



Setting Up the WSG

This chapter explains how to set up the WSG. The major sections are:

- [Before Getting Started with the WSG, page 2-1](#)
- [Configuring the WSG, page 2-21](#)

Before Getting Started with the WSG

These sections explain software and hardware configuration requirements for the WSG. They also explain what to do on the Supervisor (SUP) Engine and SAMI processors before using the WSG:

- [Single-Entity Configuration, page 2-1](#)
- [Understanding WSG Prerequisites, page 2-16](#)
- [Establishing a PPC Session, page 2-16](#)
- [Understanding WSG Prerequisites, page 2-16](#)
- [Establishing a PPC Session, page 2-16](#)
- [Assigning a Hostname to a PPC, page 2-17](#)
- [Setting Up VLAN Support, page 2-17](#)
- [Setting Up the PPCs, page 2-20](#)



Note

For all of the commands you can issue on the PPC with the WSG, see the *Cisco Service and Application Module for IP User Guide*.

Single-Entity Configuration

WSG Release 1.2 and above supports a single point of configuration and a single OAM interface per service blade. These two features are intended to increase the configurability and manageability of applications using the SAMI hardware platform.

Single- entity configuration allows you to configure all CPUs using a single director CPU. Each CPU still requires its own configuration, but the single-entity configuration duplicates the configuration to all CPUs. The benefit is that a common configuration for each CPU can all be entered into a single CPU instead of having to configure each CPU separately.

You are presented with the standard single-PPC CLI mode upon opening a session (console or SUP session) to the director PPC (PPC3). From this session, you can enter the **all** mode by using the **entity all** command. By default, commands entered in **entity all** mode are executed on all PPCs. However, certain commands such as **show clock** or **show version** causes the same output on all processors and need not be repeated for each.

From the director PPC you can open a session to a specific subordinate PPC to execute commands only on that PPC. When you open the session, you will be placed in PCC-specific EXEC mode. Commands are entered as though you are connected to the PCC individually.

From PCC-specific EXEC mode, you can use the **configure terminal** command to enter **config** mode. This mode is also PPC-specific, which allows you to execute configuration commands that are applicable only to that PPC. As with the **EXEC** mode, some commands are not applicable in the PPC-specific **EXEC** mode. These commands will print an appropriate message and exit.

You can enter the **config all** mode from the **exec all** mode by entering the **configure terminal** command. Similar to the **exec all** mode, commands entered in this mode are executed on all PPCs, unless they are present in the lookup table described above.

Commands that display statistics and other counters need to be aggregated to provide you with meaningful data. Each application will have to register these commands as single-execution CLIs using the lookup table, and use IPC mechanisms to retrieve and aggregate data. For example, to display the total number of tunnels created, the callback function needs to get that information from all PPCs, and aggregate them before they are displayed.

Configuration Details

The following list provides important configuration information for the single-entity feature:

- PPC3 is designated as the director PPC.
- **Entity all** mode is available only on the director PPC. Use the **entity all** command to enter the mode; **exit**, or **entity none** will exit the **entity all** mode.
- Commands that are entered in the **entity all** mode are executed on all PPCs.
- If a command that is not applicable is entered in the **all** mode, it will only be executed only the director PPC.
- From the director PPC, you can switch to another PPC using the **processor X** command, where *X* is the specific processor number (4-8)
- All commands are supported in the **entity-all** mode. However, some commands cannot be (for example, **interface vlan**), or need not be (for example, **show version**) executed on all PPCs.

The following message is displayed when such commands are executed:

```
Info: Command executed only on the master processor. If required, execute the command
on other processors
```

- If a **config** command fails on any of the subordinate PPCs, execution is aborted at that point (but not rolled back) with the following message:

```
Warning: Command failed on processor 4, aborting execution
```

- If an **exec** command fails on any of the subordinate PPCs, execution still continues on the remaining PPCs. The following message is printed:

```
Warning: Command failed on processor 4
```

Configuration Example

The following example shows the work flow that you can use to configure all processors from scratch using entity **all** mode. This example uses remote access crypto profile types. In remote-access configurations, the processors typically have common configuration parameters between them, including the names of the crypto profiles and address pools.

The following steps assume that a session is first opened from the SUP to the director PPC3 on the SAMI, using the command, **session slot slot_num processor proc_num**.

Configure Parameters That Must Be Unique To Each Processor

Step 1 Configure the hostname:

```
WSG# conf t
Enter configuration commands, one per line. End with CNTL/Z.
WSG(config)# host s3p3
```

Step 2 Configure the VLAN interface:

```
s3p3(config)# interface vlan 63
s3p3(config-if)# ip address 88.88.63.133 255.255.255.0
s3p3(config-if)# exit
```

Step 3 Configure the default gateway:

```
s3p3(config)# ip route 0.0.0.0 0.0.0.0 88.88.63.100
```

Step 4 Configure address pools:

```
s3p3(config)# crypto address-pool RAS-pool
s3p3(config-address-pool)# start-ip 10.133.0.1 end-ip 10.133.255.254 netmask 255.255.0.0
s3p3(config-address-pool)# exit
```

Step 5 Repeat the above commands on each processor, modifying the configuration appropriately. Using the **processor** command, you can switch to the other processors without logging out of the director. For example:

```
s3p3(mode-all)# processor 4
Trying 127.0.0.34...
Connected to 0x7f000022.
Escape character is '^]'.

MontaVista(R) Linux(R) Carrier Grade Edition 5.0 (custom)
Linux/ppc 2.6.21_mvlcge500-octeon-mips64-octeon_v2_be

Vegas Shell -- CGE 5.0 Version
Copyright (c) 1985-2008 by Cisco Systems, Inc.
All rights reserved.
```

Configure Parameters That are Common to all Processors or May Only Apply to the Director



Note

If a command only executes on the director processor, you will receive a warning:
INFO : Command executed only on master processor. If required, execute the command on other processors.

Step 1 Configure the single-entity OAM interface and its associated static route on the master PPC:

```
s3p3(config)# interface vlan 223
s3p3(config-if)# ip address 222.222.223.133 255.255.255.0
s3p3(config-if)# exit
s3p3(config)# oam mode single 223
s3p3(config-single-oam)# oam-ip route 44.44.44.0 255.255.255.0 222.222.223.100
s3p3(config-single-oam)# exit
s3p3(config)#
```

Step 2 Enter **entity all** mode.

```
s3p3# entity all
```

Step 3 Configure the DNS IP address:

```
s3p3(mode-all)(config)# ip name-server 44.44.44.201
s3p3(mode-all)(config)#
```

Step 4 Configure the IP address for external logging host:

```
s3p3(mode-all)(config)# logging ip 44.44.44.17
```

Step 5 Configure the SNMP host:

```
s3p3(mode-all)(config)# snmp-server host 44.44.44.16 traps version 2c public
```

Step 6 Configure the SNMP community strings:

```
s3p3(mode-all)(config)# snmp-server community public ro
s3p3(mode-all)(config)# snmp-server community private rw
```

Step 7 Configure the SNMP traps:

```
s3p3(mode-all)(config)# snmp-server enable traps snmp authentication
s3p3(mode-all)(config)# snmp-server enable traps interface
s3p3(mode-all)(config)# snmp-server enable traps syslog
s3p3(mode-all)(config)# snmp-server enable traps ipsec
```

Step 8 Configure the PKI certificates and trustpoints:

```
s3p3(mode-all)(config)# crypto pki wsg-cert Certs-SAMI.crt wsg-private-key
PrivateKeys-SAMI.prv
Copying Certs-SAMI.crt from SUP to PPC3...
done.
Copying PrivateKeys-SAMI.prv from SUP to PPC3...
done.
Copying Certs-SAMI.crt from SUP to PPC4...
done.
Copying PrivateKeys-SAMI.prv from SUP to PPC4...
done.
Copying Certs-SAMI.crt from SUP to PPC5...
done.
Copying PrivateKeys-SAMI.prv from SUP to PPC5...
done.
Copying Certs-SAMI.crt from SUP to PPC6...
done.
Copying PrivateKeys-SAMI.prv from SUP to PPC6...
done.
Copying Certs-SAMI.crt from SUP to PPC7...
done.
Copying PrivateKeys-SAMI.prv from SUP to PPC7...
done.
Copying Certs-SAMI.crt from SUP to PPC8...
done.
Copying PrivateKeys-SAMI.prv from SUP to PPC8...
done.
```

```
s3p3(mode-all)(config)# crypto pki trustpoint rootCA cacert.crt crl disable
Copying cacert.crt from SUP to PPC3...
done.
Copying cacert.crt from SUP to PPC4...
done.
Copying cacert.crt from SUP to PPC5...
done.
Copying cacert.crt from SUP to PPC6...
done.
Copying cacert.crt from SUP to PPC7...
done.
Copying cacert.crt from SUP to PPC8...
done.
```

Step 9 Configure the crypto profile:

```
s3p3(mode-all)(config)# crypto profile RAS-prof
s3p3(mode-all)(config-crypto-profile)# isakmp
s3p3(mode-all)(config-crypto-profile-isakmp)# self-identity id-type fqdn id SAMI.cisco.com
s3p3(mode-all)(config-crypto-profile-isakmp)# lifetime 86400
s3p3(mode-all)(config-crypto-profile-isakmp)# exit
s3p3(mode-all)(config-crypto-profile)# ipsec
s3p3(mode-all)(config-crypto-profile-ipsec)# access-permit ip 172.60.0.0 subnet 16
s3p3(mode-all)(config-crypto-profile-ipsec)# ip address-pool RAS-pool
s3p3(mode-all)(config-crypto-profile-ipsec)# security-association lifetime 28800
s3p3(mode-all)(config-crypto-profile-ipsec)# exit
```

Step 10 Activate the crypto profile:

```
s3p3(mode-all)(config-crypto-profile)# activate
```

Step 11 Display the Running Configuration.

```
s3p3(mode-all)# show run
```

CPU 3

```
Generating configuration.....
hostname s3p3
logging ip 44.44.44.17
snmp-server enable traps snmp authentication
snmp-server enable traps interface
snmp-server enable traps syslog
snmp-server host 44.44.44.16 traps version 2c public
snmp-server community public ro
snmp-server community private rw
ip name-server 44.44.44.201
snmp-server enable traps ipsec
crypto address-pool "RAS-pool"
    start-ip 10.133.0.1 end-ip 10.133.255.254 netmask 255.255.0.0
!
crypto pki wsg-cert Certs-SAMI.crt wsg-private-key PrivateKeys-SAMI.prv
crypto pki trustpoint rootCA cacert.crt crl disable
!
crypto profile "RAS-prof"
    isakmp
        lifetime 86400
        self-identity id-type fqdn id SAMI.cisco.com
    ipsec
        security-association lifetime 28800
        access-permit ip 172.60.0.0 subnet 16
        ip address-pool "RAS-pool"
    activate
!
interface vlan 63
```

```

ip address 88.88.63.133 255.255.255.0
interface vlan 223
ip address 222.222.223.133 255.255.255.0
ip route 0.0.0.0 0.0.0.0 88.88.63.100
oam mode single 223
oam-ip route 44.44.44.0 255.255.255.0 222.222.223.100

```

CPU 4

```

Generating configuration.....
hostname s3p4
snmp-server enable traps snmp authentication
snmp-server enable traps interface
snmp-server enable traps syslog

ip name-server 44.44.44.201
snmp-server enable traps ipsec
crypto address-pool "RAS-pool"
start-ip 10.134.0.1 end-ip 10.134.255.254 netmask 255.255.0.0
!
crypto pki wsg-cert Certs-SAMI.crt wsg-private-key PrivateKeys-SAMI.prv
crypto pki trustpoint rootCA cacert.crt crl disable
!
crypto profile "RAS-prof"
isakmp
lifetime 86400
self-identity id-type fqdn id SAMI.cisco.com
ipsec
security-association lifetime 28800
access-permit ip 172.60.0.0 subnet 16
ip address-pool "RAS-pool"
activate
!
interface vlan 64
ip address 88.88.64.134 255.255.255.0
ip route 0.0.0.0 0.0.0.0 88.88.64.100

```

CPU 5

```

Generating configuration.....
hostname s3p5
snmp-server enable traps snmp authentication
snmp-server enable traps interface
snmp-server enable traps syslog

ip name-server 44.44.44.201
snmp-server enable traps ipsec
crypto address-pool "RAS-pool"
start-ip 10.135.0.1 end-ip 10.135.255.254 netmask 255.255.0.0
!
crypto pki wsg-cert Certs-SAMI.crt wsg-private-key PrivateKeys-SAMI.prv
crypto pki trustpoint rootCA cacert.crt crl disable
!
crypto profile "RAS-prof"
isakmp
lifetime 86400
self-identity id-type fqdn id SAMI.cisco.com
ipsec
security-association lifetime 28800
access-permit ip 172.60.0.0 subnet 16
ip address-pool "RAS-pool"
activate
!
interface vlan 65

```

```
ip address 88.88.65.135 255.255.255.0
ip route 0.0.0.0 0.0.0.0 88.88.65.100
```

CPU 6

```
Generating configuration.....
hostname s3p6
snmp-server enable traps snmp authentication
snmp-server enable traps interface
snmp-server enable traps syslog

ip name-server 44.44.44.201
snmp-server enable traps ipsec
crypto address-pool "RAS-pool"
  start-ip 10.136.0.1 end-ip 10.136.255.254 netmask 255.255.0.0
!
crypto pki wsg-cert Certs-SAMI.crt wsg-private-key PrivateKeys-SAMI.prv
crypto pki trustpoint rootCA cacert.crt crl disable
!
crypto profile "RAS-prof"
  isakmp
    lifetime 86400
    self-identity id-type fqdn id SAMI.cisco.com
  ipsec
    security-association lifetime 28800
    access-permit ip 172.60.0.0 subnet 16
    ip address-pool "RAS-pool"
  activate
!
interface vlan 66
  ip address 88.88.66.136 255.255.255.0
ip route 0.0.0.0 0.0.0.0 88.88.66.100
```

CPU 7

```
Generating configuration.....
hostname s3p7
snmp-server enable traps snmp authentication
snmp-server enable traps interface
snmp-server enable traps syslog

ip name-server 44.44.44.201
snmp-server enable traps ipsec
crypto address-pool "RAS-pool"
  start-ip 10.137.0.1 end-ip 10.137.255.254 netmask 255.255.0.0
!
crypto pki wsg-cert Certs-SAMI.crt wsg-private-key PrivateKeys-SAMI.prv
crypto pki trustpoint rootCA cacert.crt crl disable
!
crypto profile "RAS-prof"
  isakmp
    lifetime 86400
    self-identity id-type fqdn id SAMI.cisco.com
  ipsec
    security-association lifetime 28800
    access-permit ip 172.60.0.0 subnet 16
    ip address-pool "RAS-pool"
  activate
!
interface vlan 67
  ip address 88.88.67.137 255.255.255.0
ip route 0.0.0.0 0.0.0.0 88.88.67.100
```

CPU 8

```

Generating configuration.....
hostname s3p8
snmp-server enable traps snmp authentication
snmp-server enable traps interface
snmp-server enable traps syslog

ip name-server 44.44.44.201
snmp-server enable traps ipsec
crypto address-pool "RAS-pool"
  start-ip 10.138.0.1 end-ip 10.138.255.254 netmask 255.255.0.0
!
crypto pki wsg-cert Certs-SAMI.crt wsg-private-key PrivateKeys-SAMI.prv
crypto pki trustpoint rootCA cacert.crt crl disable
!
crypto profile "RAS-prof"
  isakmp
    lifetime 86400
    self-identity id-type fqdn id SAMI.cisco.com
  ipsec
    security-association lifetime 28800
    access-permit ip 172.60.0.0 subnet 16
    ip address-pool "RAS-pool"
  activate
!
interface vlan 68
  ip address 88.88.68.138 255.255.255.0
ip route 0.0.0.0 0.0.0.0 88.88.68.100

```

Step 12 Save the configuration:

```

s3p3(mode-all)# copy running-config startup-config
running config of context Admin saved
Copying operation succeeded.

```

CPU 4

```

running config of context Admin saved
Copying operation succeeded.

```

CPU 5

```

running config of context Admin saved
Copying operation succeeded.

```

CPU 6

```

running config of context Admin saved
Copying operation succeeded.

```

CPU 7

```

running config of context Admin saved
Copying operation succeeded.

```

CPU 8

```

running config of context Admin saved
Copying operation succeeded.

```


SNMP Details

WSG Release 1.2 and above supports a single interface for SNMP management. In this instance, the director PPC acts as the target for all SNMP operations. All MIBs on the SAMI are accessible through the director PPC.

Since only the director PPC accepts SNMP protocol messages from external clients, only the director needs to be configured. All subordinate PPCs forward SNMP traps to the director, and the director will send them out.

To configure the single interface for SNMP, perform the following tasks in global configuration mode:

Step 1	WSG# snmp-server ?	
	community	Sets the community string and access privileges.
	contact	Modifies the sysContact
	enable	Enables or disable traps on each PPC.
	group	Define a User Security Model group.
	host	Specify hosts to receive SNMP notifications.
	location	Modifies the sysLocation.
	user	Defines a user who can access the SNMP engine
	view	Defines an SNMPv1/v2 MIB view.
		Note All of these commands are blocked on the subordinate PPCs (PPC4 - 8) except the command to enable traps.

Table 2-1 lists the MIBs supported by the WSG:

Table 2-1 MIBs Supported by the WSG

MIB Group	Tables	Comments
MIB-II	udpTable	—
MIB-II	tcpTable	—
MIB-II	atTable	—
MIB-II	ipAddrTable	—
MIB-II	ipRouteTable	—
MIB-II	ipNetToMediaTable	—
MIB-II	tcpConnTable	—
IF-MIB	ifTable	—
IF-MIB	inetNetToMediaTable	—
IF-MIB	ifXtable	—
UDP-MIB	udpEndpointTable	—
TCP-MIB	tcptcpConnectionTable	—
TCP-MIB	tcpListenerTable	—

Table 2-1 MIBs Supported by the WSG (continued)

MIB Group	Tables	Comments
IP-MIB	ipAddressTable	—
IP-MIB	ipAddressPrefixTable	—
IP-MIB	inetNetToMediaTable	—
IP-MIB	ipv4InterfaceTable	—
IP-MIB	ipv6InterfaceTable	—
HOST-RESOURCE-MIB	hrDeviceTable	—
HOST-RESOURCE-MIB	hrProcessorTable	—
HOST-RESOURCE-MIB	hrNetworkTable	—
HOST-RESOURCE-MIB	hrFSTable	—
HOST-RESOURCE-MIB	hrSWRunTable	—
HOST-RESOURCE-MIB	hrSWRunPerfTable	—
CISCO-PROCESS-MIB	cpmProcessTable	—
CISCO-PROCESS-MIB	cpmProcessExtRevTable	—
CISCO-SYSLOG-EXT-MIB	cseSyslogServerTable	—
CISCO-SYSLOG-MIB	clogHistoryTable	—
CISCO-IF-EXTENSION-MIB	cieIfInterfaceTable	—
CISCO-IF-EXTENSION-MIB	cieIfUtilTable	—
CISCO-IF-EXTENSION-MIB	cieIfNameMappingTable	—
CISCO-CONFIG-COPY-MIB	ccCopyTable	Supported only on the Master.
CISCO-IPSEC-FLOW-MONITOR-MIB	cikeGlobalStats	<p>The following IPsec MIB variables are supported:</p> <ul style="list-style-type: none"> • cikeGlobalActiveTunnels • cikeGlobalInitTunnelFails • cikeGlobalRespTunnelFails • cikeGlobalInOctets • cikeGlobalInPkts • cikeGlobalInDropPkts • cikeGlobalOutOctets • cikeGlobalOutPkts • cikeGlobalOutDropPkts

Table 2-1 MIBs Supported by the WSG (continued)

MIB Group	Tables	Comments
CISCO-IPSEC-FLOW-MONITOR-MIB	cikeTunnelTable	<p>The following IPSec MIB variables are supported:</p> <ul style="list-style-type: none"> • cikeTunLocalAddr • cikeTunRemoteAddr • cikeTunEncryptAlgo • cikeTunHashAlgo • cikeTunAuthMethod • cikeTunActiveTime • cikeTunInOctets • cikeTunInPkts • cikeTunInDropPkts • cikeTunOutOctets • cikeTunOutPkts • cikeTunOutDropPkts

Table 2-1 MIBs Supported by the WSG (continued)

