan 20 20:02:26 [IKEv1 DEBUG]: Group = 200.200.20.2, IP = 200.200.20.2, processing hash payload
Jan 20 20:02:26 [IKEv1 DEBUG]: Group = 200.200.20.2, IP = 200.200.20.2, processing SA payload
Jan 20 20:02:26 [IKEv1 DEBUG]: Group = 200.200.20.2, IP = 200.200.20.2, processing nonce payload
Jan 20 20:02:26 [IKEv1 DEBUG]: Group = 200.200.20.2, IP = 200.200.20.2, processing ke payload
Jan 20 20:02:26 [IKEv1 DEBUG]: Group = 200.200.20.2, IP = 200.200.20.2, processing ID payload
Jan 20 20:02:26 [IKEv1 DECODE]: Group = 200.200.20.2, IP = 200.200.20.2, ID_IPV4_ADDR_SUBNET ID received--10.20.10.0--255.255.255.0
As stated previously, with the output here being even more verbose than with Phase I debugging (which was omitted here and occurs prior to the Phase II trace displayed here), the “Capture Text” feature of most terminal emulators is especially important.

Phase II Debugging – Crypto ACL Mismatch

This final example of IPSec Site-to-Site debugging illustrates the mismatch of the configured crypto access lists (also referred to as proxies). As can be seen from this trace, a level value as low as 2 sufficient for troubleshooting this problem.

debag crypto isakmp 2

GKL-ASA# Jan 20 20:54:18 [IKEv1]: IP = 200.200.20.2, Connection landed on tunnel_group 200.200.20.2
Jan 20 20:54:18 [IKEv1]: IP = 200.200.20.2, Connection landed on tunnel_group 200.200.20.2
Jan 20 20:54:18 [IKEv1]: Group = 200.200.20.2, IP = 200.200.20.2, Rejecting IPSec tunnel: no matching crypto map entry for remote proxy 10.20.10.0/255.255.255.0/0/0 local proxy 10.10.0.0/255.255.0.0/0/0 on interface outside
Jan 20 20:54:18 [IKEv1]: Group = 200.200.20.2, IP = 200.200.20.2, QM FSM error (P2 struct &0xc8ec57e0, mess id 0xe9b58563)!
Jan 20 20:54:18 [IKEv1]: IP = 200.200.20.2, Received encrypted packet with no matching SA, dropping

Note that the format of the crypto ACL sent from the IOS Router to the ASA is the following: IP Address / Mask / IP Protocol / Port (if applicable) with the remote source parameters listed first, and the local destination parameters listed second.

Phase II Failure with Phase I Success as Seen in ASDM

While the ASDM GUI does not have the ability of being used to actively debug IPSec VPN tunnel establishment failures, the screenshot on the next page shows the result of an ASA initiating a tunnel where Phase I succeeds, but Phase II fails. Note the IKE: 1 item under the VPN Tunnels section while next to it is IPSec: 0 indicating the Phase II failure.
Remote Access VPNs – Misconfigured Group Name

Unlike the Site-to-Site VPNs examined previously, Remote Access IPSec VPNs use a logical name (vs. an IP Address) to identify the tunnel-group. With this in mind, a trace is shown below for a situation where the configured Group Name on the client software does not match any configured Group Name on the ASA. Level 2 is chosen here because, surprisingly, the misconfigured Group Name on the client is not shown at any higher level.

```
GKL-ASA# debug crypto isakmp 2
GKL-ASA# Jan 20 22:23:07 [IKEv1]: Group = DefaultRAGroup, IP = 150.150.1.20, Rem
oving peer from peer table failed, no match!
Jan 20 22:23:07 [IKEv1]: Group = DefaultRAGroup, IP = 150.150.1.20, Error: Unabl
e to remove PeerTblEntry
```

Remote Access VPNs – Misconfigured Preshared Key

With this same level of debugging, a trace of a misconfigured preshared key is clearly shown. Note also that in this case the Group Name displayed is NOT the DefaultRAGroup but rather the configured one.
Remote Access VPNs – NAT Traversal (NAT-T)

This final set of traces having to do with Remote Access VPN troubleshooting using the software client pertains to situations where the client initiates the session from behind a NAT device. While the default setting of the client is to support IPSec over UDP (NAT-T) as its selected NAT Transparency Transport option, this is NOT a default setting on the ASA. This must be enabled under Configuration --> Remote Access VPN --> Network (Client) Access --> Advanced --> IPSec --> IKE Parameters. The consequence of forgetting to do this will be that the client will connect, but no encrypted traffic will flow because of the NAT incompatibility of ESP-encapsulated IPSec. Using the Site1-PC in our Lab Topology to initiate the Remote Access VPN, the traces below show the difference between the default (no NAT-T) and configured (enabled NAT-T).