MIB Group	Tables	Comments
CISCO-ENHANCED-IPSEC-FLOW-MIB	ceipSecGlobalStats	<p>The following IPsec MIB variables are supported:</p> <ul style="list-style-type: none"> • ceipSecGlobalInDecrypts • ceipSecGlobalInOctets • ceipSecGlobalInPkts • ceipSecGlobalOutEncrypts • ceipSecGlobalOutOctets • ceipSecGlobalOutPkts • ceipSecGlobalActiveTunnels • ceipSecGlobalInAuths • ceipSecGlobalInAuthFails • ceipSecGlobalInDecryptFails • ceipSecGlobalInDrops • ceipSecGlobalInReplayDrops • ceipSecGlobalNoSaFails • ceipSecGlobalOutAuths • ceipSecGlobalOutAuthFails • ceipSecGlobalOutDrops • ceipSecGlobalOutEncryptFails • ceipSecGlobalOutCompressedPkts • ceipSecGlobalOutCompFailPkts • ceipSecGlobalOutCompSkippedPkts • ceipSecGlobalOutCompTooSmallPkts • ceipSecGlobalOutUncompOctets • ceipSecGlobalProtocolUseFails • ceipSecGlobalThroughputUtilizationInterval • ceipSecGlobalThroughputLastUpdatedTime • ceipSecGlobalLastAveragePacketSize • ceipSecGlobalLastThroughputInMbps • ceipSecGlobalLastThroughputInKpps • ceipSecGlobalLastThroughputUtilization • ceipSecGlobalPeakThroughputUtilization • ceipSecGlobalPeakThroughputDateAndTime • ceipSecGlobalPeakThroughputInMbps • ceipSecGlobalPeakAvgPacketSize

Table 2-1 MIBs Supported by the WSG (continued)

MIB Group	Tables	Comments
CISCO-ENHANCED-IPSEC-FLOW-MIB	ceipSecTunnelTable	<p>The following IPsec MIB variables are supported:</p> <ul style="list-style-type: none"> • ceipSecTunLocalAddress • ceipSecTunLocalAddressType • ceipSecTunRemoteAddress • ceipSecTunRemoteAddressType • ceipSecTunInOctets • ceipSecTunInPkts • ceipSecTunOutOctets • ceipSecTunOutPkts • ceipSecTunInAuths • ceipSecTunInAuthFails • ceipSecTunInDecompOctets • ceipSecTunInDecrypts • ceipSecTunInDecryptFails • ceipSecTunInDropPkts • ceipSecTunInReplayDropPkts • ceipSecTunOutAuths • ceipSecTunOutAuthFails • ceipSecTunOutDropPkts • ceipSecTunOutEncrypts • ceipSecTunOutEncryptFails • ceipSecTunOutCompressedPkts • ceipSecTunOutCompSkippedPkts • ceipSecTunOutCompFailPkts • ceipSecTunOutCompTooSmallPkts • ceipSecTunPmtu • ceipSecTunActiveTime
	ciscoEnhIPsecFlowActivityGroup	<p>The following IPsec MIB variables are supported:</p> <ul style="list-style-type: none"> • ceipSecGlobalPreviousTunnels • ceipSecGlobalInDecompOctets • ceipSecGlobalSysCapFails • ceipSecGlobalThroughputInMbps • ceipSecGlobalThroughputInKbps • ceipSecGlobalThroughputUtilization

Table 2-2 lists the trap table notifications on the WSG:

Table 2-2 Trap Table Notifications Supported by the WSG

Trap	Table	Comments
coldStart	SNMPv2-MIB	—
authenticationFailure	SNMPv2-MIB	Community string provided in SNMP request is wrong.
Memory congestion: • clogMessageGenerated	CISCO-SYSLOG-MIB	—
CPU congestion: • cpmCPURisingThreshold • cpmCPUFallingThreshold	CISCO-PROCESS-MIB	—
Tunnel setup: • ciscoEnhIpsecFlowTunnelStart • ciscoEnhIpsecFlowTunnelStop • ciscoEnhIpsecFlowSysFailure • ciscoEnhIpsecFlowSetupFail	CISCO-ENHANCED-IPSEC-FLOW-MIB	<ul style="list-style-type: none"> • Aggregate trap for 1000 tunnel establishment. • Aggregate trap for 1000 tunnel deletion. • Exceeds tunnel capacity threshold. • Insufficient IP addresses.
Interface state: • linkUp • linkDown	IF-MIB	<ul style="list-style-type: none"> • VLAN Interface Up. • VLAN Interface Down.
clrRedundancyStateChange	CISCO-L4L7MODULE-REDUNDANCY-MIB	No other object from the CISCO-L4L7MODULE-REDUNDANCY-MIB is supported.
Flow system failure notification: • ceipSecFailreason • ceipSecFailPktSrcAddressType • ceipSecFailPktSrcAddress • ceipSecFailPktDstAddressType • ceipSecFailPktDstAddress	CISCO-ENHANCED IPSEC-FLOW-MIB	Provides flow failure identification <ul style="list-style-type: none"> • IPsec flow fail reason • Fail packet source IP address type • Fail packet source IP address • Fail packet destination IP address type • Fail packet destination IP address
Certificate expiry notification: • ciscoEnhIpsecFlowCertExpiry	CISCO-ENHANCED IPSEC-FLOW-MIB	Provides certificate identification (Subject Name, Serial Number, Issuer Name), expiration date and time, and expiration status: <ul style="list-style-type: none"> 1—certOK 2—certGoingExpired 3—certExpired

Table 2-2 Trap Table Notifications Supported by the WSG (continued)

Trap	Table	Comments
Certificate renewal notification: • ciscoEnhIpssecFlowCertRenewal	CISCO-ENHANCED IPSEC-FLOW-MIB	Provides certificate identification (Subject Name, Serial Number, Issuer Name), expiration date and time, and renewal status: 1—renewalNotNeeded 2—renewalRequestNeeded 3—renewalRequested 4—renewalSuccess 5—renewalFailedUpdate 6—renewalFailedExpired
Performance throughput notification: • ciscoEnhIpssecFlowSysFailure	CISCO-ENHANCED IPSEC-FLOW-MIB	Provides the failure reason for IPsec flow throughput: 17—performance utilization exceeding the threshold

**Note**

Use the SNMP Object Navigator (<http://tools.cisco.com/Support/SNMP/do/BrowseOID.do?local=en>) to obtain SNMP object information. For example, enter the object name ciscoEnhIpssecFlowCertExpiry.

Syslog Details

The WSG Release 1.2 and above supports a single interface for syslog collection. As part of the Memory Usage Monitoring feature, all PPCs send the syslogs to the director, and the director PPC sends them to an external server (if configured).

The external logging server can be configured only on the director PPC. Logs from any PPC can be viewed on the director (given the correct cpuid)

Additionally, from WSG 4.4.1, we can manually configure the facility value by configuring this CLI. With this, we will be able to send the generated syslog's to the syslog server with the configured facility value.

Step 1	<pre>WSG(config)# logging ? ip lineread</pre>	<p>ip—Configures the IP address of ext logging server. Only the director needs external logging server.</p> <p>lineread—Configures the number of lines to read log. The number of lines can still be configured (for show below).</p>
---------------	---	---

Step 2	<pre>WSG# show logging ? config message</pre>	<p>config—Shows the syslog configuration.</p> <p>message—Shows syslog messages. Messages can still be viewed on each PPC (using the correct cpuid).</p>
Step 3	<pre>WSG(config)# crypto facility ? <0-23> Value 0 to 23 WSG(config)# config message</pre>	<p>config—Configures the facility level to desired value.</p> <p>message—Sends the generated syslog's to the configured syslog server with the configured facility value (Will be able to configure only on the director PPC(PPC3)).</p>

Understanding WSG Prerequisites

The WSG requires a Cisco 7600 system with these:

- SUP Engine 720, with a Multilayer Switch Feature Card (MSFC), running Cisco IOS Release 12.2(33)SRC3, and a Compact flash (min 128MB) in Disk:0 slot, or
Cisco 7600 Series SUP 32, with a MSFC, running Cisco IOS Release 12.2(33)SRC3, and a Compact flash (min 128MB) in Disk:0 slot.
For details on upgrading the Cisco IOS release running on the SUP, see the “Upgrading to a New Software Release” section in the *Release Notes for Cisco IOS Release 12.2(33)SRC3*.
- Any module with ports connected to the network.
- Cisco Service and Application Module for IP (Cisco Product Number: WS-SVC-SAMI-BB-K9) with the 2 GB memory option (Cisco Product Number: MEM-SAMI-6P-2GB[=]). Release 1.1 of the WSG application ships loaded on the SAMI.

Establishing a PPC Session

To set up VLAN support, establish a session with a PPC. Perform this procedure for each of the six PPCs on a SAMI with WSG.



Note

Under conditions like low processor memory, a session to the SAMI may fail. If this occurs, use the physical front-panel console connections to access the SAMI (see the “Establishing a Console Connection on the SAMI” section of the *Cisco Service and Application Module for IP User Guide*).

To set up a PPC session from the SUP Console, enter:

Step 1	Sup# show module	Returns system information, like which slot contains the SAMI with the PPC to connect to.
Step 2	Sup# session slot <i>slot_number</i> processor <i>proc_number</i>	Sets up a session to a PPC where: <ul style="list-style-type: none"> • <i>slot_number</i>—Number of the slot in which the SAMI is installed. • <i>proc_number</i>—Number of the PPC. Valid values are 3 through 8.

Assigning a Hostname to a PPC

The default session prompt when you set up a session with a PPC is “switch.” To assign a hostname to a PPC other than switch, enter:

Step 1	switch# configure	Enables global configuration mode.
Step 2	switch(config)# hostname <i>name</i>	New hostname for the PPC. Enter a case sensitive text string with 1 to 32 alphanumeric characters.

This example shows a session with PPC 3 on a SAMI in slot 6, the hostname is changed to PPC3:

```
Sup> enable
Sup# session slot 6 processor 3
Trying... Open

switch# config
Enter configuration commands, one per line. End with CNTL/Z.
switch(config)# hostname PPC3
PPC3(config)#
```

Setting Up VLAN Support

SAMI does not have outside physical interfaces to receive traffic from the network. Instead, the SAMI uses VLAN interfaces to the SUP. To set up VLAN support between the SUP and PPCs, complete tasks in these sections:

- [Setting Up the SUP, page 2-17](#)
- [Setting Up the PPCs, page 2-20](#)

Setting Up the SUP

For PPCs to receive traffic from the SUP, complete these tasks on the SUP:

- Set up a VLAN for each PPC.
- Assign the VLANs to a VLAN group.
- Assign the VLAN groups to the SAMIs.
- Set up a default gateway VLAN.

Setting Up VLANs for the PPCs

To set up the VLANs for each PPC on the SUP, enter:

	Command	Purpose
Step 1	Sup> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Sup# configure terminal	Enters global configuration mode.
Step 3	Sup(config)# vlan <i>vlan-id</i>	Configures a VLAN where <i>vlan-id</i> is the number of the VLAN. Valid values are from 1 to 4094.
Step 4	Sup(config-vlan)# description <i>interface_description</i>	(Optional) Describes interface.
Step 5	Sup(config-vlan)# end	Exits VLAN configuration mode.

To create VLANs 71 to 76 on the SUP Console plus an OAM Vlan 100, enter:

```
Sup> enable
Sup# configure terminal
Sup(config)# vlan 71
Sup(config-vlan) exit
Sup(config)# vlan 72
Sup(config-vlan) exit
Sup(config)# vlan 73
Sup(config-vlan) exit
Sup(config)# vlan 74
Sup(config-vlan) exit
Sup(config)# vlan 75
Sup(config-vlan) exit
Sup(config)# vlan 76
Sup(config-vlan) exit
Sup(config)# vlan 100
Sup(config-vlan) exit
```

Assigning VLANs to the SAMI

SAMI PPC VLANs must be assigned to the same VLAN group. You cannot assign the same VLAN to many groups. However, you can assign a group to many SAMIs.

By default, one switched virtual interface (SVI) (required if the SUP participates in Layer-3 forwarding) can exist between an MSFC and a SAMI. Create and enable SVIs using the **svclc multiple-vlan-interfaces** command.

To assign VLANs to a SAMI, enter:

Step 1	Sup> enable	Enables privileged EXEC mode.
Step 2	Sup# configure terminal	Enters global configuration mode.

Step 3	<pre>Sup(config)# svclc vlan-group vlan_group_number vlan_range</pre>	<p>Assigns the VLANs to a secure group.</p> <ul style="list-style-type: none"> • <i>vlan_group_number</i>—Number of the VLAN group. • <i>vlan_range</i>—Number of the VLAN or VLANs identified as a single number (<i>n</i>), as a range of numbers (<i>n-m</i>), or as separate numbers or range of numbers, separated by commas (for example, 5,7-10,13,45-100).
Step 4	<pre>Sup(config)# svclc module slot_number vlan-group group_number_range</pre>	<p>Assigns VLAN groups to the SAMI, where:</p> <ul style="list-style-type: none"> • <i>slot_number</i>—Number of the slot in which the SAMI is installed. To view chassis slot numbers and modules, use the show module command in privileged EXEC mode. • <i>group_number_range</i>—VLAN group number identified as a single number (<i>n</i>), as a range of numbers (<i>n-m</i>), or as separate numbers or range of numbers, separated by commas (for example, 3,5,7-10). Only VLAN groups created using the svclc vlan-group global configuration command. <p>Note One VLAN group can be assigned to many SAMIs.</p>
Step 5	<pre>Sup(config)# svclc multiple-vlan-interfaces</pre>	<p>Enables many SVIs to be set up for SVCLC modules.</p>

For example:

- To create a VLAN group, group 50, with a VLAN range of 71 to 76, and 100 enter:

```
Sup> enable
Sup# configure terminal
Sup(config)# svclc vlan-group 50 71-76, 100
```

- To assign VLAN group 50 to the SAMI in slot 5, enter:

```
Sup(config)# svclc module 5 vlan-group 50
```

- To enable many SVIs to be set up for SVCLC modules (the SAMIs), enter:

```
Sup(config)# svclc multiple-vlan-interfaces
```

- To view the SAMI group configuration and associated VLANs, enter:

```
Sup(config)# exit
Sup# show svclc vlan-group
```

- To view VLAN group numbers for all modules, enter:

```
Sup# show svclc module
```

Setting Up the PPCs

To complete the configuration tasks for VLAN support do the following on each PPC:

1. Set up the default gateway.
2. Set up a VLAN interface.
3. Assign the interface to the corresponding VLAN on the SUP.



Note

Sharing of the same VLAN ID with different PPC's on SAMI WSG is not supported unless if it is for HA VLAN interface.

To set up the PPCs, enter the following commands from the SUP:

	Command	Purpose
Step 1	Sup> enable	Enables privileged EXEC mode.
Step 2	Sup# session slot <i>slot_number</i> processor <i>proc_number</i>	Sets up a session to a PPC where: <ul style="list-style-type: none"> • <i>slot_number</i>—Number of the slot in which the SAMI is installed. • <i>proc_number</i>—Number of the PPC. Valid values are 3 through 8. Note Set up one session per PPC.
Step 3	WSG# config	Enters global configuration mode.
Step 4	WSG(config)# interface vlan <i>number</i>	Creates a VLAN interface for the VLAN, and enters interface configuration mode.
Step 5	WSG(config-if)# description <i>interface_description</i>	(Optional) Describes interface.
Step 6	WSG(config-if)# ip address <i>ipv4 address</i>	Assigns an IP address to a VLAN interface for connectivity. IKE and ESP traffic from the endpoints use this IP address.
Step 7	WSG(config-if)# no shutdown	Enables a VLAN interface.
Step 8	WSG(config-if)# do show interface <i>vlan number</i>	Verifies that a VLAN is active.
Step 9	WSG(config-if)# do ping <i>ip_address</i>	Verifies network connectivity.
Step 10	WSG(config-if)# do show arp	Shows the ARP table.
Step 11	WSG(config-if)# exit	Exits interface configuration mode.
Step 12	WSG(config)# ip route <i>0.0.0.0 0.0.0.0 ip-addr</i>	Defines a default gateway (router).
Step 13	WSG(config)# end	Exits configuration mode.
Step 14	WSG# exit	Returns to the SUP console session.

For example:

- To create a VLAN interface on PPC3 on a SAMI in slot 5, enter:

```
Sup# session 5 processor 3
WSG> config
```

```
WSG(config)# interface vlan71
WSG(config-if)# ip address 10.22.22.2 255.255.255.0
WSG(config-if)# exit
```

- To verify the interface configuration of VLAN71, enter:

```
WSG# show interface vlan71
```

- To define a default gateway, enter:

```
Sup# session 5 processor 3
switch> config
```

```
switch(config)# ip route 0.0.0.0 0.0.0.0 10.22.22.1
```

Configuring the WSG

The WSG application is preloaded on the SAMIs. Set up IPSec parameters for your network using the WSG CLI. To set up the WSG to perform these procedures:

- [Single OAM Interface, page 2-22](#)
- [Resource Monitoring, page 2-23](#)
- [Configuring WSG Global Parameters, page 2-24](#)
- [Configuring the WSG Profile, page 2-28](#)
- [Configuring IKE, page 2-31](#)
- [High Availability, page 2-31](#)
- [Configuring High Availability on SAMI COSLI, page 2-35](#)
- [IKE SA Handling, page 2-50](#)
- [IPSec SA Handling, page 2-51](#)
- [Configuring IPSec, page 2-52](#)
- [Site-to-Site Scalability, page 2-53](#)
- [Certificate Management Protocol, page 2-56](#)
- [Online Certificate Status Protocol, page 2-62](#)
- [DHCP Address Allocation, page 2-62](#)
- [IPv6, page 2-67](#)
- [Blacklisting, page 2-69](#)
- [RADIUS Accounting, page 2-70](#)
- [EAP Peer Authentication, page 2-73](#)
- [Reverse Route Injection \(RRI\), page 2-73](#)
- [VRF Configuration, page 2-75](#)
- [Configuring WSG Performance/Throughput Indicators, page 2-80](#)
- [Configuring IKE/IPSec Stats Collection and Timing Enhancements for SNMP, page 2-85](#)

To apply changes to the default configuration, save the running configuration to the configuration file using the **copy running-config startup-config** command from a PPC session.

**Note**

Individually set up each of the six PPCs.

Single OAM Interface

As described in the previous sections, the WSG uses a single interface for SNMP and syslog messages from all PPCs through the director PPC.

The single OAM interface works with the single-entity mode, allowing all management traffic (such as DNS, CRL, and HTTP) to flow through the same interface on the director PPC. It is desirable to use a single interface per blade for management traffic when only a limited number of management IP addresses are available. Rather than using a separate interface on each PPC on the management network, you can configure the single OAM interface to allow all PPCs on the WSG to use the same interface on the director PPC. Configuring the single OAM interface is a two-step process. First, identify the interface used for OAM traffic. Second, configure `oam-ip` routes for the management subnets. Once configured, the WSG internally routes all the traffic to the connected OAM subnet (and all management networks configured through the `oam-ip` route) through the director PPC.

**Note**

You should only use this feature should for protocols that generate minimal amount traffic (for example, CRL).

**Note**

SNMP and SYSLOG are only run on the director PPC. Even though SNMP and Syslog can use the single OAM interface to reach the management network, they do not need to be routed through the director PPC (there is no need to configure the OAM interface and OAM routes for SNMP and SYSLOG purposes).

Configuring the Single OAM Interface

To configure the Single OAM interface feature on the WSG, perform the following tasks:

Step 1	<code>WSG(config)# oam mode single vlan 223</code>	This command identifies the interface used for single mode OAM traffic. All management traffic from the director and subordinate PPC destined to vlan 223 subnet will now be directed through this interface.
Step 2	<code>WSG(config)# oam-ip route</code>	This command configures static routes on the director PPC and subordinate PPCs for the management subnet(s). This command is similar to ip route in functionality, with the exception that it affects the routes on the subordinate PPCs as well.

Here is an example:

```
interface vlan 223
 ip address 222.222.223.123 255.255.255.0
 oam mode single 223
 oam-ip route 44.44.44.0 255.255.255.0 222.222.223.100

R7-S2P3(mode-all)# sh ip route
```

```
127.0.0.0/24 dev eth0 src 127.0.0.23
44.44.44.0/24 via 222.222.223.100 dev eth0.223
222.222.223.0/24 dev eth0.223 src 222.222.223.123
```

```
CPU 4
127.0.0.0/24 dev eth0 src 127.0.0.24
44.44.44.0/24 via 127.0.0.23 dev eth0
222.222.223.0/24 via 127.0.0.23 dev eth0
```

Resource Monitoring

Resource monitoring notifies you (using SNMP traps) when utilization of one or more system resources such as CPU, memory, and storage of a system crosses certain predefined thresholds.

Monitoring CPU Usage

The CPU Threshold Notification feature notifies users by generating a SNMP trap message when a predefined threshold of CPU usage is crossed. Two types of CPU utilization threshold are supported: rising threshold and falling threshold. A rising CPU utilization threshold specifies the percentage of CPU resources that, when exceeded for a configured period of time, triggers the `cpmCPURisingThreshold` notification. Similarly, a falling CPU utilization threshold specifies the percentage of CPU resources that, when CPU usage falls below this level for a configured period of time, triggers `cpmCPUFallingThreshold` notification.

To configure the CPU Threshold Notification feature, perform the following tasks: use the **`process cpu threshold rising percentage interval seconds`** [**`falling percentage interval seconds`**] command.

Step 1	<code>WSG# configure</code>	Enables global configuration mode.
Step 2	<code>WSG(config)#process cpu threshold rising percentage interval seconds</code> [<code>falling percentage interval seconds</code>]	Enables the CPU Threshold Notification feature and establishes the rising and falling percentages threshold values. Threshold values: min 1% to max 100%. Threshold interval: 5 – 86400 seconds. falling threshold should always be less than, or equal to the configured rising threshold value. This parameter is optional.

The following example shows how to set a rising CPU threshold notification for total CPU utilization. When total CPU utilization exceeds 95 percent for a period of 5 seconds or longer, a rising threshold notification is sent.

```
ppc3(config)# process cpu threshold rising 95 interval 5
```

Monitoring Memory Usage

The Memory usage monitoring feature allows syslogs to be generated to indicate that free memory has fallen below a configured threshold, or the system has recovered from a low memory situation.

To configure the Memory Usage Monitoring feature, perform the following tasks:

Step 1	<code>WSG# memory free low watermark processor threshold</code>	Configures the memory threshold that, when free memory falls below, generates a syslog. The free memory threshold value can range from 1024KB to 1996000KB.
---------------	---	---

The following example specifies a threshold of 10000 KB of free processor memory before a low-memory syslog is generated:

```
ppc3(config)# memory free low-watermark processor 10000
```

Once the available free memory rises to above 5 percent of the threshold (1.05 x 10000 in the above example), another message is generated that indicates that the free memory has recovered.

Configuring WSG Global Parameters

Modifications to most of the global parameters will take effect only when the next **activate** command is issued for the profile.

Configuring IKE Retry Count

To set the number of IKE retry connection attempts, perform the following task:

Step 1	<code>WSG(config)# crypto ike-retry-count value</code>	Sets the number of IKE retry connection attempts.
---------------	--	---

Configuring Remote Secret

To set the remote shared secret, perform the following task:

Step 1	<code>WSG(config)# crypto remote-secret id_type id secret</code>	Sets the remote shared secret.
---------------	--	--------------------------------

Configuring a Local Address Pool

The WSG keeps a pool of private addresses from the protected network. When the WSG receives an endpoint SA with an internal IP address, it assigns an unused address from the address pool. The address does not expire as long as the SA is up. When the SA is removed, the address is released to the local pool.

When setting up an address pool, note:

- Set up address pools using the **start-ip** command.
- On the SUP, set up a static route to the PPC. This handles traffic sent to an address in the local address pool to be routed to the PPC.



Note

This configuration is optional for site-to-site profiles.

For example, to set up an address pool from which to assign addresses on the SAMI, enter:

```
WSG# config
Enter configuration commands, one per line. End with CNTL/Z.
WSG(config)# crypto address-pool foo
WSG(config-address-pool)# start-ip 192.168.10.1 end-ip 192.168.10.10 netmask 255.255.255.0
```


Adding the DNS Server to the Address Pool

To specify the DNS server that is passed to the access point (the remote end point) when there is a request for a DNS server during IKE negotiation, perform the following tasks:

	Command	Purpose
Step 1	<pre>WSG# conf t Enter configuration commands, one per line. End with CNTL/Z. WSG(config)# crypto address-pool foo WSG(config-address-pool)# dns-server ? <A.B.C.D> Enter IP address WSG(config-address-pool) # dns-server 172.20.10.1</pre>	<p>Specifies the DNS server IP address that is to be passed to the access point (the remote end point) when there is a request for a DNS server during IKE negotiation.</p> <p>If the DNS server IP address is not required to be sent to the remote access point, this command is not required.</p>

Configuring Authentication Parameters

For secure communication, the WSG requests and sends X.509 digital certificates to authenticate IPsec endpoints.

Multiple CA Trust Anchors

A trust anchor is a third party the WSG trusts and to which it has a certification path. The trust anchor certifies the WSG. This certificate has information about prefixes that a WSG is allowed to use in router advertisements. Authorization delegation discovery enables a node to adopt a WSG as its default router.

CA Certificate Chaining

A certificate chain is a sequence of certificates with dependent trust relationships. The first certificate is self-signed by the CA. Each subsequent certificate creates an association between a certificate owners, or CAs in the chain. This process creates a trust chain from trusted peer to a CA. The CA endorses the identity of the peer certificate by signing it.

The WSG keeps a list of trusted CA certificates in its root certificate directory. If the CA certificate is not on this list, the WSG will refuse to authenticate the peer until a CA certificate is obtained and validated. If the CA certificate is on this list, the WSG trusts the signed peer certificate and will allow a security association in that peer.



Note

This release supports manual certificate installation.

To enable digital certification, complete these tasks:

- [Generating an RSA Key Pair and CSR, page 2-26](#)
- [Submitting the CSR to the CA, page 2-27](#)

- [Specifying Certificates and a Private Key on the WSG, page 2-27](#)
- [Configuring the WSG Profile, page 2-28](#)

Generating an RSA Key Pair and CSR

RSA key pairs sign, encrypt, and decrypt. To get a Certificate Authority (CA) you first need a Certificate Signing Request (CSR).

1. The **crypto rsa-keygen** command makes a private key (wsg.prv file) and a CSR (wsg-pem.csr file) based on the CSR parameters you enter.
2. The private key file is copied to the SUP bootflash or bootdisk, depending on which is available.
3. The public key, the second key of the key pair, is embedded in the CSR.



Note

If all WSG instances in a SAMI must share a certificate, use the **crypto rsa-keygen** command only once on one PPC in the SAMI. If the WSGs must use separate certificates, use the **crypto rsa-keygen** command on each PPC in the SAMI.

To generate an RSA key pair and CSR, enter the following on a PPC in the SAMI:

	Command	Purpose
Step 1	WSG# crypto rsa-keygen id-type id-type id id	<p>Makes an RSA key pair and CSR using information to authenticate the site, where:</p> <ul style="list-style-type: none"> • id-type id-type configures the local identity type. Possible values are fqdn and ip. <p>Note Changing local identity type drops all existing tunnels.</p> <ul style="list-style-type: none"> • id sets the IP address or domain name for the key pair

To generate an RSA key pair and CSR for a remote peer, enter:

	Command	Purpose
Step 1	WSG# crypto rsa-keygen id-type id	<p>Makes an RSA key pair and CSR for a remote peer:</p> <ul style="list-style-type: none"> • id-type—fqdn • id—test.cisco.com <p>Generating certificate request...done. Copying private key (wsg.prv) to SUP...done. Copying certificate request (wsg-pem.csr) to SUP...done.</p> <pre> -----BEGIN CERTIFICATE REQUEST----- MIIBrjCCARcCAQAwNTElMAkGA1UEBhMCVVMxDTALBgNV AsTBFNNQ1UxFzAVBgNV BAMTDnNlZ3cuY2l2Y28uY29tMIGfMA0GCSqGSIb3DQEBA QUAA4GNADCBiQKBgQCr xsJE11PDRytSqzGH7aVi4fmf8rXygmNYCcOPvnIQybMoJ t5PdObtbXREJ2r4ON6Y gh4E+IXbIe3yig6friBFMEkYgQJuLe13P8wELDdHyWA6v BLzVgZuwa34Me8B0nKa LMaU7kZ47sConEOE1c27NB16mI5D4rVdBnacj4/GCQIDA QABoDkwNwYJKoZIHvcN AQkOMSowKDALBgNVHQ8EBAMCBaAwGQYDVR0RBBIwEIIoc 2Vndy5jaXNjby5jb20w DQYJKoZIhvcNAQEFBQADgYEASEqXB00k1VfguVdUf9LU4 Im1+3l+hWErFp/M5Nh4 r+h5ukmCW9ldPPIZxOkV2n2wedLf6mUKTcdzdOLUiwgrS ozHSfLWgpXW+upxZDgn Nk/LvIW3+NpwnjzCmYJEZKFpWglxKzzwMAe99AOpH+Z6y hrw5ffcc9qZCcWXkeHw 1Iw= -----END CERTIFICATE REQUEST----- </pre>



Note

RSA key-pairs can be generated outside the PPC. If the CSR and private key are generated outside of the WSG, copy the private key file to the SUP before defining the certificates on the WSG.

Submitting the CSR to the CA

After generating the RSA key pair and CSR:

1. Submit the CSR to a CA using FTP or a cut-and-paste from the console session.
2. The CA signs the CSR using its RSA private key.
3. The CSR becomes a WSG certificate.
4. Copy the certificate to the SUP.
5. Set up the PPCs to use the private key and certificates.

Specifying Certificates and a Private Key on the WSG

When you enter the **crypto pki wsg-cert** and **crypto pki trustpoint** commands, the certificates (and optionally, the private key) are copied from the SUP to the WSG, and the WSG configuration is updated.



Note Before you can issue the **crypto pki wsg-cert** and the **crypto pki trustpoint** commands, the certificates and a private key must exist on the SUP.

To set the certificates for the WSG to use during authentication, enter:

	Command	Purpose
Step 1	WSG# config	Enters global configuration mode.
Step 2	WSG(config)# crypto pki wsg-cert <i>cert-filename.crt</i> [wsg-private-key <i>private-key-filename.prv</i>]	Configures the WSG to use the certificate for certificate based authentication. The certificate and private key are copied from the Cisco 7600 SUP to the WSG. Up to 20 certificate/private key pairs may be configured on the WSG. <i>cert-filename.crt</i> —Name of the WSG certificate file. Ensure certificate filenames end with a .crt file extension. <i>private-key-filename.prv</i> —Private key filename. To use the private key, set the name of the private key file, ending with a .prv extension. Note The WSG uses the private key from the crypto rsa-keygen command if you do not set a private key.
Step 3	WSG(config)# crypto pki trustpoint { rootCA subCA } <i>filename.crt</i> [cr1 disable]	Sets up a CA certificate.

For example:

- To set up the WSG to use a certificate with the name cert1.crt and a private key file named wsg.prv contained on the SUP, enter:

```
WSG# config
Enter configuration commands, one per line. End with CNTL/Z.
WSG(config)# crypto pki wsg-cert cert1.crt wsg-private-key wsg.prv
Copying cert1.crt from SUP...done
```

- To set up the WSG to use a CA certificate with the name cert-ca1.crt on the SUP, enter:

```
WSG# config
Enter configuration commands, one per line. End with CNTL/Z.
WSG(config)# crypto pki trustpoint rootCA cert-ca1.crt
Copying cert1-ca1.crt from SUP...done
```

Configuring the WSG Profile

As described in the earlier sections, part of the configuration for the WSG is entered globally. The rest of the configuration is entered by creating a crypto profile.

A profile is a combination of IKE and IPSec parameters that apply to all the tunnels that will get established on the WSG. A profile is created using the **crypto profile** command. The IKE and IPSec related parameters are entered in the **isakmp** and **ipsec** submodes of the **crypto profile** command mode. A profile must be activated using the **activate** command before the tunnels can be established.

The profile parameters can only be modified when the profile is in an deactivated state. The profile can be deactivated by issuing the **no activate** command under the **crypto profile** submode.

The global configuration for WSG is also effective only when the profile is active. If any global WSG configuration is modified while the crypto profile is in active state, the changes will not take effect till the profile is first deactivated, and then activated again.



The *profile name* should be unique; you cannot use the same name for two different profiles.

- [Configuring the WSG Parameters](#)
- [Configuring IKE](#)
- [Configuring IPSec](#)

Configuring the WSG Parameters

To set up an IKE identity for the WSG/PPC to use during authentication, enter:

	Command	Purpose
Step 1	WSG# config	Enters global configuration mode.
Step 2	WSG(config)# crypto profile WSG(config-crypto-profile)# profile-type ? remote-access Profile Type remote-access (default) site-to-site Profile Type site-to-site	Enters the crypto profile. A crypto profile can be either remote access type, or site-to-site type. The profile-type command is used to specify the type of each profile that you create. If the type is not specified the default is remote-access. Only one remote access profile can be active. Multiple site-to-site profiles can be active. Note You should take special care to configure the proper access-permit command that corresponds to the profile type used, as described in the access-permit command. Note The maximum number of possible site-to-site profiles supported is 25.
Step 3	WSG(config-crypto-profile)# isakmp	Enters ISAKMP submode.

Command	Purpose
Step 4 WSG(config-crypto-profile-isakmp)# self-identity id-type type id id	Defines the IKE identity of the local IPSec client, where: <ul style="list-style-type: none"> • self-identity must match the certificate's identity. <div style="margin-left: 40px;">  <p>Note The supported characters while configuring the self-identity are dash, dot, underscore, a-z, A-Z and 0-9.</p> </div> <ul style="list-style-type: none"> • id-type configures the IKE identify of the local client. The IKE identity is the identity the client uses when authenticating to the gateway. Valid values are: <ul style="list-style-type: none"> - ip—IP address - dn—Distinguished name - fqdn—Fully-qualified domain name - email—E-mail address <div style="margin-left: 40px;">  <p>Note The maximum size supported for the id-types is 256 bytes.</p> </div> <ul style="list-style-type: none"> • id sets the ID data of the remote IKE client (name, IP address, dn, FQDN, or e-mail address).
Step 5 WSG(config-crypto-profile-isakmp)# sequence-number [extended short]	Specifies that a 32-bit (short) or 64-bit (extended) sequence number is used for a profile. 32-bit is the default value.
Step 6 WSG(config-crypto-profile-isakmp)# exit	Exits ISAKMP submode.

For example, to set up an id-type of “fqdn,” id “wsg.cisco.com,” on the PPC using the crypto profile “remote-access”:

```
WSG# config
Enter configuration commands, one per line. End with CNTL/Z.
WSG (config)# crypto profile remote-access
WSG(config-crypto-profile)# isakmp
WSG(config-crypto-profile-isakmp)# self-identity id-type fqdn id wsg.cisco.com
WSG(config-crypto-profile-isakmp)# sequence-number extended
WSG(config-crypto-profile-isakmp)# exit
WSG(config-crypto-profile)# exit
```

Extended Sequence Number

WSG Release 1.2 and above adds Extended Sequence Number (ESN) support as longer lifetimes are expected in customer deployments. Additionally, more traffic is expected in site-to-site setups. Extended Sequence Number (64-bit sequence number) implementation is required in such cases. In this release,

the sequence number length cannot be negotiated by the peer with the SAMI. The peer will have to match the setting on the SAMI (default is 32-bit sequence number). The 64-bit sequence number can be configured using the above CLI.

Configuring IKE

The goal of IKE negotiation is to find a compatible key exchange on both peers:

1. Peer A sends its allowed IKE policies to Peer B.
2. Peer B compares Peer A's policies to its highest priority policy.
3. Peer B tries to find policies that have the same values for the following:
 - Encryption
 - Hash algorithm
 - Authentication
 - Diffie-Hellman parameters
 - Lifetime
4. If Peer B cannot find a match, negotiation fails and IPsec is not established. If Peer B finds a match, negotiation completes and IPsec SAs are established.

To set up IKE values, perform the following tasks:

	Command	Purpose
Step 1	<code>WSG# config</code>	Enters global configuration mode.
Step 2	<code>WSG(config)# crypto profile name</code>	Sets the crypto profile name.
Step 3	<code>WSG(config)# isakmp</code>	Enters ISAKMP submode.
Step 4	<code>WSG(config-crypto-profile-isakmp)# encryption {des 3des aes aes192 aes256}</code>	Sets the IKE secret encryption scheme.
Step 5	<code>WSG(config-crypto-profile-isakmp)# hash {sha1 sha2 md5 aes-xcbc}</code>	Sets the hash algorithm.
Step 6	<code>WSG(config-crypto-profile-isakmp)# version {1 2 both}</code>	Sets the IKE version.
Step 7	<code>WSG(config-crypto-profile-isakmp) authentication {rsa-sig pre-share}</code>	Sets IKE authentication.
Step 8	<code>WSG(config-crypto-profile-isakmp)# group {1 2 5 14 15 16 17 18}</code>	Sets a DH group ID.
Step 9	<code>WSG(config-crypto-profile-isakmp)# lifetime {seconds}</code>	Sets the SA lifetime.

High Availability

In WSG Release 2.0 and above, inter-chassis stateful 1:1 redundancy is supported. Two redundancy modes are supported: active-standby and active-active (introduced in WSG Release 4.0). In the active-standby mode, redundancy works at the SAMI level. Only PPC3 or all 6 PPCs on the SAMI are in either active or standby state. The PPC of the active SAMI synchronizes its state to the corresponding

PPC of the redundant, standby SAMI. In the active-active mode, redundancy works at the PPC level. Only PPC3 and PPC4 on the SAMI are used—one PPC is in active state while the other PPC is in standby state.

The WSG redundancy feature works with all IPsec supported features including IKEv1, IKEv2, ESN, anti-replay, DPD, and NAT-traversal. WSG redundancy is applicable to both remote access and site-to-site tunnels.

If a primary SAMI fails, traffic is switched to the newly active SAMI. The established tunnels stay up and continue to pass traffic after failover, and the IKE/IPsec internal state is synchronized between the active and redundant WSGs. The expected traffic loss during failover is less than one second after detection of the failure. The WSG responds to IKE packets (DPDs and SA INIT messages) and load balancer probes within one second.

There is no impact on the WSG routing for inner and outer addresses after a failover. The WSG maintains IP addresses that are assigned from the local address pool. The newly active SAMI allocates IP addresses from the local pool when a SA is created and releases IP addresses to the local pool when a SA is deleted.

**Note**

WSG does not support the single OAM feature in the active-active redundancy mode.

**Note**

WSG does not support single-entity configuration for application configuration commands.

Configuring High Availability

Follow these steps to configure high availability on redundant WSGs:

- Step 1** Shutdown the standby WSG (WSG B) to ensure no IP address conflicts.
- Step 2** Configure the node-specific configuration (interface, IP address, alias IP) on the active WSG (WSG A).
- Step 3** Configure SVCLC in your SUP for this SAMI slot.

**Note**

If SVCLC is already configured in your SUP, first remove the SVCLC configuration before performing Step 2. Then, reconfigure the SVCLC as in Step 3. These steps ensure that when you configure the alias IP, the ARP broadcast doesn't reach the SUP.

- Step 4** Verify the HA VLAN connectivity.

**Caution**

Do not ping the alias IP to trigger an ARP broadcast.

- Step 5** Save the configuration and reload WSG B. The redundant WSGs will pair up and the standby WSG will sync the remaining configuration from the active WSG.

Example of High Availability Active-Standby Redundant Pairs

The following example shows how to configure high availability active-standby redundant pairs:

On SAMI A (slot 3/PPC3):

- Step 1** Under entity-all mode, configure HA VLAN interface.


```
ha interface vlan start-id 51 processor-count 2 increment 1
ip address start-ip 51.51.51.3 increment 1.1.1.0 mask 255.255.255.0
```

Step 2 Configure HA redundancy mode as active-standby with preferred-role set to primary.

```
ha redundancy-mode active-standby preferred-role primary revertive
```

Step 3 Save the configuration.

```
copy running-config startup-config
```

On SAMI B (slot 4/PPC3):

Step 4 Under entity-all mode, configure HA VLAN interface.

```
ha interface vlan start-id 51 processor-count 2 increment 1
ip address start-ip 51.51.51.4 increment 1.1.1.0 mask 255.255.255.0
```

Step 5 Configure HA redundancy mode as active-standby with preferred-role set to secondary.

```
ha redundancy-mode active-standby preferred-role secondary revertive
```

Step 6 Save the configuration.

```
copy running-config startup-config
```

Step 7 Reload both SAMI A and B.

Example of High Availability Active-Active Redundant Pairs

The following example shows how to configure high availability active-active redundant pairs:

On SAMI A (slot 3/PPC3):

Step 1 Under the entity-all mode, configure the HA VLAN interface.

```
ha interface vlan start-id 51 processor-count 2 increment 1
ip address start-ip 51.51.51.3 increment 1.1.1.0 mask 255.255.255.0
```

Step 2 Configure HA redundancy mode as active-active with preferred-role set to primary.

```
ha redundancy-mode active-active preferred-role primary revertive
```

Step 3 Save the configuration.

```
copy running-config startup-config
```

On SAMI B (slot 4/PPC3):

Step 4 Under the entity-all mode, configure HA VLAN interface.

```
ha interface vlan start-id 51 processor-count 2 increment 1
ip address start-ip 51.51.51.4 increment 1.1.1.0 mask 255.255.255.0
```

Step 5 Configure HA redundancy mode as active-active with preferred-role set to secondary.

```
ha redundancy-mode active-active preferred-role secondary revertive
```

Step 6 Save the configuration.

```
copy running-config startup-config
```

Step 7 Reload both SAMI A and B. Failure to do so results in incorrect HA configuration and deployment.

Step 8 When both SAMI A and B are reset at about the same time, the result is:

Slot 3	Slot 4
PPC3 (Active)	PPC3 (Standby)
PPC4 (Standby)	PPC4 (Active)

Step 9 If the reset didn't happen at the same time, the result is:

Slot 3	Slot 4
PPC3 (Active)	PPC3 (Standby)
PPC4 (Active)	PPC4 (Standby)

Step 10 Slot 3/PPC 4 will then reset due to the revertive option, so the redundant pair will become:

Slot 3	Slot 4
PPC3 (Active)	PPC3 (Standby)
PPC4 (Standby)	PPC4 (Active)

Role Revert After Failover

Network topology ensures that the traffic is distributed across the two IPSec gateways in order to avoid a single point of failure. During a failover scenario, traffic is switched to only the secondary IPSec gateway. When the failed primary card comes back, traffic still flows to only the secondary IPSec gateway. The WSG allows you to restore traffic flow to the primary IPSec gateway so that traffic remains distributed.