No NAT-T:

debug crypto isakmp 5
GKL-ASA# Jan 21 15:05:17 [IKEv1]: IP = 200.200.20.52, IKE Peer included IKE fragmentation capability flags: Main Mode: True Aggressive Mode: False
Jan 21 15:05:17 [IKEv1]: IP = 200.200.20.52, Connection landed on tunnel_group IPsecRemoteAccess
Jan 21 15:05:17 [IKEv1 DEBUG]: Group = IPsecRemoteAccess, IP = 200.200.20.52, IKE SA Proposal # 1, Transform # 5 acceptable Matches global IKE entry # 1

NAT-T:

duggage crypto isakmp 5
GKL-ASA# Jan 21 11:11:35 [IKEv1 DEBUG]: IP = 200.200.20.52, IKE
Peer included IKE fragmentation capability flags: Main Mode: True  Aggressive Mode: False  

Jan 21 11:11:35 [IKEv1]: IP = 200.200.20.52, Connection landed on tunnel_group IPsecRemoteAccess 

Jan 21 11:11:35 [IKEv1 DEBUG]: Group = IPsecRemoteAccess, IP = 200.200.20.52, IKE SA Proposal # 1, Transform # 5 acceptable  


SSL VPNs 

Unlike the two-phase IPSec operation, SSL VPNs are implemented by Cisco as a relatively simple https (usually TCP port 443) connection via a web browser or SSL VPN Client (now called AnyConnect). Although similar CLI debug commands exist for SSL VPNs (referred to typically as WebVPN in most documentation), the output of “debug webvpn <level>” merely shows the username. Similarly, the more generic “debug vpn-session-db <level>” requires a value of 128 to show significant output. A trace is shown below for the adminjr user logging into the WebVPN interface on the ASA launched from the Outside-PC via https://200.200.20.12. 

GKL-ASA# debug vpn-sessiondb 128 

dump vpn-sessiondb enabled at level 128 

< - output omitted with Raw Hex values for clarity -> 

VPN-SESSION_DB in SESS_Mgmt_AddEntry p->........ 

Protocol = 11 

EncrAlg = 3 

HashAlg = 2 

ignoreAcct = 0 

CompAlg = 0 

SSOType = 0 

pfsGroup = 0 

IkeNegMode = 0 

EncapMode = 5 

AuthenModeIKE = 0 

AuthenModeSSL = 2 

AuthenModePPP = 0 

AuthenModeX = 0 

AuthorModeX = 0 

DiffHelmanGrp = 0 
<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>*TunnelGroupName</td>
<td>DefaultWEBVPNGroup</td>
</tr>
<tr>
<td>server_group_Id</td>
<td>0</td>
</tr>
<tr>
<td>RekeyTime</td>
<td>0</td>
</tr>
<tr>
<td>RekeyKBytes</td>
<td>0</td>
</tr>
<tr>
<td>pGetCounters</td>
<td>0x8ca10b0</td>
</tr>
<tr>
<td>pClearCounters</td>
<td>0x8ca180</td>
</tr>
<tr>
<td>pGetfSessData</td>
<td>0x8ca11f0</td>
</tr>
<tr>
<td>IdleTime</td>
<td>30</td>
</tr>
<tr>
<td>ConnectTime</td>
<td>0</td>
</tr>
<tr>
<td>pKill</td>
<td>0x8ca2d30</td>
</tr>
<tr>
<td>*handle</td>
<td>0x4e000</td>
</tr>
<tr>
<td>publicIpAddr</td>
<td>150.150.1.20</td>
</tr>
<tr>
<td>LocAddrType</td>
<td>0</td>
</tr>
<tr>
<td>LocProxyAddr1</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>LocProxyAddr2</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>LocProxyProtocol</td>
<td>0x0</td>
</tr>
<tr>
<td>LocProxyPort</td>
<td>0x0</td>
</tr>
<tr>
<td>RemAddrType</td>
<td>0</td>
</tr>
<tr>
<td>RemProxyAddr1</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>RemProxyAddr2</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>RemProxyProtocol</td>
<td>0x0</td>
</tr>
<tr>
<td>RemProxyPort</td>
<td>0x0</td>
</tr>
<tr>
<td>assignedIpAddr</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>assignedIpv6Addr</td>
<td>::</td>
</tr>
<tr>
<td>hubInterface</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>WINSServer-&gt;server_type</td>
<td>0</td>
</tr>
<tr>
<td>WINSServer-&gt;server_count</td>
<td>0</td>
</tr>
<tr>
<td>WINSServer-&gt;server_addr_array[0]</td>
<td>0x0</td>
</tr>
<tr>
<td>DNSServer-&gt;server_type</td>
<td>0</td>
</tr>
<tr>
<td>DNSServer-&gt;server_count</td>
<td>0</td>
</tr>
<tr>
<td>DNSServer-&gt;server_addr_array[0]</td>
<td>0x0</td>
</tr>
<tr>
<td>*UserName</td>
<td>adminjr</td>
</tr>
<tr>
<td>*ClientOSVendor</td>
<td></td>
</tr>
<tr>
<td>*ClientOSVersion</td>
<td></td>
</tr>
<tr>
<td>*ClientVendor</td>
<td>Web Browser</td>
</tr>
<tr>
<td>*ClientVersion</td>
<td>Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.8.1) Gecko/20061010 Firefox</td>
</tr>
<tr>
<td>InstId</td>
<td>0</td>
</tr>
<tr>
<td>TcpSrcPort</td>
<td>0</td>
</tr>
</tbody>
</table>
TcpDstPort = 443
UdpSrcPort = 0
UdpDstPort = 0
filterId = 0
*aclId =
ipv6filterId = 0
*ipv6aclId =
vcaSession = 0
sessIndex = 0x4e000