The procedure to revert back after a failover is as follows:

1. After a failover, the secondary card comes up as active.
2. When the primary card comes back up as the standby, IKE/IPSec data is synced to the standby card (which is still configured as the primary).
3. The secondary card that is active is then reset.
4. The primary card becomes active again. After the reset, the secondary card becomes standby again.



Note

In an active-active redundancy configuration, the revertive option is mandatory. In the case where an active PPC configured as primary revertive fails over, upon coming back up the PPC regains the active role from the redundant PPC.

Configuring Application VLAN/Alias IP Address

For each PPC processor on the SAMI, configure a VLAN with an IP address. This IP address is used by the IKE and ESP traffic from the endpoints. In case of redundancy, this is the same VLAN number configured on both the active and standby PPCs. The alias IP address is configured for a VLAN on both the active and standby PPCs. FAP/HNB uses the alias IP address instead of the active IP address. This alias IP address must be in the same subnet as the VLAN's active IP address. When a failover occurs, the newly active node starts receiving traffic destined to this alias IP address.



Note

Sharing of the same VLAN ID with different PPC's on SAMI WSG is not supported unless if it is for HA VLAN interface.

In the following example, WSG A is configured with public IP address 88.88.23.33, WSG B is configured with public IP address 88.88.23.34, and the alias IP address is 88.88.23.35. In this case, traffic is sent to the alias IP address, 88.88.23.35. If the active WSG A fails, the standby WSG B takes over, and the newly active WSG B keeps the same tunnel state with the same alias IP address.

The following example shows how to configure the alias IP address on two WSGs:

WSG A, Slot 1/PPC3:

```
WSG (config) # interface vlan 50
WSG (config-if) # ip address 88.88.23.33 255.255.255.0
WSG (config-if) # alias 88.88.23.35 255.255.255.0
```

WSG B, Slot 3/PPC3:

```
WSG (config) # interface vlan 50
WSG (config-if) # ip address 88.88.23.34 255.255.255.0
WSG (config-if) # alias 88.88.23.35 255.255.255.0
```

Configuring High Availability on SAMI COSLI

In the SAMI COSLI HA infrastructure, a cluster contains a pair of PPCs from two SAMIs, which can be on the same (intra-chassis) or different (inter-chassis) Cisco 7600 router chassis. The PPCs with the same number on a redundant pair of SAMIs are paired together (e.g. SAMI A, Slot 1/PPC3 is paired with SAMI B, Slot 2/PPC3). To accomplish this, configure a unique subnet for these two PPCs.

In the active-standby redundancy mode, for a redundant pair of SAMIs, there are 6 different subnets configured for 6 pairs of PPCs. Even though 6 pairs of PPCs between the two SAMIs are paired independently, all 6 PPCs on the same SAMI are assigned the same role (either active or standby). Configure the same preferred role (either primary or secondary) for all 6 PPCs on each SAMI. A failure from any PPC triggers a switchover to the other SAMI.

In the active-active redundancy mode, only the PPC3 and PPC4 pairs between the SAMIs are used.

There are two kinds of CLIs on COSLI for configuring high availability: node specific CLIs (e.g. the IP address of the node) or non-node specific CLIs (e.g. SNMP). All node specific CLIs have to be entered on each PPC. Non-node specific CLIs are not available when a SAMI is in standby mode. The HA infrastructure filters these CLIs out when syncing between a pair of PPCs.

Configuring the VLAN/IP Address for HA Infrastructure

The HA VLAN and IP addresses are used internally by the HA infrastructure to communicate among the nodes in the same cluster (subnet). To configure VLAN and IP addresses, perform the following steps:

	Command	Purpose
Step 1	WSG> enable	Enables privileged EXEC mode.
Step 2	WSG# configure terminal	Enters global configuration mode.
Step 3	WSG(config)# ha interface vlan <i>vlan_ID</i>	Configures the VLAN for HA functionality.
Step 4	WSG(config)# ip address <i>ip_address</i> <i>netmask</i>	IP address and subnet netmask for this interface.

**Note**

These CLI commands must be configured on each PPC. The two PPCs that are to be paired together must be configured to have the same VLAN ID. As a result, 6 different VLAN IDs are used for 6 pairs of PPCs. If you only need to configure site-to-site (S2S) tunnels, only PPC3 needs to be configured. The other PPCs are left unconfigured.

**Note**

Starting in WSG Release 4.0, S2S is supported on PPC3 only (HA active-standby mode) or PPC3 and PPC4 (HA active-active mode).

**Note**

You must also configure these VLANs on the SUP.

The following example shows how to configure the HA VLAN ID and IP addresses for PPC3:

On Slot 1/PPC3:

```
WSG(config)# ha interface vlan 611
WSG(config-if)# ip address 11.11.1.13 255.255.255.0
```

On Slot 3/PPC3:

```
WSG(config)# ha interface vlan 611
WSG(config-if)# ip address 11.11.1.23 255.255.255.0
```

Single Point Configuration of VLAN/IP Address for HA Infrastructure

To configure the VLAN and IP address using a single point configuration, perform the following steps:

	Command	Purpose
Step 1	WSG> enable	Enables privileged EXEC mode.
Step 2	WSG# configure terminal	Enters global configuration mode.
Step 3	WSG(config)# ha interface vlan start-id vlan_ID increment increment vlan_ID	Configures the VLAN using a single point configuration.
Step 4	WSG(config)# ip address start-ip ip_address increment increment ip_address netmask	IP address and subnet netmask for this interface.

These CLI commands are available in the entity-all mode on the director PPC (PPC3).

If you execute the following CLI commands on the PPC3:

```
WSG(mode-all)(config)# ha interface vlan start-id 212 increment 2
WSG(mode-all)(config-if)# ip address start-ip 11.11.1.11 increment 0.0.1.2 mask
255.255.255.0
```

The configurations of the 6 PPCs will be:

PPC3:

```
WSG(config)# ha interface vlan 212
WSG(config-if)# ip address 11.11.1.11 255.255.255.0
```

PPC4:

```
WSG(config)# ha interface vlan 214
WSG(config-if)# ip address 11.11.2.13 255.255.255.0
```

PPC5:

```
WSG(config)# ha interface vlan 216
WSG(config-if)# ip address 11.11.3.15 255.255.255.0
```

PPC6:

```
WSG(config)# ha interface vlan 218
WSG(config-if)# ip address 11.11.4.17 255.255.255.0
```

PPC7:

```
WSG(config)# ha interface vlan 220
WSG(config-if)# ip address 11.11.5.19 255.255.255.0
```

PPC8:

```
WSG(config)# ha interface vlan 222
WSG(config-if)# ip address 11.11.6.21 255.255.255.0
```

Configuring Redundancy-mode and Preferred-role

To configure the redundancy mode of the HA feature, perform the following steps:

	Command	Purpose
Step 1	WSG> enable	Enables privileged EXEC mode.
Step 2	WSG# configure terminal	Enters global configuration mode.
Step 3	WSG(config)# ha redundancy-mode {active-active active-standby} preferred-role {primary secondary} [revertive]	The preferred-role is used to indicate which node should come up as active (primary) or standby (secondary) when both nodes are rebooted at about the same time.

The **preferred-role** is used to indicate which node should come up as active (primary) or standby (secondary) when both nodes are rebooted at about the same time.

The following example shows how to configure a redundant pair of PPCs in active-standby redundancy mode:

On Slot 3/PPC3:

```
WSG(config)# ha redundancy-mode active-standby preferred-role primary
```

On Slot 4/PPC3:

```
WSG(config)# ha redundancy-mode active-standby preferred-role secondary
```



Note

These CLI commands are available only on PPC3. If executed in the **all** mode, the command is applied to all 6 PPCs—the same role is assigned to all 6 PPCs. If the command is executed in the **single** mode, it is applied only to PPC3, and the remaining 5 PPCs will have no redundancy mode configured. The SAMI that is configured with the preferred-role of **secondary** needs to be reset before the redundant pairs can take effect.

The following example shows how to configure a redundant pair of PPCs in active-active redundancy mode:

On Slot 3/PPC3:

```
WSG(config)# ha redundancy-mode active-active preferred-role primary revertive
```

On Slot 4/PPC3:

```
WSG(config)# ha redundancy-mode active-active preferred-role secondary revertive
```



Note

In the active-active redundancy mode, only the PPC3 and PPC4 pairs are used.

Removing HA Redundancy Between a Pair of PPCs

The following example shows how to remove a redundant pair of PPCs in active-standby redundancy mode:

On Slot 3/PPC3:

```
WSG(config)# no ha interface vlan 212
WSG(config)# no ha redundancy-mode active-standby preferred-role primary
```

On Slot 4/PPC3:

```
WSG(config)# no ha interface vlan 212
WSG(config)# no ha redundancy-mode active-standby preferred-role secondary
```

The following example shows how to remove a redundant pair of PPCs in active-active redundancy mode:

On Slot 3/PPC3:

```
WSG(config)# no ha interface vlan 212
WSG(config)# no ha redundancy-mode active-active preferred-role primary revertive
```

On Slot 4/PPC3:

```
WSG(config)# no ha interface vlan 212
WSG(config)# no ha redundancy-mode active-active preferred-role secondary revertive
```



Note

You must clean up the remaining (non-HA) configuration and bring the system back to operational state. The system will not be automatically rebooted as a result of removing the HA configuration.

Verifying the HA Configuration

Use the following commands to display information about the HA state at various levels:

	Command	Purpose
Step 1	WSG> enable	Enables privileged EXEC mode.
Step 2	WSG# show ha info brief show ha info show ha info detail	Displays various levels of HA information.

The **show ha info** command shows the configuration, states, and statistics of the local node and its peer:

```
WSG# show ha info
Redundancy mode (configured) : active-standby
Redundancy state : Redundant
My Node
  Current State : Active
  Preferred Role : Primary
  IP Address    : 51.51.51.43
  Slot/PPC     : 4/3
Peer Node
  IP Address    : 51.51.51.53
  Slot/PPC     : 5/3
Bulk Sync Status : Success
Bulk Sync done  : Thu Sep 15 01:24:36 2011
HA Revertive   : Disabled
```

```
S2P4# sh ha info
Redundancy mode (configured) : Active-Active
Redundancy state : Redundant
My Node
  Current State : Active
  Preferred Role : Primary
  IP Address    : 77.77.84.24
  Slot/PPC     : 2/4
Peer Node
  IP Address    : 77.77.84.34
  Slot/PPC     : 3/4
Bulk Sync Status : Success
Bulk Sync done  : Tue Jun 19 06:54:38 2012
HA Revertive   : Enabled
```

The **show ha info brief** command shows the configuration and the state of the local node:

WSG# **show ha info brief**

Interface	IP-Address	Redundancy-State	Mode	Current-State	Preferred-Role	HA-Revertive
VLAN51	51.51.51.43	Redundant	active-standby	Active	Primary	Disabled

S2P4# **show ha info brief**

Interface	IP-Address	Redundancy-State	Mode	Current-State	Preferred-Role	Revertive
VLAN2084	77.77.84.24	Redundant	Active-Active	Active	Primary	Enabled

The **show ha info detail** command includes extra information about the cluster and node names:

WSG# **show ha info detail**

Redundancy mode (configured) : active-standby

Redundancy state : Redundant

My Node

nodename : node1

Current State : Active

Last State : Un-assigned

Preferred Role : Primary

IP Address : 51.51.51.43

Slot/PPC : 4/3

Peer Node

nodename : node2

IP Address : 51.51.51.53

Slot/PPC : 5/3

Bulk Sync Status : Success

Bulk Sync done : Thu Sep 15 01:24:36 2011

HA Revertive : Disabled

ISync Counters

Total Request Sent : 0

Total Response Rcvd : 0

Total Fail Count : 0

Total Request Rcvd : 0

Total Response Sent : 0

Cluster : cluster12

Active Mgr : node1

Standby Mgr : node2


```
S2P4# show ha info detail
Redundancy mode (configured) : Active-Active
Redundancy state : Redundant
My Node
  nodename : node1
  Current State : Active
  Last State : Un-assigned
  Preferred Role : Primary
  IP Address : 77.77.84.24
  Slot/PPC : 2/4
Peer Node
  nodename : node2
  IP Address : 77.77.84.34
  Slot/PPC : 3/4
Bulk Sync Status : Success
Bulk Sync done : Tue Jun 19 06:54:38 2012
HA Revertive : Enabled
ISync Counters
Total Request Sent : 3
Total Response Rcvd : 3
Total Fail Count : 0
Total Request Rcvd : 0
Total Response Sent : 0
Cluster : cluster12
Active Mgr : node2
Standby Mgr : node1
```

Adding or Removing a Redundant Pair

The following sections describe how to configure, add, and remove redundant nodes:

- [How to Configure Active-Standby Redundancy Before Both SAMIs are in Service](#)
- [How to Configure Active-Active Redundancy Before Both SAMIs are in Service](#)
- [How to Add Standby WSG to an Active WSG Already in Service \(Active-Standby Mode\)](#)
- [How to Remove Standby WSG from an Active WSG Already in Service \(Active-Standby Mode\)](#)

How to Configure Active-Standby Redundancy Before Both SAMIs are in Service

Perform the following steps on the secondary SAMI:

-
- Step 1** Before the secondary SAMI is inserted, please do the following on the SUP:

- Remove the startup-config files for the secondary SAMI on the bootflash or bootdisk:

```
SUP-7600# del bootflash:SLOT2SAMIC*.cfg
```

- Remove the VLAN groups that are tied to the secondary SAMI:

```
SUP-7600# no svclc module 2 vlan-group 2,30,50,70
```

- (Inter-chassis only) Ensure the time is synced between the two SUPs.

Step 2 Insert the secondary SAMI.

Step 3 Add HA VLAN interface for each PPC:

```
switch(config)# ha interface vlan 611
switch(config-if) ip address 11.11.1.2 255.255.255.252
```

Step 4 Add redundancy-mode with preferred-role for each PPC using entity-all option from PPC3:

```
switch(mode-all)(config)# ha redundancy-mode active-standby preferred-role secondary
```

Step 5 Configure all node-specific commands for each PPC:

```
switch(config)# ip route 77.77.77.0 255.255.255.0 88.88.23.1
```

Step 6 Configure alias IP addresses for the WSG for each PPC:

```
switch(config)# interface vlan 50
switch(config-if)# ip address 88.88.23.35 255.255.255.0
switch(config-if)# alias 88.88.11.35 255.255.255.0
```

Step 7 Save the configuration for each PPC:

```
switch# copy running-config startup-config
```

Perform the following steps on the primary SAMI:

Step 1 Add HA VLAN interface for each PPC:

```
switch(config)# ha interface vlan 611
switch(config-if)# ip address 11.11.1.1 255.255.255.252
```

Step 2 Add redundancy-mode with preferred-role for each PPC using entity-all option from PPC3:

```
switch(mode-all)(config)# ha redundancy-mode active-standby preferred-role primary
```

Step 3 Configure all node-specific commands for each PPC:

```
switch(config)# ip route 77.77.77.0 255.255.255.0 88.88.23.1
```

Step 4 Configure alias IP addresses and all other commands for the WSG for each PPC:

```
switch(config)# interface vlan 50
switch(config-if)# ip address 88.88.23.34 255.255.255.0
switch(config-if)# alias 88.88.23.11 255.255.255.0
switch(config)# crypto profile "prof-1"
```

Step 5 Save the configuration for each PPC:

```
switch# copy running-config startup-config
```

Step 6 Check HA status:

```
switch# show ha info
Redundancy mode (configured) : active-standby
```

```

Redundancy state : Non-Redundant
My Node
  Current State : Active
  Preferred Role : Primary
  IP Address    : 11.11.1.1
  Slot/PPC     : 4/3
  Bulk Sync Status : Not-Initiated
  HA Revertive  : Enabled

```

The primary SAMI is now ready for service. There is no need to reboot.

Perform the following steps on the secondary SAMI:

Step 1 Reload the secondary SAMI. While it is booting back up, configure its VLAN groups on the SUP:

```
SUP-7600# svc1c module 2 vlan-group 2,30,50,70
```



Note If this command is not executed before the SAMI comes back up, the SAMI cannot come up as the standby. In this case, repeat Step 1.

Step 2 After the SAMI is back up and running, check HA status:

```

switch# sh ha info
  Redundancy mode (configured) : active-standby
  Redundancy state : Redundant
  My Node
    Current State : Standby
    Preferred Role : Secondary
    IP Address    : 11.11.1.2
    Slot/PPC     : 2/3
  Peer Node
    IP Address    : 11.11.1.1
    Slot/PPC     : 4/3
  Bulk Sync Status : Success
  Bulk Sync done   : Tue May 25 00:13:31 2010
  HA Revertive    : Enabled

```

If the secondary SAMI is not in the standby state, check the following:

- Ensure VLAN groups for this SAMI are added to the SUP
- Ensure preferred roles are configured correctly on both cards
- Ensure the peer's HA IP address is reachable

Step 3 Check whether WSG CLIs are synced from the active card:

```
switch# sh running-config
```

How to Configure Active-Active Redundancy Before Both SAMIs are in Service

Perform the following steps on the secondary SAMI:

Step 1 Before the secondary SAMI is inserted, please do the following on the SUP:

- Remove the startup-config files for the secondary SAMI on the bootflash or bootdisk:

```
SUP-7600# del bootflash:SLOT2SAMIC*.cfg
```

- Remove the VLAN groups that are tied to the secondary SAMI:

```
SUP-7600# no svclc module 2 vlan-group 2,30,50,70
```

- (Inter-chassis only) Ensure the time is synced between the two SUPs.

Step 2 Insert the secondary SAMI.

Step 3 Add HA VLAN interface for each PPC:

```
switch(config)# ha interface vlan 611
switch(config-if) ip address 11.11.1.2 255.255.255.252
```

Step 4 Add redundancy-mode with preferred-role for each PPC using entity-all option from PPC3:

```
switch(mode-all)(config)# ha redundancy-mode active-active preferred-role secondary
revertive
```

Step 5 Configure all node-specific commands for each PPC:

```
switch(config)# ip route 77.77.77.0 255.255.255.0 88.88.23.1
```

Step 6 Configure alias IP addresses for the WSG for each PPC:

```
switch(config)# interface vlan 50
switch(config-if)# ip address 88.88.23.35 255.255.255.0
switch(config-if)# alias 88.88.11.35 255.255.255.0
```

Step 7 Save the configuration for each PPC:

```
switch# copy running-config startup-config
```

Perform the following steps on the primary SAMI:

Step 1 Add HA VLAN interface for each PPC:

```
switch(config)# ha interface vlan 611
switch(config-if)# ip address 11.11.1.1 255.255.255.252
```

Step 2 Add redundancy-mode with preferred-role for each PPC using entity-all option from PPC3:

```
switch(mode-all)(config)# ha redundancy-mode active-active preferred-role primary
revertive
```

Step 3 Configure all node-specific commands for each PPC:

```
switch(config)# ip route 77.77.77.0 255.255.255.0 88.88.23.1
```

Step 4 Configure alias IP addresses and all other commands for the WSG for each PPC:

```
switch(config)# interface vlan 50
switch(config-if)# ip address 88.88.23.34 255.255.255.0
switch(config-if)# alias 88.88.23.11 255.255.255.0
switch(config)# crypto profile "prof-1"
```

Step 5 Save the configuration for each PPC:

```
switch# copy running-config startup-config
```

Step 6 Check HA status:

```

S2P4# sh ha info
Redundancy mode (configured) : Active-Active
Redundancy state : Redundant
My Node
  Current State : Active
  Preferred Role : Primary
  IP Address : 11.11.1.1
  Slot/PPC : 2/4
Peer Node
  IP Address : 11.11.1.2
  Slot/PPC : 3/4
Bulk Sync Status : Success
Bulk Sync done : Tue Jun 19 06:54:38 2012
HA Revertive : Enabled

```

The primary SAMI is now ready for service. There is no need to reboot.

Perform the following steps on the secondary SAMI:

Step 1 Reload the secondary SAMI. While it is booting back up, configure its VLAN groups on the SUP:

```
SUP-7600# svc1c module 2 vlan-group 2,30,50,70
```



Note If this command is not executed before the SAMI comes back up, the SAMI cannot come up as the standby. In this case, repeat Step 1.

Step 2 After the SAMI is back up and running, check HA status:

```

S2P4# sh ha info
Redundancy mode (configured) : Active-Active
Redundancy state : Redundant
My Node
  Current State : Standby
  Preferred Role : Secondary
  IP Address : 11.11.1.2
  Slot/PPC : 3/4
Peer Node
  IP Address : 11.11.1.1
  Slot/PPC : 2/4
Bulk Sync Status : Success
Bulk Sync done : Tue Jun 19 06:54:38 2012
HA Revertive : Enabled

```

If the secondary SAMI is not in the standby state, check the following:

- Ensure VLAN groups for this SAMI are added to the SUP
- Ensure preferred roles are configured correctly on both cards
- Ensure the peer's HA IP address is reachable

Step 3 Check whether WSG CLIs are synced from the active card:

```
switch# sh running-config
```

How to Add Standby WSG to an Active WSG Already in Service (Active-Standby Mode)

Perform the following step on the active SAMI:

Step 1 Ensure the active SAMI is not paired with another SAMI:

```
switch# show ha info
Redundancy mode (configured) : active-standby
Redundancy state : Non-Redundant
My Node
  Current State : Active
  Preferred Role : Primary
  IP Address : 11.11.1.1
  Slot/PPC : 4/3
  Bulk Sync Status : Not-Initiated
  HA Revertive : Enabled
```

If it has a peer SAMI, follow the procedure in the section below to remove the standby WSG.

The preferred-role should be set to primary. If not, set it to primary using the **ha redundancy-mode** command (this will not impact service):

```
switch(mode-all)(config)# ha redundancy-mode active-standby preferred-role primary
```

Perform the following steps on the secondary SAMI:

Step 1 Before the secondary SAMI is inserted, please do the following on the SUP:

- Remove the startup-config files for the secondary SAMI on the bootflash or bootdisk:

```
SUP-7600# del bootflash:SLOT2SAMIC*.cfg
```

- Remove the VLAN groups that are tied to the secondary SAMI:

```
SUP-7600# no svcl module 2 vlan-group 2,30,50,70
```

- (Inter-chassis only) Ensure the time is synced between the two SUPs.

Step 2 Insert the secondary SAMI.**Step 3** Add HA VLAN interface for each PPC:

```
switch(config)# ha interface vlan 611
switch(config-if) ip address 11.11.1.2 255.255.255.252
```

Step 4 Add redundancy-mode with preferred-role for each PPC using entity-all option from PPC3:

```
switch(mode-all)(config)# ha redundancy-mode active-standby preferred-role secondary
```

Step 5 Configure all node-specific commands for each PPC:

```
switch(config)# ip route 77.77.77.0 255.255.255.0 88.88.23.1
```

Step 6 Configure alias IP addresses for the WSG for each PPC:

```
switch(config)# interface vlan 50
switch(config-if)# ip address 88.88.23.35 255.255.255.0
switch(config-if)# alias 88.88.11.35 255.255.255.0
```

Step 7 Save the configuration for each PPC:

```
switch(config)# copy running-config startup-config
```

Step 8 Reload the secondary SAMI. While it is booting back up, configure its VLAN groups on the SUP:

```
SUP-7600# svcl module 2 vlan-group 2,30,50,70
```



Note If this command is not executed before the SAMI comes back up, the SAMI cannot come up as the standby. In this case, repeat Step 1.

Step 9 After the SAMI is back up and running, check HA status:

```
switch# sh ha info
Redundancy mode (configured) : active-standby
Redundancy state : Redundant
My Node
  Current State : Standby
  Preferred Role : Secondary
  IP Address    : 11.11.1.2
  Slot/PPC     : 2/3
Peer Node
  IP Address    : 11.11.1.1
  Slot/PPC     : 4/3
Bulk Sync Status : Success
Bulk Sync done   : Tue May 25 00:13:31 2010
HA Revertive    : Enabled
```

If the secondary SAMI is not in the standby state, check the following:

- Ensure VLAN groups for this SAMI are added to the SUP
- Ensure preferred roles are configured correctly on both cards
- Ensure the peer's HA IP address is reachable

Step 10 Check whether WSG CLIs are synced from the active card:

```
switch# sh running-config
```

How to Remove Standby WSG from an Active WSG Already in Service (Active-Standby Mode)

Perform the following steps on the secondary SAMI:

Step 1 Ensure the secondary SAMI is in the standby state and paired with an active WSG:

```
switch# sh ha info
Redundancy mode (configured) : active-standby
Redundancy state : Redundant
My Node
  Current State : Standby
  Preferred Role : Secondary
  IP Address    : 11.11.1.2
  Slot/PPC     : 2/3
Peer Node
  IP Address    : 11.11.1.1
  Slot/PPC     : 4/3
Bulk Sync Status : Success
Bulk Sync done   : Tue May 25 00:13:31 2010
HA Revertive    : Enabled
```

Step 2 Remove the VLAN groups that are tied to this SAMI from the SUP:

```
SUP-7600# no svc1c module 2 vlan-group 2,30,50,70
```

Step 3 Remove the VLAN interfaces that have alias IP address configured for each PPC:

```
switch(config)# no interface vlan 50
```

Step 4 Remove HA VLAN interface for each PPC:

```
switch(config)# no ha interface vlan 611
```

Step 5 Remove redundancy-mode with preferred-role for each PPC using entity-all option from PPC3:

```
switch(mode-all)(config)# no ha redundancy-mode active-standby preferred-role
secondary
```

Step 6 Save the configuration for each PPC:

```
switch(config)# copy running-config startup-config
```

WSG Deployment Modes

Site-to-site or remote access tunnels can be established using active-active HA redundancy mode, but you must switch between WSG deployment modes. The two deployment modes are site-to-site and remote-access. Switch between the two modes by first deactivating all profiles and reloading the SAMIs. Once the SAMIs are back up, activate the particular profile. Depending on the profile type, only tunnels of that type can be established while being in that deployment mode. Verify the deployment mode using the **show crypto deployment-mode** CLI command.

Bulk Sync

Bulk sync procedure involves the configuration sync of the standby SAMI with the active SAMI, when a card running with the **no** redundancy scheme is configured for redundancy, and it subsequently receives a standby notification.

The configuration sync done on the standby card is a two part sync procedure; the first part involves syncing the running configuration from the active card, and the second part consists of syncing the startup configuration of the active card. These two config sync are autonomous in operation. The module that is responsible for carrying out this bulk sync procedure is the configuration controller.



Note

In the active-active HA redundancy mode, bulk sync takes place between the PPC pairs and not the SAMI pairs.

Bulk Sync Procedure

When the config-controller is assigned the active role, it performs the following actions:

- If assigned the active role, it parses the startup-config file and applies all the commands except the HA-specific CLIs (HA-specific commands are applied during bootup before the role is assigned).

If the config-controller is assigned a standby role, it will perform the following actions:

- If it is assigned the standby role, it sends a bulk-sync request to the active peer PPC.
- Upon receiving the bulk-sync request, the config-controller on the active peer strips the node-specific CLIs from the running-configuration, and transfers them to the standby card.
- The standby card applies those CLIs.
- After it completes the running-configuration, the config-controller performs the same procedure on the startup-configuration. The standby card merges it with its own node-specific commands in the startup-configuration, and saves it to the SUP.
- Notifies the HA manager that it is now Standby-Ready.

In active-active HA redundancy mode, bulk sync takes place between PPC pairs and not between SAMI pairs.

Incremental Sync

When a non-node-specific command is applied to the active card, the configuration needs to be sent to the standby peer, if it exists. A registered callback function for that command is invoked on the standby peer to apply it locally.

A message is sent to the standby peer when a **copy running-config startup-config** is applied on the active card. The startup-config file for the standby peer is saved on the SUP.

Failover

When a fatal error is detected on a node of the active card, the process is terminated due to the setting in the information model. Since the HA configuration for the system manager is set to **reset** for the restart option, it causes the SAMI to reboot. The standby card gets notification from the cluster manager to go active. The newly-active node configures the alias IP address and sends an ARP announcement for this alias IP address, if it is configured.

When a fatal error is detected on a node of the standby card, the standby card reboots.

In active-active HA redundancy mode, failover can occur between PPC pairs without affecting other PPC pairs on the SAMIs involved.

Configuring Revert Back After Failover

Reverting back after failover may not be required for all the deployment scenarios, so this functionality is configurable.

In WSG releases prior to 4.0, perform the following tasks to enable the revert-back feature:

	Command	Purpose
Step 1	WSG> enable	Enables privileged EXEC mode.
Step 2	WSG# configure terminal	Enters global configuration mode.
Step 3	WSG(config)# ha revertive	Resets the active card on secondary. This ensures that card that is configured as primary always has the state as active, and the secondary card has the standby state. This is a non-node specific command so that it is also synched across the standby.

Starting in WSG Release 4.0, the **ha revertive** functionality is replaced by a configurable option in the **ha redundancy-mode** command using the **revertive** keyword.

	Command	Purpose
Step 1	WSG> enable	Enables privileged EXEC mode.

	Command	Purpose
Step 2	WSG# configure terminal	Enters global configuration mode.
Step 3	WSG(config)# ha redundancy-mode {active-standby active-active} preferred-role {primary secondary} [revertive]	Resets the active card on the secondary to ensure that the primary card has the active state and the secondary card has the standby state. The revertive keyword is optional for the active-standby mode but required for the active-active mode.

The failover revert-back configuration is displayed in the output of **show ha info** and **show ha info brief** commands. Follow these steps to revert back after a failover:

1. Failover occurs on SAMI 1 (preferred-role = primary, state = active).
2. SAMI 2 (preferred-role = secondary, state = standby) transitions to active state.
3. SAMI 1 comes up again, and COSLI bulk sync occurs with SAMI 2.
4. WSG also bulk syncs all of its data with SAMI 2. Once the bulk sync is complete, the WSG application on SAMI 2 indicates this to the configuration controller using a new MTS message.
5. The configuration controller FSM handles this new event from the WSG in its event handler of active state.
6. In this event handler, the configuration controller first checks if the **ha revertive** command is configured. If command is not configured, no action is taken. Otherwise, step 7 is performed.
7. In case the configured **preferred-role** of the node is **secondary** and the HA state is active, SAMI 2 reloads.
8. SAMI 1 transitions to active state, and SAMI 2 comes up in standby state.

IKE SA Handling

IKE SA Create

IKE SAs are created on the standby WSG when IKE SAs are created on the active WSG.

IKE SA Update

IKE SAs are updated on the standby when IKE SAs are updated on the active for following reasons:

- IKE SA window has been updated due to initiator or responder exchange
- Remote access attributes have been set
- NAT reboot has been detected and IKE SA remote address or port has changed

IKE SA Rekey

Rekeyed IKE SA is imported, and the old IKE is deleted on the standby WSG when the IKE SA is rekeyed on the active WSG.

IKE SA Delete

IKE SAs are deleted on the standby WSG when the IKE SAs are deleted on the active WSG.

IKE Message ID

The WSG synchronizes IKE Message IDs between the active and standby WSGs, otherwise IKE SAs are unusable. If an informational exchange like DPD is performed after the SA is imported on the standby, the SA needs to be updated.

The WSG maintains the DPD initiation/response successfully after failover. Additionally, the WSG maintains IKE parameters (like encryption and hashing Algorithms), and Diffie-Hellman groups after failover.

IKE SA Life Time

The WSG maintains the Phase 1 lifetime value instead of resetting on the newly active after failover.

IKE NAT Keepalives

The WSG maintains the different NAT states after failover. NAT keepalives are still successfully initiated and responded to after failover, and on time.

IPSec SA Handling

IPSec SA Create

IPSec SAs are created on the standby WSG when IPSec SAs are created on the active WSG.

IPSec SA Update

IPSec SAs are updated on the standby when IPSec SAs are updated on the active for following reason:

- NAT reboot is detected for the parent IKE SA, and the peer address or port has changed.

IPSec SA Rekey

Rekeyed IPSec SA are imported, and old IPSec is deleted on the standby WSG when the IPSec SA is rekeyed on the active WSG.

IPSec SA Delete

IPSec SAs are deleted on the standby WSG when the IPSec SAs are deleted on the active WSG.

IPSec SA Parameters

WSG maintains IPSec parameters protocol, encryption and authentication algorithms, PFS groups, anti-replay window size, and sequence numbers both 32 bit and 64 bit (ESN). It also maintains UDP encapsulation after failover.

IPSec SA Life Time

The WSG maintains the Phase 1 lifetime value instead of resetting on the new active after failover.

IPSec Outbound SA Sequence Numbers

The Sequence number is incremented for each data packet transmitted. 32 bit Sequence number is transmitted in the ESP header of each packet. In case of ESN, only the lower 32 bit sequence number is transmitted. The high-order 32 bits are maintained as part of the sequence number counter by both transmitter and receiver, and are included in the computation of the ICV. If the receiver receives a sequence number lower than the expected number, the packets may be dropped depending on Anti-replay parameters.

The WSG on standby updates the outbound sequence number of the IPSec SA to the estimated value, otherwise the sequence number goes out sync.

IKE and IPSec SAs are imported to the standby when it comes up. During import, the policy manager calls the fast API to Program Nitrox and IXP modules for inbound and outbound SAs. Outbound sequence number is the estimated sequence number based on IMIX or 64-byte traffic.

The sequence number is updated from the active to the standby at periodic intervals. The policy manager triggers the fast API to Program Nitrox and IXP modules for each periodic update.

seqnumber = active sequence number + estimated sequence number (packets processed for 5 minutes)

IPSec SA Replay Window

The WSG on the standby updates inbound anti-replay window base and mask of IPSec SA to prevent replay attack. The base value is the highest sequence number that has been received so far. This will limit the number of packets that can be replayed after a failover.

The Anti-replay window base and mask is updated periodically from the active to the standby at regular intervals.

When the standby comes up, IKE and IPSec SAs are imported to the standby. During import, the policy manager calls fast API to Program Nitrox and IXP modules for inbound and outbound SAs.

Certificate, CRL, DNS Handling

The WSG maintains certificate status between the the active and standby (Local Certificate, private keys and trustpoint certificates should be synced between the redundant pair) by using sync message or export and import mechanism. When a new standby WSG is inserted, the static certificates are synced to the standby. WSG does not maintain the DNS and CRL cache across failover. The DNS and CRLs are cached on the new active during new tunnel establishment.

Configuring IPSec

To protect addresses to which traffic is allowed from the tunnel, perform the following tasks:

	Command	Purpose
Step 1	switch# config	Enters global configuration mode.
Step 2	switch (config)# crypto profile P1 switch(config-crypto-profile)# ipsec	Enters IPSec submodule.

	Command	Purpose
Step 3	<pre>switch(config-crypto-profile-ipsec)# remote-access: access-permit ip ip-address subnet subnet site-to-site: access-permit rule name protocol {any sctp udp tcp} [src-ip start src ip end src ip src-port start src port end src port dst-ip start dst ip end dst ip dst-port start dst port end dst port]</pre>	<p>Configures the protected IP address to which traffic is allowed from a remote access tunnel, or traffic selectors and multiple child SA features for site-to-site tunnels.</p> <p>In the 1.2 Release, the <i>rule name</i> argument is added, and applies to site-to-site type profiles only. The remote-access type profile still accepts the short access-permit syntax.</p>
Step 4	<pre>switch(config-crypto-profile-ipsec)# transform-set esp aes256 aes-xcbc</pre>	<p>Configures the Encapsulating Security Payload (ESP) encryption and hash type. ESP is a security protocol that gives data privacy services, data authentication, and anti-replay services. ESP encapsulates data to be protected.</p>
Step 5	<pre>switch(config-crypto-profile-ipsec)# pfs {group1 group2 group5 group14 group15 group16 group17 group18}</pre>	<p>Sets a Perfect Forward Secrecy (PFS) group ID to use for negotiations during a new SA exchange</p>
Step 6	<pre>ip address-pool myPool</pre>	<p>Sets the address-pool name to be used in this profile.</p> <p>Note This step is optional for site-to-site profiles.</p>

```
WSG# config
Enter configuration commands, one per line. End with CNTL/Z.
WSG (config)# crypto profile remote-access
WSG (config-crypto-profile)# ipsec
WSG (config-crypto-profile-ipsec)# access-permit ip 100.1.3.0 subnet 24
```

Here is an example of the new **access-permit** command with the **protocol** options:

```
access-permit nameA
  protocol udp src-ip 12.12.12.1 12.12.12.20 src-port 23 23 dst-ip 0.0.0.0
255.255.255.255 dst-port 0 65535
  protocol tcp src-ip 12.12.12.1 12.12.12.20 src-port 23 23 dst-ip 0.0.0.0
255.255.255.255 dst-port 0 65535

access-permit nameB
  protocol any src-ip 13.13.13.1 13.13.13.20 src-port 23 23 dst-ip 0.0.0.0
255.255.255.255 dst-port 0 65535
```

Site-to-Site Scalability

In site-to-site (S2S) scenario, tunnels are established between the WSG and the remote peer just like in a remote access scenario. The source of the traffic inside the tunnel is from multiple IP addresses on the remote peer's side. As in the remote access scenario, the addresses on the trusted network behind the WSG can be a single IP or a network.

Additionally, in a S2S scenario, the remote peer can setup multiple tunnels to the WSG.

The TS can contain the protocol, source port range and destination port range, in addition to the source and destination IP range.

In case of site-to-site type tunnels, the WSG can initiate the tunnels to the peer device.



Note

DHCP is supported for RAS profiles and not for site-to-site profiles.

Scalability and Throughput Improvement Description

In previous WSG releases, the S2S traffic selector lookup was done by looking up an array of TS on the IXP. This linear search limited the performance of the site-to-site traffic selector lookup algorithm. In WSG Release 2.0 and above, the traffic selector lookup algorithm speeds up the performance for site-to-site. There is no change to the remote access traffic selector lookup because it is different from the lookup algorithm for site-to-site, and is already optimized.

The new site-to-site traffic selector lookup is based on a hash lookup of the packet's source and destination IP addresses after applying a network mask.

The list of source-mask and destination-mask combinations needs to be provided by the user through the CLI. The size of this list is limited to 6 entries. For best performance, the subnet combination that carries the most traffic needs to be configured on the top of the list.

The subnet combination values need to be figured out during network design and configured initially. Graceful addition, modification, and deletion of entries is not supported once the S2S profiles are activated. To modify the subnet combination configuration, deactivate all S2S profiles, change the subnet combination configuration, and then reactivate the S2S profiles. All established tunnels are lost during this procedure.

The **access-permit** command under site-to-site profile now accepts a netmask instead of an end IP address. The access-permit network configured for a profile can be larger than what is negotiated by a remote peer. This allows multiple peers to connect to the same site-to-site profile, and negotiate a subnet of the configured network. The TS negotiated by the peers of a WSG must be unique (no overlap).

The negotiated TS for the tunnels must be subnets, and not arbitrary ranges.

When a peer negotiates the TS with WSG, it must intersect one of the configured entries in the subnet combination table. If not, the negotiation is rejected. Debug and syslog messages are generated if tunnel negotiation fails under this condition.

TS negotiations based on protocol and port are also supported, but algorithm improvements only take effect when the different child SAs have TS that have unique source and destination IP subnets. There is a performance decrease if there are child SAs that differ only by port or protocol in their TS.

Upto 16666 S2S tunnels are supported per SAMI. S2S tunnels can only be configured on the director PPC.

WSG Release 2.0 and above does not support the IKE protocol that allows a peer to negotiate multiple TSs for the same S2S tunnel. Each S2S tunnel can negotiate only one TS. All other features that are currently supported for site-to-site and remote access are maintained in this release.

Configuring Scalability and Throughput

To configure the WSG to accept the netmask for the source and destination IP address, perform the following tasks:

	Command	Purpose
Step 1	WSG# conf	Enters global configuration mode.

	Command	Purpose
Step 2	WSG (config)# crypto profile P1 WSG(config-crypto-profile)# ipsec	Enters IPsec submode.
Step 3	WSG(config-crypto-profile-ipsec)# access-permit rule name protocol {any sctp udp tcp} [src-ip src ip/subnet size src-port start src port end src port dst-ip dst ip/subnet size dst-port start dst port end dst port]	Configures the protected IP address to which traffic is allowed from a remote access tunnel, or traffic selectors and multiple child SA features for site-to-site tunnels. The <i>src ip/ subnet size</i> and <i>dst ip/subnet size</i> arguments apply to only site-to-site type profiles. The remote-access type profile still accepts the short access-permit syntax.

Here is an example of the configuration:

```
WSG(config-crypto-profile-ipsec)#
access-permit nameA
    protocol udp src-ip 12.12.0.0 16 src-port 23 23 dst-ip 10.10.10.0 24 dst-port 0 65535

access-permit nameB
    protocol any src-ip 13.13.13.0 24 src-port 23 23 dst-ip 44.44.44.44 32 dst-port 0
65535
```

Configuring Subnet Combination

This command assists the IXP for site-to-site performance improvement. This configuration is made based on the design of the network.

It is mandatory to enter one or more of this command before activating any S2S profiles. S2S profile cannot be activated if this command is not configured on the WSG.

To enable this feature, perform the following tasks:

	Command	Purpose
Step 1	WSG# config	Enters global configuration mode.
Step 2	WSG (config)# crypto profile P1	Enters IPsec submode.
Step 3	WSG(config)# crypto site-to-site-lookup priority priority source-netmask src-netmask destination-netmask dst-netmask	Configures the list of source-mask and destination-mask combinations. <i>priority</i> —Priority of this lookup. The range is 1 to 6 <i>src-netmask</i> —Source IP network mask in format N. The N subnet mask format is increased from 0-32 to 0-128 for IPv6. <i>dst-netmask</i> —Destination IP network mask in format N. The N subnet mask format is increased from 0-32 to 0-128 for IPv6.