Note that the user **adminjr** and tunnel group **(DefaultWEBVPNGroup)** have been shown in **BOLD** font for emphasis; however, from this output, we do not know which group policy applies to this user. For a better idea of the user-specific attributes for adminjr, we need to examine both the **Monitoring** and **Configuration** sections of the ASDM interface. Note below the view of the user **adminjr** under **Monitoring --> VPN --> VPN Statistics --> Sessions**.

**Screen 2**

As shown above, not only is the tunnel group displayed, but the applied group policy can be seen as well. Note that the **DefaultWEBVPNGroup** connection profile is the tunnel group here; this option defaults as being not user-selectable at login. This can be changed by going to the **Connection Profiles** portion of the **Configuration** tab for **Remote Access VPN** for **Clientless SSL VPN Access** as shown by the screenshot on the next page. If the “Allow user to select connection, identified by alias in the table above, at login page” box is checked, the user has an option to select any connection profiles for which the “Aliases” field is not blank.
Finally, the screen below shows how the type of VPN access for the adminjr user can be configured to be either dependent on what the group policy states (AdminPolicy here -- this would require the Inherit box to be checked) OR, as shown below, Clientless SSL VPN, the SSL client, and IPSec are available specifically for this user.
Summary

A prime objective in all of the debug traces shown in this paper (as in the first paper in this 3-part series) was an effort to “target” the troubleshooting as much as possible. This was done by choosing the lowest number for the level as possible. Another option on the ASA, not explored in this paper, is the “debug crypto condition” command for IPSec debugs involving either a specific site-to-site peer or for IPSec clients belonging to a specific group.

The CLI debugs shown for IPSec cannot be done within the GUI interface and must be seen via a separate serial, telnet, or (preferable) SSH window. The Phase I and Phase II traces chosen for IPSec represent the most common errors made in configuration of either the layer 3 devices or the software client.

Since the CLI output of debug webvpn and debug vpn-sessiondb are lacking in many user and group attributes necessary for effective troubleshooting of SSL WebVPN, the focus was instead on the specific and appropriate areas of ASDM. As for any troubleshooting, the most effective use of any trace or debug tool is realized by a thorough knowledge of the operation of the protocols used for VPN session establishment.

Be sure to check out Part I: Connectivity Through ASA or PIX Firewalls and Part III.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge. Check out the following Global Knowledge courses:

- IINS (Introduction to IOS Network Security)
- SNRS (Securing Networks with Cisco® Routers and Switches)
- SNAF (Securing Networks with ASA Fundamentals)
- SNAA (Securing Networks with ASA Advanced)
- MARS (Cisco® Monitoring Analysis and Reporting System)
- CANAC (Cisco® Appliance for Network Admission Control)

For more information or to register, visit www.globalknowledge.com or call 1-866-925-7765 to speak with a sales representative. Our courses offer practical skills, exercises, and tips that you can immediately put to use. Our expert instructors draw upon their experiences to help you understand key concepts and how to apply them to your specific work situation. Choose from our more than 700 courses, delivered through Classrooms, e-Learning, and On-site sessions, to meet your IT, project management, and professional skills training needs.
About the Author

Douglas B. McKillip, P.E., CCIE #1851 is the president and principal consultant of Innovative Integrators Incorporated, a Delaware Corporation actually based in Delaware.

In addition to a BS and MS in Chemical Engineering from M.I.T., Doug also later obtained an M.S. from the University of Delaware in Computer and Information Science. After 15 years of experience at DuPont and a brief stint with the original startup company associated with the Raptor Eagle™ Firewall, Doug began his now 15+ year career of teaching and consulting, specializing in Internet Security with hardware from Cisco Systems® since 1998.

Doug’s hobbies include playing piano at his local church and physical activities such as running, cycling, swimming, golf, skiing, and windsurfing. He and his wife Karen reside in Wilmington, Delaware. Doug can be reached at innovativeint@mindspring.com.