Here is an example of the configuration:

```
crypto site-to-site-lookup priority 5 source-netmask 112 destination-netmask 112
```

Certificate Management Protocol

WSG releases prior to Release 2.0 can generate a private key, a certificate request and support manual enrollment with CA server. WSG Release 2.0 introduced support for Certificate Management Protocol (CMPv2). CMP allows for automatic enrollment with CA server. The Keypair is generated locally and the certificate request can be sent to CA server using existing network connectivity. The certificate can be downloaded without the need for manual intervention.

Only one outstanding CMPv2 initialize, enroll, or update request is permitted at any time from a WSG. You can place a new request after the certificate is obtained for the outstanding request. Alternatively, you can clear the pending request and place a new request. The certificate and the private key will be saved on the SUP. If the SUP is redundant, the files are also copied to the redundant SUP.

In case of the WSG HA, the certificate is copied over to the SUP on the redundant chassis only when the certificate configuration command is entered. If the SUP on the secondary chassis is redundant, the certificate is also copied to the redundant SUP.

Files written to the SUP storage cannot be deleted using the WSG CLI. They need to be deleted using the SUP CLI. Revoke and recover functions are not supported in WSG Release 2.0 and above.

**Note**

A pending CMPv2 request from the WSG is not saved across reboots or WSG switchover. Do not reboot the WSG when a request is pending. If the WSG is rebooted or switched over during a pending request, the request must be reinitiated.


This feature only brings in an additional method of obtaining the certificates for WSG using new EXEC commands. The original manual enrollment process can still be used.


WSG Release 3.0 introduced support for automatic renewal of certificates, which includes automatic update and automatic retrieve. The automatic update of CMPv2 certificates is similar to the WSG CMP update command, but is performed by the system when the certificate is within a specified window before expiring (2 to 60 days). The automatic retrieval of certificates retrieves an updated certificate from the SUP when the certificate is within a specified window before expiring (2 to 60 days). Up to 20 certificates may be configured for automatic renewal.

Prior to Cisco 7600 WSG Release 4.4, WSG supported only TCP as the CMPv2 Transport Protocol. With Release 4.4 and above, WSG will support both transport protocols, TCP and HTTP. The HTTP based flows will be RFC 4210/4211/6712 compliant.

Configuring Certificate Management Protocol

To configure the WSG to generate the private key and make an initialize request to the CA server using CMPv2, perform the following tasks. The filenames for the private key and the certificate files will be automatically generated by the system.

	Command	Purpose
Step 1	<pre>WSG(config)# crypto cmp transport transport protocol</pre>	<p>Configures the Transport Protocol for CMPv2 Messages.</p> <p><i>transport protocol</i>—Transport Protocol options are <i>http</i>, and <i>tcp</i>.</p> <p><i>http</i>—HTTP will be used as transport Protocol for all CMPv2 messages.</p> <p><i>tcp</i>—TCP will be used as transport Protocol for all CMPv2 messages.</p> <p>The default is <i>tcp</i>.</p>
Step 2	<pre>WSG# crypto cmp initialize modulus modulus id-type id-type id id subject-name subject string ca-psk reference-number:key ca-root root certificate ca-url url</pre>	<p>Configures the WSG to generate the private key and make an initialize request to the CA server using CMPv2. This request for the client's initial certificate is authenticated using the reference number and corresponding PSK received from the CA.</p> <p><i>modulus</i>—Modulus of the generated certificate: 512, 1024, or 2048.</p> <p><i>id-type</i>—Type of the ID: fqdn or ip.</p> <p><i>id</i>—Word that is of id-type.</p> <p><i>subject string</i>—Subject string of the certificate in double quotes.</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p> Note The supported characters while configuring the subject-name are dash, dot, underscore, a-z, A-Z and 0-9. The maximum size supported for the string is 256 bytes.</p> </div> <p><i>reference-number:key</i>—PSK provided by the CA server for CMPv2 operation.</p> <p><i>root certificate</i>—Filename of the root certificate of the CA server (file present on SUP disk).</p> <p><i>url</i>—URL where the CA server listens to requests.</p>

Command	Purpose
<p>Step 3</p> <pre>WSG# crypto cmp enroll current-wsg-cert wsg_certificate current-wsg-private-key wsg_privatekey modulus modulus id-type id-type id id subject-name subject string ca-root root certificate ca-url url</pre>	<p>Makes an enroll request to the CA server using CMPv2. The existing WSG certificate and private key are user provided as input parameters to the CLI. The filenames for the new private key and the certificate files are automatically generated by the system. This request is similar to initialize except that it is authenticated using public-key methods.</p> <p><i>wsg_certificate</i>—Current valid WSG certificate.</p> <p><i>wsg_privatekey</i>—Current valid private key corresponding to the certificate provided in the previous parameter.</p> <p><i>modulus</i>—Modulus of the generated certificate: 512, 1024, or 2048.</p> <p><i>id-type</i>—Type of the ID: fqdn or ip.</p> <p><i>id</i>—Word that is of id-type.</p> <p><i>subject string</i>—Subject string of the certificate in double quotes.</p> <div data-bbox="1045 898 1091 940" style="text-align: center;">  </div> <p>Note The supported characters while configuring the subject-name are dash, dot, underscore, a-z, A-Z and 0-9. The maximum size supported for the string is 256 bytes.</p> <p><i>root certificate</i>—Filename of the root certificate of the CA server (file present on SUP disk).</p> <p><i>url</i>—URL where the CA server listens to requests.</p>
<p>Step 4</p> <pre>WSG# crypto cmp update current-wsg-cert wsg_certificate current-wsg-private-key wsg_privatekey ca-root root certificate ca-url url</pre>	<p>Sends an update request to the CA server using CMPv2 to update the existing WSG certificate. The existing WSG certificate and private key is provided by the user as input parameters to the CLI. The filenames for the new private key and the certificate files are automatically generated by the system.</p> <p><i>wsg_certificate</i>—Current valid WSG certificate.</p> <p><i>wsg_privatekey</i>—Current valid private key corresponding to the certificate provided in the previous parameter.</p> <p><i>root certificate</i>—Filename of the root certificate of the CA server (file present on SUP disk).</p> <p><i>url</i>—URL where the CA server listens to requests.</p>

Here is an example of the **crypto cmp transport** command:

```
crypto cmp transport http
```

Here is an example of the **crypto cmp initialize** command:

```
crypto cmp initialize modulus 1024 id-type fqdn id wsg.cisco.com subject-name
"C=US,O=Cisco,OU=Security,CN=Example" ca-psk 32438:this_is_very_secret ca-root root-ca.crt
ca-url http://212.246.144.35:8700/pkix/
```

Here is an example of the **crypto cmp enroll** command:

```
crypto cmp enroll current-wsg-cert wsg.crt current-wsg-private-key wsg.prv modulus 1024
id-type fqdn id wsg.cisco.com subject-name "C=US,O=Cisco,OU=Security,CN=Example" ca-root
root-ca.crt ca-url http://212.246.144.35:8700/pkix/
```

Here is an example of the **crypto cmp update** command:

```
crypto cmp update current-wsg-cert wsg.crt current-wsg-private-key wsg.prv ca-root
root-ca.crt ca-url http://212.246.144.35:8700/pkix/
```

The CA server may not immediately return the certificate. In this case, periodically use the **crypto cmp poll** command to check for availability. These commands are used if the request for a Privileged EXEC CMPv2 configuration command is pending.

Command	Purpose
WSG# show crypto cmp request	Displays the current status of the last CMPv2 request with an outstanding update request having priority over initialize and enroll requests. This shows the request that will be polled by the crypto cmp poll command. The output also indicates if no request is pending.
WSG# crypto cmp poll	Configure the WSG to query the CA server for the availability of the certificate requested by a previous crypto cmp initialize , enroll , or update command.
WSG# clear crypto cmp	Clears outstanding initialize, enroll, and update requests generated by this WSG. This allows you to make another initialize, enroll, or update request before the previous request is honored. No cancellation is sent to the CA server; only the state of the previous request on the WSG is cleared.

To configure the WSG to use the certificate, use the following configuration commands. The filenames for the private key and certificate files will be the same filenames generated using the Privileged EXEC CMP initialize and enroll commands.

	Command	Purpose
Step 1	WSG# config	Enters global configuration mode.
Step 2	WSG(config)# crypto pki wsg-cert <i>cert-filename.crt</i> [wsg-private-key <i>private-key-filename.prv</i>]	Configures the WSG to use the certificate for certificate based authentication. The certificate and private key are copied from the Cisco 7600 SUP to the WSG. Up to 20 certificate/private key pairs may be configured on the WSG. <i>cert-filename.crt</i> —Name of the WSG certificate file. Ensure certificate filenames end with a .crt file extension. <i>private-key-filename.prv</i> —Private key filename. To use the private key, set the name of the private key file, ending with a .prv extension. Note The WSG uses the private key from the crypto rsa-keygen command if you do not set a private key.
Step 3	WSG(config)# crypto pki wsg-cert-trap expiry notification <i>hours</i>	Configures the WSG to generate a syslog and SNMP trap a specified number of hours before the certificate expires. The range is from 1 to 720 hours. The default is 24 hours.
Step 4	WSG(config)# snmp-server enable traps ipsec trap	Configures the WSG to generate IPsec traps. If no trap option is specified with this command, all IPsec traps will be generated. If only certificate traps are needed, specify the trap options cert-expiry and cert-renewal . Note If only tunnel-rate create/delete traps are needed, enable the trap options tunnel-rate separately.

To configure CMPv2 automatic renewal, use the following configuration commands. The filenames for the private key and the certificate files will be the same filenames generated using the Privileged EXEC CMP initialize, enroll, or update commands. There is no implied command sequence in this list. Up to 20 certificate/private key pairs may be configured for automatic renewal on the WSG.

	Command	Purpose
Step 1	WSG# config	Enters global configuration mode.
Step 2	WSG(config)# crypto cert renewal retrieve current-wsg-cert <i>cert-filename.crt</i> current-wsg-private-key <i>private-key-filename.prv</i> time <i>days</i>	Configures the WSG to try to retrieve the certificate and private key files from the Cisco 7600 SUP a specified number of days prior to the certificate expiration. If the retrieved certificate has not been renewed, the WSG will continue trying to retrieve the renewed certificate until a renewed certificate is retrieved or the current certificate expires. After a renewed certificate is retrieved, it is copied to the Cisco 7600 standby SUP and standby WSG. <i>cert-filename.crt</i> —Certificate filename. <i>private-key-filename.prv</i> —Private key filename. <i>days</i> —Number of days before certificate expiration to start trying to retrieve a renewed certificate.
Step 3	WSG(config)# crypto cmp auto-update current-wsg-cert <i>cert-filename.crt</i> current-wsg-private-key <i>private-key-filename.prv</i> ca-root <i>root-filename.crt</i> ca-url <i>url</i> time <i>days</i> [key-reuse]	Configures the WSG to try to renew the certificate and optionally the private key file with the CA a specified number of days prior to the certificate expiration. Refer to the command reference chapter for a description of the retry mechanism. After the certificate is renewed, it is copied to the Cisco 7600 SUPs and standby WSG. The automatic renewal does not change the certificate or private key filename, so no additional configuration is required after automatic renewal. <i>cert-filename.crt</i> —Certificate filename. <i>private-key-filename.prv</i> —Private key filename. <i>days</i> —Number of days before certificate expiration to start trying to retrieve a renewed certificate. key-reuse —Optional parameter used to specify that the current private key should be reused. The default is to generate a new private key.

Timeline for CMPv2 certificate generation and manual certificate update:

1. CMP Initialize
2. CMP Enroll
3. Copy files to other Cisco 7600s
4. Use **crypto pki wsg-cert** command on all Cisco 7600s using the certificate
5. CMP Update
6. Copy files to other Cisco 7600s
7. Use **crypto pki wsg-cert** command on all Cisco 7600s using the certificate
8. Repeat before every expiration

Timeline for CMPv2 certificate generation and automatic certificate update:

1. CMP Initialize
2. CMP Enroll
3. Copy files to other Cisco 7600s
4. Use **crypto pki wsg-cert** command on all Cisco 7600s using the certificate
5. Use **crypto cmp auto-update** on one WSG and **crypto cert renewal retrieve** on all other WSGs using the certificate
6. No further action, unless the CA does not renew the certificate

Online Certificate Status Protocol

Currently, the WSG uses a CRL (Certificate Revocation List obtained from the CRL server), a file containing the list of certificates that have been revoked. If a peer negotiates a tunnel with a revoked certificate listed in the CRL file, the WSG will reject that negotiation. The CRL file is maintained one per trust anchor. The first negotiated tunnel for that trust anchor will retrieve the CRL file and cache it on the WSG. Subsequent negotiations will reuse the CRL file. The CRL file has an expiry data that denotes whether it is valid.

Some characteristics of the CRL mechanism make it undesirable in certain solutions. The longer the list, the longer it takes to download the CRL file. Additionally, a certificate can be revoked at any time, and the WSG will use the cached entry until the next time it retrieves the file.

The Online Certificate Status Protocol (OCSP) feature addresses some of the limitations of CRL. OCSP works to achieve the same objective as the CRL mechanism; to determine if a certificate offered by a peer has been revoked. However, OCSP differs from CRL in that the revocation status is obtained on a per-certificate basis rather than a trust anchor-basis. Since the revocation status is obtained when the certificate is first seen by the WSG, the status is up-to-date.

There is no explicit configuration. The only requirement is that CRL should be enabled for the corresponding trust anchor (we have a common knob for CRL and OCSP).

OCSP is not the best mechanism to check for certificate status in all solutions. There will be an impact to the tunnel setup rate, as each setup requires that the certificate status be verified.

DHCP Address Allocation

Previous WSG releases supported allocating IPv4 addresses from a DHCP server, or from a local IP address pool. The WSG communicates with the DHCP server as a DHCP relay agent, as configured by RFC 3046.

From Release 4.3 onwards, WSG also supports allocating IPv6 addresses from a DHCPv6 server. The WSG communicates with the DHCPv6 server as a DHCPv6 relay agent, as configured by RFC 3315. This section describes the DHCP interface from the WSG to the DHCP server, and how address pools are allocated.

Also, the WSG supports requesting DNS server IPs from the DHCP server or it can even be configured locally. As the WSG supports both DHCP configured and locally configured DNS IPs, preference will be given to IPs received from the DHCP Server. As an example, if we receive 3 DNS IPs from DHCP server and we have one DNS IP locally configured, then WSG will send only the 3 DNS IP which are

received from DHCP server, since the WSG at most supports 3 DNS IPs. In case if we receive only 2 DNS IPs from DHCP server and having one locally configured DNS IP, then all locally configured and DHCP received DNS IPs will be forwarded to remote IPsec client.

The WSG sends messages to the DHCP server under various conditions. It, the WSG, uses “RAPID COMMIT” option to assign DHCPv6 addresses. All DHCP messages are unicast to the DHCP servers by the WSG. The WSG does not send or receive broadcast messages.

Each PPC is configured with its own unique giaddr in case of IPv4 and link address in case of IPv6. Each address pool is associated with a giaddr/link address. The giaddr/link address is sent in a DHCP message and indicates to the DHCP server which address pool an IP address should be allocated from. The address pool configured on the DHCP server for each PPC can be from one or more subnets. Each subnet can be of any size. Multiple PPC of SAMI WSG can share the same address pool on the DHCP server. Based on the available pool addresses, the server will allocate addresses for the client. Server may or may not allocate the same address for different requests from the same client, depending on server configuration.

All DHCP interactions on the WSG can be debugged using the appropriate debug command. Syslogs are generated for DHCP operation status at the appropriate log level. The IP address assigned to a remote access client can be discovered using CLI commands that show details of an IPsec SA. The WSG maintains statistics for a number of DHCP messages (per type) that are sent and received. You can display these statistics using the appropriate CLI command.

During tunnel set-up time, the WSG requests a lease time for the address assigned by the DHCP server. If the lease time returned by the DHCP server is less than 2 times the IKE lifetime, the tunnel setup fails and the address is released. The DHCP statistics, debugs, and syslog events are recorded for this event. Under normal circumstances, the WSG renews the lease during each phase-1 rekey, and the lease is never allowed to expire. However, if the lease renewal fails, the assigned address is released and the tunnel is deleted. The DHCP statistics, debugs, and syslog record this event, as well.

The DHCP address allocation feature is compatible with the Single-OAM feature on the WSG. However, the DHCP server must be configured to respond to the IP address in the discover/request rather than the giaddr.

**Note**

DHCP is supported for RAS profiles and not for site-to-site profiles.

Client Identification

When the WSG contacts the DHCP server to obtain an IP address, it sends a unique ID in the DHCP messages. This client id (CID) is sent in the Client Identifier option (61) of the DHCP messages. The CID is either the entire IKE ID, or the CN field of a DN formatted IKE ID. The **type** field in the Client Identifier option is by default set to 0 unless you explicitly configure it for a different value.

Tunnel Rekey

The DHCP lease is renewed only on a phase-1 rekey. No DHCP action is taken on a phase-2 rekey (whether initiated by WSG or client).

IKE Timeout

When the WSG times out while waiting for a response to an IKE request from the HNB, the DHCP lease is released.

Load Balancing

The presence of a load balancer between an HNB and WSG(s) does not affect how the DHCP feature works in normal situations. However, if an HNB disconnects without deleting the tunnel, and then reconnects later, the load balancer might send the HNB to a different PPC. If this occurs, the DHCP server still has an active lease from the address pool for the original PPC. When the DHCP server sees the same HNB (same client identifier) from a different PPC, it needs to release the original lease and allocate a new IP address from the new PPC address pool.

High Availability

The DHCP Address Allocation feature is designed to work with a WSG is operating in a 1-1 redundant mode. The DHCP user configuration is synced from the active to standby WSG. The complete DHCP internal state is exported from the active WSG to the standby WSG. After failover the DHCP operations continue as specified for the active tunnels. But tunnels that were being setup during failover will not survive during the failover. Once the tunnel is setup, the DHCP operations must complete on either the old or new active card on failover.

Configuring DHCP Address Allocation

To configure the WSG to perform DHCP Address Allocation, perform the following tasks:

	Command	Purpose
Step 1	WSG# config	Enters global configuration mode.
Step 2	WSG (config)# crypto dhcp-server ip <i>A.B.C.D X:X:X::X port port number</i>	Configures the DHCP server IP and port number. Maximum of two DHCP servers of type either IPv4 or IPv6 can be configured by repeating the command. The no form of the command removes the DHCP server from the configuration. At least one DHCP server must be specified if DHCP address allocation is required.
Step 3	WSG(config)# crypto dhcp-client giaddr ip <i>address server-port port number</i> <i>client-port port number</i>	Specifies the giaddr, and the server and client ports used on the WSG. The server and client port number can be the same or different values. The WSG sends DHCP messages with the client port number, and receives responses from the server on the server port number. The giaddr must be unique for each PPC talking to the DHCP server. This command is required if you require DHCP address allocation.
Step 4	WSG (config)# crypto dhcp-client client-id-type extract-CN	Specifies the client id that is sent by the WSG (in option 61 of DHCP message). By default the HNB's IKE ID is used as the client ID. If the HNB IKE ID is in the DN format, and the CN part of the DN is to be sent as the client ID, then this command must be configured. The no form of the command reverts the client ID to the default setting.

	Command	Purpose
Step 5	WSG(config)# crypto dhcp-client link-addr <i>ipv6 address server-port port number</i> client-port <i>port number</i>	Specifies the global unicast IPv6 Link-Address in Relay Forward message used by the WSG. This address is more like an identifier to which address pool to be used in the server side. This command is mandatory if DHCPv6 address allocation is required.
Step 6	WSG(config)# crypto profile <i>profile-name</i> ipsec ip address-pool { dhcp <i>local-pool-name</i> }	Use this command when a profile is required to use DHCP-based address allocation. When the profile is activated, the mandatory global DHCP configuration is checked for completeness. If any profile is activated with DHCP address allocation, the global DHCP configuration commands cannot be modified or removed.

To monitor and troubleshoot the DHCP Address Allocation feature, perform the following tasks:

	Command	Purpose
Step 1	WSG# show crypto dhcp	Displays DHCP address statistics.
Step 2	WSG# debug crypto dhcp { errors events verbose }	Enables debugging for DHCP crypto parameters .

Here is a running configuration example:

```
hostname WSG
ha interface vlan 2073
  ip address 77.77.73.133 255.255.255.0
interface vlan 63
  ip address 88.88.63.133 255.255.255.0
  alias 88.88.63.3 255.255.255.0
interface vlan 223
  ip address 222.222.223.133 255.255.255.0
  alias 222.222.223.3 255.255.255.0
ip route 0.0.0.0 0.0.0.0 88.88.63.100
oam mode single 223
  oam-ip route 44.44.44.0 255.255.255.0 222.222.223.100
logging ip 44.44.44.17
snmp-server community public ro
snmp-server community private rw
snmp-server host 44.44.44.16 traps version 2c public
ip name-server 44.44.44.201
snmp-server enable traps ipsec
crypto syslog-level 1
!
crypto pki wsg-cert sami-cert.crt wsg-private-key sami-key.prv
crypto pki trustpoint rootCA cacert.crt crl disable
!
crypto dhcp-client giaddr 88.88.63.3 server-port 2133 client-port 2133
!
crypto profile "prof-1"
  isakmp
    lifetime 7200
    self-identity id-type fqdn id SAMI.cisco.com
  ipsec
    security-association lifetime 86400
    access-permit ip 172.60.0.0 subnet 16
    ip address-pool dhcp
```

```

    activate
    !

```

Here is a running configuration example of WSG with DHCPv6 server:

```

hostname s9p3
ha interface vlan 17
    ip address 192.168.17.9 255.255.255.0
interface vlan 2
    ip address 10.77.161.34 255.255.255.192
interface vlan 77
    ip address 62.21.99.95 255.255.255.0
interface vlan 8
    ip address 192.168.9.12 255.255.255.0
    alias 192.168.9.93 255.255.255.0
    ipv6 address 2003::55:55:55:110/112
interface vlan 25
    ip address 192.168.25.9 255.255.255.0
    ipv6 address 2006::77:77:77:93/112
    ipv6 alias 2006::77:77:77:103/112

router bgp 9
    neighbor 2006::77:77:77:1 remote-as 9 next-hop-alias 2006::77:77:77:93

ip route 192.168.10.12 255.255.255.255 192.168.5.1
ip route 0.0.0.0 0.0.0.0 10.77.161.1
ip route 192.168.5.0 255.255.255.0 192.168.9.1
ip route 192.168.25.32 255.255.255.255 192.168.9.1
ip route 192.178.0.0 255.255.0.0 192.168.9.1
ip route 192.179.0.0 255.255.0.0 192.168.9.1
ip route 192.177.0.0 255.255.0.0 192.168.9.1
ip route 192.168.6.33 255.255.255.255 192.168.9.1
ip route 20.20.20.1 255.255.255.255 192.168.9.1

crypto remote-secret fqdn test.cisco.com secret "test"
!
crypto pki wsg-cert sami-cert.crt wsg-private-key sami-key.prv
crypto pki trustpoint rootCA ca-cert.crt crl disable
!
crypto rri enable

crypto dhcp-server ip 2006::77:77:77:32 port 547
!
crypto dhcp-client link-addr 2006::77:77:77:93 server-port 547 client-port 546
!

crypto profile "ipv6-psk-ras"
    isakmp
        self-identity id-type email id sami@cisco.com
        local-secret "test"
        authentication pre-shared
    ipsec
        access-permit ip 2002::5:5:5:1 subnet 112
        ip address-pool dhcp
    activate
!

```

IPv6

IPv6 support in WSG Release 3.0 includes support for both IPv6 IKE and IPv6 ESP packets and related IPv6 addressing where required. WSG Release 3.0 and above supports all four of the following combinations of IPv4/IPv6 encapsulation in the tunnels:

- IPv6 Over IPv6
- IPv6 Over IPv4
- IPv4 Over IPv6
- IPv4 Over IPv4

Configuring IPv6

The PPC requires an IPv6 VLAN interface to be configured to act as an IPv6 IKE server. The other changes required are to the traffic selectors inside the profile. The traffic selector accepts IPv6 addresses.

All the profile-based configuration remains the same. Changes have been made to the CLI at the IP address option level. Every option to enter an IP address now accepts either an IPv4 or IPv6 address.

IPv6 IP addresses for self identity are supported. Access Permit, Address pools, Local IP, Peer IP and all other CLIs that accept an IP address as a parameter have been enhanced for IPv6.

The traffic selector determines the protocol of the secured traffic inside the tunnel. The local-ip and peer-ip determine the protocol used for IKE exchange and the IP addresses in the outer header of the ESP packets sent out of WSG datapath to the peer.

The following is a sample configuration of an IPv6-over-IPv6 tunnel:

```
interface vlan 39
  ipv6 address 2001:88:88:94::4/96
  ipv6 route ::/0 2001:88:88:94::1

crypto profile "s2s-IPv6-over-IPv6"
  profile-type site-to-site
  isakmp
  peer-ip 3001:99:99:94::4
  self-identity id-type email id ppcl@cisco.com
  local-secret "cisco123"
  authentication pre-shared
  ipsec
  access-permit "one"
    protocol any src-ip 4001:100:100:94::4 96 src-port 0 65535
    dst-ip 5001:200:200:94::4 96
    dst-port 0 65535
  local-ip 2001:88:88:94::4
  activate
```

The following is a sample configuration of an IPv6-over-IPv4 tunnel:

```
interface vlan 39
  ip address 39.39.39.30 255.255.255.0
  ipv6 address 2001:88:88:94::4/96
  ip route 0.0.0.0 0.0.0.0 39.39.39.3
  ipv6 route ::/0 2001:88:88:94::1
  crypto profile "s2s-IPv6-over-IPv4"
  profile-type site-to-site
```

```

isakmp
 peer-ip 59.59.59.50
 self-identity id-type email id ppc1@cisco.com
 local-secret "cisco123"
 authentication pre-shared
 ipsec
  access-permit "one"
   protocol any src-ip 4001:100:100:94::4 96 src-port 0 65535
   dst-ip 5001:200:200:94::4 96
   dst-port 0 65535
 local-ip 39.39.39.30
 activate

```

The following is a sample configuration of an IPv4-over-IPv6 tunnel:

```

interface vlan 39
 ip address 39.39.39.30 255.255.255.0
 ipv6 address 2001:88:88:94::4/96
 ip route 0.0.0.0 0.0.0.0 39.39.39.3
 ipv6 route ::/0 2001:88:88:94::1
 crypto profile "s2s-IPv4-over-IPv6"
 profile-type site-to-site
 isakmp
  peer-ip 3001:99:99:94::4
  self-identity id-type email id ppc1@cisco.com
  local-secret "cisco123"
  authentication pre-shared
 ipsec
  access-permit "one"
   protocol any src-ip 60.0.0.0 8 src-port 1 65535 dst-ip 40.0.0.0 8 dst-port 1 65535
 local-ip 2001:88:88:94::4
 activate

```

The following is a sample configuration of an IPv4-over-IPv4 tunnel:

```

interface vlan 39
 ip address 39.39.39.30 255.255.255.0
 ip route 0.0.0.0 0.0.0.0 39.39.39.3
 crypto profile "s2s-IPv4-over-IPv4"
 profile-type site-to-site
 isakmp
  peer-ip 59.59.59.50
  self-identity id-type email id ppc1@cisco.com
  local-secret "cisco123"
  authentication pre-shared
 ipsec
  access-permit "one"
   protocol any src-ip 60.0.0.0 8 src-port 1 65535 dst-ip 40.0.0.0 8 dst-port 1 65535
 local-ip 39.39.39.30
 activate

```

The subnet combination command for the site-to-site tunnels is modified to accept a full range of subnet sizes for IPv6, subnet sizes more than 32 are not used to handle IPv4 traffic in the datapath, whereas all configured subnet sizes will be used for handling IPv6 traffic.

Example:

```

crypto site-to-site-lookup priority 1 source-netmask 128 destination-netmask 128
crypto site-to-site-lookup priority 2 source-netmask 96 destination-netmask 96
crypto site-to-site-lookup priority 3 source-netmask 32 destination-netmask 32

```

Tunnel counters and show commands:

```
show crypto ipsec sa
show crypto ipsec sa remote-ip
show crypto isakmp sa
show crypto isakmp sa remote-ip
```

Blacklisting

You might want to block certain access points (APs) from connecting to the WSG. In case certificates are used to authenticate an AP, the CRL mechanism is used to revoke the certificate of the AP that needs to be blocked, which prevents it from setting up a tunnel with the WSG. However, when an AP is only required to be blocked temporarily (for instance because of an outstanding balance on the billing account), blacklisting is an easier and faster mechanism to block an AP.

The WSG blacklisting feature prevents an AP from setting up a tunnel to the WSG. When an AP attempts to setup a tunnel with WSG, the IKE ID of the AP is searched in a blacklist file available to the WSG. If a match is found, the AP is prevented from establishing a tunnel and the AUTH request fails.

You must edit the blacklist file outside of the Cisco 7600 chassis, and copy it to the SUP disk. Initially, you should configure the WSG with the filename of the blacklist file. During this configuration, the blacklist file is internally rcp-ed from the SUP disk to the WSG ram disk, and the IKE stack is informed of the location of the file. The IKE stack performs blacklisting based on the entries in the file. If you need to update the blacklist entries, follow this procedure:

- Edit the blacklist file outside the Cisco 7600 chassis.
- Copy the blacklist to the SUP disk with the same file name that you initially used.
- Execute the **crypto blacklist file resync** command on the WSG. The WSG copies the updated file from the SUP disk to its ramdisk, and informs the IKE stack about the updated file. The IKE stack now uses the new blacklist file.

You must execute the blacklist file configuration and resync operation on all PPCs of the WSG card where blacklisting is required.

The blacklist file is a text file with multiple lines. Each line is one blacklisted IKE ID. It is possible for the blacklist file to be empty (no blacklisted entries). Here is an example of a blacklist file:

```
fqdn "LS1-0.cisco.com"
fqdn "LS1-1.cisco.com"
fqdn "LS1-2.cisco.com"
fqdn "LS1-3.cisco.com"
ip "192.168.10.10"
ip "192.168.10.50"
email "user@sample.com"
dn "C=US,ST=CA,L=San Jose,O=Cisco,OU=SMBU,CN=organization.bu.org"
```

Configuring Blacklisting on the WSG

To configure and monitor the WSG's blacklisting feature, perform the following tasks:

	Command	Purpose
Step 1	wsg# config	Enters global configuration mode.
Step 2	wsg(config)# crypto blacklist file filename	Configures the blacklist filename on the WSG. The blacklist file must be present on the SUP disk before this configuration is done. If the file is not present on the SUP, the configuration fails. The default value is that the feature is off.
Step 3	wsg(config)# crypto blacklist file resync	Recopies the blacklist file from the SUP disk and inform the IKE stack about the update.
Step 4	wsg(config)# clear crypto isakmp sa remote-id remote ID	Deletes all IKE and IPsec SAs associated with a remote ID.
Step 5	wsg# show crypto blacklist file	Lists all of the currently blacklisted IDs.
Step 6	wsg# show crypto blacklist stats	Displays the following information: <ul style="list-style-type: none"> • Number of IDs in a blacklist • Number of tunnel setup attempts blocked due to blacklisting

Here is an example of the **show crypto blacklist stats** command:

```
wsg# show crypto blacklist stats
```

```
Blacklist Statistics
Number of blacklisted entries : 500
IKEv2 [R] initial exchanges      : Allowed = 53, Blocked = 101
IKEv2 [R] create child exchanges  : Allowed = 0, Blocked = 0
IKEv2 [R] IPsec SA rekeys         : Allowed = 98, Blocked = 0
IKEv2 [R] IKE SA rekeys           : Allowed = 49, Blocked = 0
IKEv2 [I] IPsec SA rekeys         : Allowed = 0, Blocked = 0
IKEv2 [I] IKE SA rekeys           : Allowed = 0, Blocked = 0
IKEv1 [R] main mode exchanges     : Allowed = 0, Blocked = 0
IKEv1 [R] aggressive mode exchanges : Allowed = 0, Blocked = 0
IKEv1 [R] quick mode exchanges    : Allowed = 0, Blocked = 0
IKEv1 [I] IPsec SA rekeys         : Allowed = 0, Blocked = 0
IKEv1 [I] DPD SA creations        : Allowed = 0, Blocked = 0
```



Note

The WSG blacklisting feature is independent of other blacklisting or security functionality that may exist as part of other Cisco products and solutions.

RADIUS Accounting

In some femto networks, an AP sets up an IPsec tunnel with the WSG, and then sends an registration message through the tunnel to a Femto Gateway (FGW). The register message is an IP packet that also contains the ID of the AP registering with the FGW. The ID used by the AP is the same as the IKE ID used by the AP during IPsec tunnel setup. The FGW must ensure that an authenticated AP is not presenting itself as another AP during registration. The FGW compares the source IP address of the

registration packet with the internal IP address assigned by the WSG for the same AP (the ID is the lookup key). Similarly, the WSG needs to send the ID to an internal IP address mapping to the FGW each time it assigns an IP to the AP.

RADIUS Accounting messages are used to send the IKE ID to the assigned IP address mapping from the WSG to the AAA server running in the FGW.

After a tunnel is successfully established, a RADIUS accounting message is sent to the AAA server to record the mapping from the AP IKE ID to the WSG assigned internal AP IP address. The RADIUS timeout and retry mechanism is used to handle cases where a RADIUS server might be down. However, failure to update the RADIUS server will not fail the tunnel setup.

Table 2-3 shows the accounting request message attributes for tunnel setup.

Table 2-3 Accounting Start Request

Attribute Type	Comment
From RFC2865...	—
User-Name	Set to IKEv2 IDi.
Framed-IP-Address (IPv6: Framed-IPv6-Prefix, from RFC3162)	Set to allocated inner IP address.
NAS-IP-Address	Set to the WSG IP address that the IKE messages were received on to setup the tunnel.
From RFC2866...	—
Acct-Status-Type	Start
Acct-Session-Id	Set to an unique value so that the start and stop records can be matched.

When the tunnel is deleted, the record is deleted from the AAA server.

Table 2-4 displays the accounting request message attributes for tunnel deletion.

Table 2-4 Accounting Stop Request

Attribute Type	Comment
From RFC2865...	—
User-Name	Set to IKEv2 IDi.
NAS-IP-Address	Set to the WSG IP address that the IKE messages were received on to setup the tunnel.
From RFC2866...	—
Acct-Status-Type	Stop
Acct-Session-Id	Set to an unique value so that the start and stop records can be matched.



Note

When upgrading to WSG Release 3.0 from a previous 2.X release, if a RADIUS server configuration exists, the crypto profile(s) will be inactive after the upgrade. To reactivate, configure the **crypto radius nas-id** or **crypto radius nas-ip** commands and then activate the profile(s).

Features of RADIUS Accounting

The RADIUS Accounting feature has the following requirements and limitations:

- The RADIUS Accounting feature supports both IKEv1 and IKEv2.
- The RADIUS Accounting feature supports both IPv4 and IPv6.
- You can enable and disable this feature at the global level, but all crypto profiles have to be in a deactivated state before enabling/disabling the feature.
- You can specify the RADIUS accounting server IP and port. If the RADIUS accounting port is not specified, then the default value of 1813 is used. There is an existing CLI (API) to use to configure for the RADIUS server. The command is modified so that specifying the auth-port/acct-port optional.
- There is an option to specify the source IP of the RADIUS packets. If the source IP is not specified, then the source IP is picked by the Linux kernel based on routing rules. There is an existing CLI to specify the source IP.
- The RADIUS Accounting update is sent whenever the WSG assigns an internal IP address during tunnel setup to the remote peer from the local address pool, or from external sources like DHCP.
- Syslogs and debugs are generated at the appropriate level for all blacklisting RADIUS accounting feature operations.
- RADIUS protocol statistics such as the different types of messages sent, timeouts, retries, etc., are maintained.

Configuring RADIUS Accounting on the WSG

To configure and monitor the RADIUS accounting feature on the WSG, perform the following tasks:

	Command	Purpose
Step 1	wsg# config	Enters global configuration mode.
Step 2	wsg(config)# crypto radius accounting {enable}	Enables the RADIUS Accounting feature. The default value is that the feature is disabled.
Step 3	wsg# show crypto radius statistics	Displays the count of different RADIUS messages sent and received. Also RADIUS timeout and retry counters are displayed.



Note

Existing CLI are used to configure the RADIUS server IP addresses, RADIUS server ports and source IP address for RADIUS messages. When multiple RADIUS servers are configured, the accounting messages are sent to the first server in the list that the WSG is successfully able to communicate with.

Here is sample output for the **show crypto radius statistics** command:

```
wsg# show crypto radius statistics
Radius Accounting Statistics
  Accounting requests sent           : 1
  Accounting-On requests sent        : 0
  Accounting-Off requests sent       : 0
  Accounting-Start requests sent     : 1
  Accounting-Stop requests sent      : 0
  Accounting Responses on received   : 0
  Accounting Invalid responses received : 0
  Accounting requests failed         : 0
```



```
Accounting requests, Invalid IKE ID : 0
Accounting requests timedout       : 1
Accounting requests retransmission : 4
Accounting requests cancelled      : 0
```

EAP Peer Authentication

WSG supports authentication of a peer through EAP-MD5, EAP-AKA and EAP-SIM protocols. These protocols are only supported with certificates for authenticating the WSG to the peer. These protocols require the IPSec stack to talk to an external RADIUS server to authenticate a peer device. Use of preshared keys to authenticate the WSG to the peer is not allowed by the standards, but might be required to support some legacy equipment. The EAP authentication is supported for IKEv2 only.

Sample configuration output:

```
crypto radius source-ip 88.88.93.3
!
crypto radius-server host 44.44.44.200 key "cisco" auth-port 1812 acct-port 1813

crypto profile "ras_eap-aka"
  isakmp
    self-identity id-type email id sami@cisco.com
    eap-type aka
  ipsec
    access-permit ip 172.0.0.0 subnet 8
    ip address-pool "myPool"
  activate

crypto profile "ras_eap-md5"
  isakmp
    self-identity id-type email id sami@cisco.com
    eap-type md5
  ipsec
    access-permit ip 172.0.0.0 subnet 8
    ip address-pool "myPool"
  activate

crypto profile "ras_eap-sim"
  isakmp
    self-identity id-type email id sami@cisco.com
    eap-type sim
  ipsec
    access-permit ip 172.0.0.0 subnet 8
    ip address-pool "myPool"
  activate
```

Reverse Route Injection (RRI)

The RRI feature is introduced in WSG Release 3.0 and obviates the need to manually configure static routes on the SUP for clear traffic routing purposes in the reverse direction. RRI route entries are injected into the SUP when IPSec tunnels are created. These route entries are correspondingly withdrawn from the SUP when the IPSec tunnels are deleted. The BGP protocol is used to re-distribute the routes from WSG to the SUP. For WSG Release 3.0, the RRI feature supports only IPv4. IPv6 is supported starting in WSG Release 4.0. Also, only site-to-site profiles are supported. The VRF feature on the WSG cannot be enabled when the RRI feature is already configured.

The BGP peer on the SUP needs to establish BGP sessions to two separate BGP neighbors on the WSG; one with the active card and one with the standby card for redundancy. The BGP AS-number is configured for an iBGP setup. Depending on the network topology, an eBGP setup is also supported between the WSG and SUP.

A sample configuration on the SUP (for IPv4):

```
router bgp 7675
  bgp router-id 10.10.14.1
  bgp log-neighbor-changes
  neighbor 33.33.33.33 remote-as 7675
  neighbor 33.33.33.34 remote-as 7675
!
address-family ipv4
  neighbor 33.33.33.33 activate
  neighbor 33.33.33.34 activate
  no auto-summary
exit-address-family
```

A sample configuration on the SUP (for IPv6):

```
ipv6 unicast-routing
mls cef error action reset
mls cef maximum-routes ipv6 32 (**)

router bgp 7675
  bgp router-id 10.12.12.1
  bgp log-neighbor-changes
  bgp graceful-restart restart-time 120
  bgp graceful-restart stalepath-time 360
  bgp graceful-restart
  no bgp default ipv4-unicast
  neighbor 2001:88:88:94::46 remote-as 7675
  neighbor 2001:88:88:94::46 ebgp-multihop 255 (*)
  neighbor 2001:88:88:94::47 remote-as 7675
  neighbor 2001:88:88:94::47 ebgp-multihop 255 (*)

address-family ipv6
  neighbor 2001:88:88:94::46 activate
  neighbor 2001:88:88:94::47 activate
exit-address-family
```

The **crypto rri enable** command is required to enable the RRI feature. The profile configuration is like before. BGP configuration on the WSG requires neighbor and next-hop-alias information so that there is a common next-hop point for the SUP to route to either the active or standby card.

A sample configuration on the WSG (for IPv4):

```
router bgp 7675
  neighbor 33.33.33.3 remote-as 7675 next-hop-alias 33.33.33.30

crypto rri enable

crypto profile "rri-site"
  profile-type site-to-site
  isakmp
    lifetime 7200
    self-identity id-type email id ppcl@cisco.com
    local-secret "cisco123"
    authentication pre-shared
  ipsec
    security-association lifetime seconds 3600
    access-permit "primary"
    protocol any src-ip 60.0.0.0 8 src-port 1 65535 dst-ip 40.0.0.0 8 dst-port 1 65535
```

```

    local-ip 33.33.33.30
  activate

```

A sample configuration on the WSG (for IPv6):

```

router bgp 7675
neighbor 2001:88:88:94::44 remote-as 7675 next-hop-alias 2001:88:88:94::43

crypto rri enable

crypto profile "rri-site-ipv6"
profile-type site-to-site
isakmp
lifetime 7200
self-identity id-type email id ppcl@cisco.com
local-secret "cisco123"
authentication pre-shared
ipsec
security-association lifetime seconds 3600
access-permit "primary"
protocol any src-ip 2001:77:77:77::1 64 src-port 0 65535 dst-ip 2001:98:98:98::1 64
dst-port 0 65535
local-ip 2001:88:88:94::43
activate

```



Note

The above IPv6 configurations require the following:

- Requires SUP software version 15.1(2)S2 or later
- Requires additional configuration on the SUP above (marked with **) to support 17K IPv6 RRI entries **mls cef maximum-routes ipv6 [maximum number RRI entries]**
- Requires additional configuration on the SUP above (marked with *) to support IPv6 in eBGP mode **neighbor [ipv6_bgp_neighbor] ebgp-multihop 255**
- When the RRI feature is used, expect an approximate 3 second delay from the time a tunnel is added to the time when the injected RRI route actually shows up on the SUP. Reverse/clear traffic will only start passing approximately 3 seconds after a tunnel is created.

VRF Configuration

Virtual Routing and Forwarding (VRF) allows the creation of multiple virtual networks within a single network entity. Each VRF comprises an IP routing table and a forwarding table, allowing the use of the same or overlapping IP addresses without conflicts.

In a single network entity, multiple VRFs can be used to create isolation between virtual networks. VRFs allow encrypted/decrypted traffic separation, by having the encrypted traffic in one VRF and the decrypted traffic in another VRF.

- Inside VRF (indoor) contains decrypted traffic
- Outside VRF (front door) contains encrypted traffic

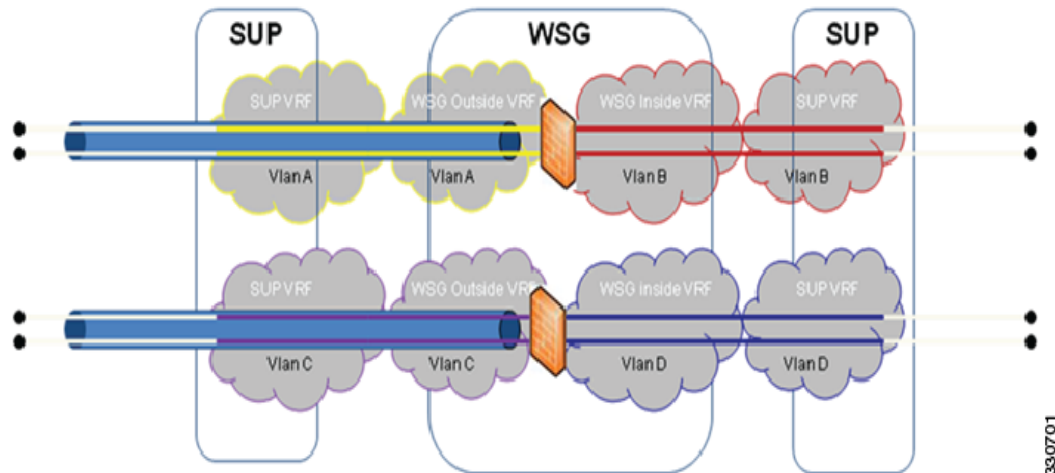
The typical case for this is an ISP that provides VPN service to multiple enterprise customers on the same box, the users and branches connect using internet for the encrypted traffic, but the decrypted traffic needs to go to the private network of each separate customer and this traffic cannot be mixed.

The sample configuration below is for two profiles. Each profile has its own inside and outside VRF configured. This ensures the encrypted and decrypted traffic for both profiles are separated onto different VRFs. The same IP address can be used on both profiles, while also remaining in separate routing tables. This configuration is flexible. The outside VRFs could be configured on the same VRF or on the global VRF if there is no potential to overlap IP addresses.

**Note**

Traffic is considered to be on the same VRF from the SUP to the WSG if the VLANs are the same.

Figure 2-1 WSG VRF Configuration.



WSG VRF Configuration on a PPC

VRF Definitions

```
ip vrf OutsideYellow
ip vrf OutsidePurple
ip vrf InsideRed
ip vrf InsideBlue
```

VRF Interface Definitions

```
interface vlan 33
  vrf OutsideYellow
  ip address 33.33.33.30 255.255.255.0
interface vlan 34
  vrf OutsidePurple
  ip address 33.33.33.30 255.255.255.0
interface vlan 77
  vrf InsideRed
  ip address 77.77.77.70 255.255.255.0
interface vlan 78
  vrf InsideBlue
  ip address 77.77.77.70 255.255.255.0
```

VRF Default Route Definitions

```
ip route 0.0.0.0 0.0.0.0 33.33.33.3 vrf OutsideYellow
ip route 0.0.0.0 0.0.0.0 33.33.33.3 vrf OutsidePurple
```

```
ip route 0.0.0.0 0.0.0.0 77.77.77.7 vrf InsideRed
ip route 0.0.0.0 0.0.0.0 77.77.77.7 vrf InsideBlue
```

VRF Profile Configuration

```
crypto profile "site-to-site"
  profile-type site-to-site
  isakmp
    vrf-outside OutsideYellow
    peer-ip 50.0.0.1
    self-identity id-type email id ppcl@cisco.com
  ipsec
    vrf-inside InsideRed
    access-permit "s2s-1"
      protocol any src-ip 60.0.0.0 24 src-port 1 65535 dst-ip 40.0.0.0 24 dst-port 1
        65535
    access-permit "s2s-2"
      protocol any src-ip 60.0.0.0 24 src-port 1 65535 dst-ip 40.0.0.0 24 dst-port 1 65535
    local-ip 33.33.33.30

crypto profile "site-to-site-2"
  profile-type site-to-site
  isakmp
    vrf-outside OutsidePurple
    peer-ip 50.0.0.1
    self-identity id-type email id ppcl@cisco.com
  ipsec
    vrf-inside InsideBlue
    access-permit "s2s-1"
      protocol any src-ip 60.0.0.0 24 src-port 1 65535 dst-ip 40.0.0.0 24 dst-port 1
        65535
    access-permit "s2s-2"
      protocol any src-ip 60.0.0.0 24 src-port 1 65535 dst-ip 40.0.0.0 24 dst-port 1 65535
    local-ip 33.33.33.30
```

WSG IPv4 VRF Configuration on 7600 SUP

VRF Definitions

```
ip vrf Yellow
  rd 2:2
ip vrf Purple
  rd 3:3
ip vrf Red
  rd 4:4
ip vrf Blue
  rd 5:5
```

Physical Interface Configuration

```
interface GigabitEthernet1/1
  switchport
  switchport access vlan 30
  switchport mode access

interface GigabitEthernet1/2
  switchport
  switchport access vlan 77
  switchport mode access

interface GigabitEthernet1/3
  switchport
  switchport access vlan 31
```

```

switchport mode access

interface GigabitEthernet1/4
switchport
switchport access vlan 78
switchport mode access

```

VLAN Configuration

```

interface Vlan30
ip vrf forwarding Yellow
ip address 22.22.22.3 255.255.255.0

interface Vlan31
ip vrf forwarding Purple
ip address 22.22.22.3 255.255.255.0

interface Vlan33
ip vrf forwarding Yellow
ip address 33.33.33.3 255.255.255.0

interface Vlan34
ip vrf forwarding Purple
ip address 33.33.33.3 255.255.255.0

interface Vlan77
ip vrf forwarding Red
ip address 77.77.77.3 255.255.255.0

interface Vlan78
ip vrf forwarding Blue
ip address 77.77.77.3 255.255.255.0

```

Static Route Definitions

```

ip route vrf Yellow 50.0.0.0 255.0.0.0 22.22.22.4
ip route vrf Yellow 60.0.0.0 255.0.0.0 77.77.77.4
ip route vrf Red 44.44.0.0 255.255.0.0 77.77.77.33

ip route vrf Purple 50.0.0.0 255.0.0.0 22.22.22.4
ip route vrf Purple 60.0.0.0 255.0.0.0 77.77.77.4
ip route vrf Blue 44.44.0.0 255.255.0.0 77.77.77.33

```

WSG IPv6 VRF Configuration on 7600 SUP

VRF Definitions

```

ip vrf OutsideYellow
ip vrf OutsidePurple
ip vrf InsideRed
ip vrf InsideBlue

```

VLAN Configuration

```

interface vlan 113
vrf OutsideYellow
ip address 88.88.113.33 255.255.255.0
alias 88.88.113.93 255.255.255.0
ipv6 address 2001:88:88:113::33/64
ipv6 alias 2001:88:88:113::93/64

interface vlan 203
vrf OutsidePurple
ip address 203.203.113.33 255.255.255.0

```

```

alias 203.203.113.93 255.255.255.0
ipv6 address 2001:203:203:113::33/64
ipv6 alias 2001:203:203:113::93/64
interface vlan 77
 vrf InsideRed
 ip address 77.77.77.33 255.255.255.0
 alias 77.77.77.93 255.255.255.0
 ipv6 address 2001:77:77:77::33/64
 ipv6 alias 2001:77:77:77::93/64
interface vlan 78
 vrf InsideBlue
 ip address 78.78.78.33 255.255.255.0
 alias 78.78.78.93 255.255.255.0
 ipv6 address 2001:78:78:78::33/64
 ipv6 alias 2001:78:78:78::93/64

```

Static Route Definitions

```

ipv6 route ::/0 2001:77:77:77::7 vrf InsideRed
ipv6 route ::/0 2001:88:88:113::100 vrf OutsideYellow
ipv6 route ::/0 2001:78:78:78::8 vrf InsideBlue
ipv6 route ::/0 2001:203:203:113::100 vrf OutsidePurple

```

Crypto Profile Configuration

```

crypto profile "s2s"
 profile-type site-to-site
 isakmp
  vrf-outside OutsideYellow
  lifetime 720000
  dpd-timeout 270
  self-identity id-type email id sami@cisco.com
 ipsec
  security-association lifetime seconds 360000
  access-permit "ap-ipv4"
   protocol any src-ip 172.110.0.0 16 src-port 0 65535 dst-ip 10.33.0.0 16 dst-port 0 65535
  access-permit "ap-ipv6"
   protocol any src-ip :: 0 src-port 0 65535 dst-ip :: 0 dst-port 0 65535
  access-permit "ap-ipv4-2"
   protocol any src-ip 172.0.0.0 24 src-port 0 65535 dst-ip 10.23.0.0 16 dst-port 0 65535
  vrf-inside InsideRed
 activate

....

crypto profile "s2s-tims-tc93-ipv4-ts"
 profile-type site-to-site
 isakmp
  vrf-outside OutsidePurple
  self-identity id-type email id sami@cisco.com
 ipsec
  access-permit "ap2"
   protocol any src-ip 2001:172:110:: 64 src-port 0 65535 dst-ip 2001:10:3:3:1:2:: 96 dst-port 0 65535
  access-permit "ap1"
   protocol any src-ip :: 0 src-port 0 65535 dst-ip :: 0 dst-port 0 65535
  access-permit "ap3"
   protocol any src-ip 2001:172:110:: 64 src-port 0 65535 dst-ip 2001:10:3:3:1:3:: 96 dst-port 0 65535
  vrf-inside InsideBlue
 activate

```

Configuring WSG Performance/Throughput Indicators

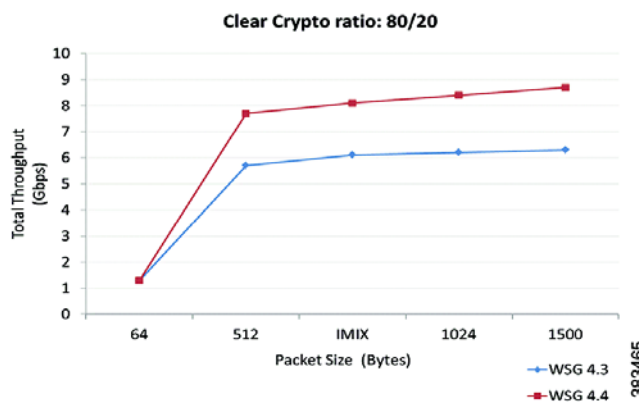
With WSG release 4.4, the IXP Traffic distribution feature is included to increase the overall throughput of the WSG SAMI. This feature provides a method to divide the Clear traffic between 2 IXPs (IXP0 and IXP1) and enables even distribution of traffic among 2 IXPs. The IXP1 now handles more of the post encryption data which was originally handled by IXP0.

Table 2-5 shows the throughput comparison between Release 4.3 and 4.4 when Clear-ESP traffic ratio is 80/20. Figure 2-2 illustrates the same through a line chart.

Table 2-5 Max Throughput comparison between release 4.3 and 4.4 for 80/20 traffic



Packet Size (Bytes)	WSG 4.3 throughput (Gbps)	WSG 4.4 throughput (Gbps)
64	1.3	1.3
512	5.7	7.7
IMIX	6.1	8.1
1024	6.2	8.4
1500	6.3	8.7


Figure 2-2 Line chart for max throughput comparison between release 4.3 and 4.4 for 80/20 traffic



To generate SNMP trap when WSG throughput utilization goes above the configured value, perform the following tasks:

	Command	Purpose
Step 1	<code>switch# config</code>	Enters global configuration mode.
Step 2	<code>switch (config)# snmp-server enable traps ipsec throughput-threshold</code>	Enables the SNMP trap when WSG throughput utilization goes above the configured or default value for a sustained number of intervals.

Command	Purpose
Step 3 switch (config)# crypto throughput threshold <i>threshold interval interval</i>	Sets the throughput utilization threshold in percentage and number of sustained intervals. By default threshold is 50% and interval is 2. Note Each interval is of 5 mins.
Step 4 wsg# show crypto throughput	Displays the throughput data for the last calculated interval on WSG.
Step 5 wsg# show crypto throughput ixp <1/2>	Displays the throughput data for packets to/from Nitrox and the average throughput utilization for the last calculated interval on WSG for each IXP. IXP0 display also shows the packet data punted to IXP1. <ul style="list-style-type: none"> • <i>ixp</i> — Selects IXP number <ul style="list-style-type: none"> - 1 — IXP0 - 2 — IXP1 <div style="text-align: right;">  Note This command is added with Release 4.4. </div>
Step 6 wsg# show crypto throughput history interval <i>interval type</i>	Displays the history of throughput in Mbp/s and Packets/s. <ul style="list-style-type: none"> • <i>interval</i> — Set refresh interval for history stats collection: <ul style="list-style-type: none"> - 1 - 5 minutes - 2 - 1 hour - 3 - 3 hours • <i>type</i> — Sets type of unit of throughput: <ul style="list-style-type: none"> Mbps Kpps (Kilo-Packets-per-second)
Step 7 wsg# show crypto throughput history interval interval type ixp <1/2>	Displays the history of throughput in Mbp/s and Packets/s separately for each IXP. <ul style="list-style-type: none"> • <i>ixp</i> — Selects IXP number <ul style="list-style-type: none"> - 1 — IXP0 - 2 — IXP1 <div style="text-align: right;">  Note This command is added with Release 4.4. </div>
Step 8 wsg# show crypto throughput distribution history	Displays the number of intervals the throughput fell in a certain bucket range. Each interval is 5 minutes.

Command	Purpose
Step 9 wsg# show crypto throughput distribution history ixp <1/2>	Displays the number of intervals the throughput fell in a certain bucket range for each IXP. Each interval is 5 minutes. <ul style="list-style-type: none"> • <i>ixp</i> — Selects IXP number <ul style="list-style-type: none"> - 1 — IXP0 - 2 — IXP1 <div style="text-align: right; margin-top: 10px;">  Note This command is added with Release 4.4. </div>
Step 10 wsg# clear crypto throughput counters	This optional command clears throughput counters.

```

WSG# config
Enter configuration commands, one per line. End with CNTL/Z.
WSG (config)# snmp-server enable traps ipsec throughput-threshold
WSG (config)# crypto throughput threshold 80 interval 5

```

Here are examples of the **show crypto throughput** commands:

```

wsg# show crypto throughput
Throughput (Mbp/s) : 4992
Throughput (Kpp/s) : 626
Average Packet Size (bytes) : 996
Throughput Utilization (%) : 58
Peak Throughput Utilization (%) : 100 Sat Sep 06 15:39:50.012 UTC
Peak Throughput (Mbp/s) : 18400
Peak Packet Size (bytes) : 509

wsg# show crypto throughput ixp 1
Throughput - First Path (Mbp/s) : 3941
Throughput - First Path (Kpp/s) : 501
Average Packet Size - First Path (bytes) : 983
Throughput - Return Path (Mbp/s) : 1051
Throughput - Return Path (Kpp/s) : 125
Average Packet Size - Return Path (bytes) : 1051
Throughput Utilization (%) : 58
Peak Throughput Utilization (%) : 100 Sat Sep 06 15:39:50.012 UTC
Peak Throughput - First Path (Mbp/s) : 9200
Peak Packet Size - First Path (bytes) : 876
Peak Throughput - Return Path (Mbp/s) : 9200
Peak Packet Size - Return Path (bytes) : 1021
Punted to IXP2 (Mbp/s) : 2956
Punted to IXP2 (Kpp/s) : 376

```

Here is an example of the **show crypto throughput history** command:

```

wsg# show crypto throughput history interval 5minutes Kpps ixp 1
3200
3000
2800
2600
2400
2200
2000
1800
1600

```

```

1400
1200 #
1000
800
600
400 #####
200
.... 1...1...2...2...3...3...4...4...5...5...6...6...7..
0 5 0 5 0 5 0 5 0 5 0 5 0 5 0
Kpps per five min (last 6 hrs)

```

Here is an example of the **show crypto throughput distribution history** command:

```

wsg# show crypto throughput distribution history ixp 2
% Throughput Utilization bucket                               Number of Intervals

 1 - 25                                                         0
26 - 50                                                         0
51 - 60                                                         4
61 - 65                                                         0
66 - 70                                                         0
71 - 75                                                         0
76 - 80                                                         0
81 - 82                                                         0
83 - 84                                                         0
85 - 86                                                         0
87 - 88                                                         0
89 - 90                                                         0
91 - 92                                                         0
93 - 94                                                         0
95 - 96                                                         0
97 - 98                                                         0
99 - 100                                                        1

```

Traffic distribution — Hash distribution

To distribute the post encryption traffic among 2 IXPs, a hash table is programmed. The PPC can program this hash table in 2 ways:

- Scheme 1 – Sequential Distribution of hash entries
- Scheme 2 – Random Distribution of hash entries (recommended)

Based on traffic type and distribution of user source/destination ip addresses, an administrator can use either of the scheme to get the best throughput utilization results between both the IXPs. Overall or per IXP utilization can be displayed by the crypto throughput CLIs in PPC as explained above.

Command	Purpose
<pre>switch (config)# crypto clear-traffic load <50%-100%></pre>	<p>Sets the number of punt entries to be programmed into traffic distribution hash table in IXP0 based on the current Clear traffic load %.</p> <ul style="list-style-type: none"> • load — Percentage of clear traffic load <ul style="list-style-type: none"> - 50% — IXP0 is handling 50% of total incoming traffic. No punt entries will be programmed. . . . 100% — IXP0 is handling 100% of total incoming traffic.
<pre>switch (config)# crypto clear-traffic switch-distribution-sch eme <1/2></pre>	<p>Sets the traffic distribution hash table in IXP0 either with sequential punt entries or random punt entries.</p> <ul style="list-style-type: none"> • switch-distribution-scheme — Selects the scheme number <ul style="list-style-type: none"> - 1 — Sequential hashing - 2 — Random hashing (default)

Given below is the IXP show command executed only from IXP0 (proc 1):

Command	Purpose
<pre>ucdump -t puntbl</pre>	<p>Displays the entries programmed in the traffic distribution hash table in IXP0 when the CLI commands are executed from the PPC to enable the traffic distribution.</p>

Here is an example of the **ucdump -t puntbl** command:

```
# ucdump -t puntbl
Punt table entry base is 0x907d0e10 End is 0x907d1210
=====
Word Offset   Hash Addr    Count Addr   Access Count
-----
0             0x1800e10    0x1802210    596587
1             0x1800e14    0x1802214    602678
2             0x1800e18    0x1802218    602680
3             0x1800e1c    0x180221c    602683
4             0x1800e20    0x1802220    602687
:
:
:
250           0x18011f8    0x18025f8    603275
252           0x1801200    0x1802600    603278
254           0x1801208    0x1802608    603282
Test Summary
Total Punt Entries: 192
Punt Entries Accessed: 192
Total Packets: 154715912
Total Packets Punted: 116112365
Achieved Punt %: 75.05
```

Configuring IKE/IPSec Stats Collection and Timing Enhancements for SNMP

To configure the statistics refresh interval for SNMP in manual mode, or be set automatically (auto mode) based on number of IPSec tunnels, perform the following tasks:

	Command	Purpose
Step 1	<code>switch# config</code>	Enters global configuration mode.
Step 2	<code>switch (config)# crypto snmp stats-refresh-interval {auto manual interval}</code>	<p>Configure the statistics refresh interval for SNMP. It can be a fixed interval (manual mode), or be set automatically (auto mode) based on number of IPSec tunnels.</p> <ul style="list-style-type: none"> • auto — Sets referesh interval automatically based on number of tunnels, on average about 1.5 sec for 1000 tunnels. • <i>interval</i> — Set refresh interval manually in range from 1 to 300 sec.

```
WSG# config
```

Enter configuration commands, one per line. End with CNTL/Z.

```
WSG (config)# crypto snmp stats-refresh-interval auto
```

