61000 Security Systems
CLI
R75.050, R75.051, R75.052
Reference Guide

10 June 2013
Important Information

Latest Software
We recommend that you install the most recent software release to stay up-to-date with the latest functional improvements, stability fixes, security enhancements and protection against new and evolving attacks.

This document is relevant only to 61000 R75.050 version.

Latest Documentation
The latest version of this document is at:
http://supportcontent.checkpoint.com/documentation_download?ID=18161

For more about this appliance, see the Check Point Data Center Security Appliances 61000 R75.050 home page (http://supportcontent.checkpoint.com/solutions?id=sk77880)

For additional technical information, visit the Check Point Support Center (http://supportcenter.checkpoint.com).

Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 June 2013</td>
<td>New: Licensing and Registration section</td>
</tr>
<tr>
<td>1 May 2013</td>
<td>New: Changing SSM160 admin password</td>
</tr>
<tr>
<td>24 April 2013</td>
<td>Modified: HTU – HA Time Unit (0.1s)</td>
</tr>
<tr>
<td></td>
<td>asg_session_control - session rate throttling.</td>
</tr>
<tr>
<td></td>
<td>DC deployment section.</td>
</tr>
<tr>
<td></td>
<td>asg_stat - Bond monitoring - logical interface health for chassis grade calculation.</td>
</tr>
<tr>
<td></td>
<td>asg_sync_manager - support connections synchronization to all SGMs. (user defined)</td>
</tr>
<tr>
<td></td>
<td>asg_alert – route alert added</td>
</tr>
<tr>
<td>6 March 2013</td>
<td>New: Hardware troubleshooting section</td>
</tr>
<tr>
<td>2 January 2013</td>
<td>Edit Configuring a Unique IP address per Chassis (UIPC): Unique IP can be added to one of the data ports.</td>
</tr>
<tr>
<td>31 December 2012</td>
<td>New: Jumbo Frame configuration on Bond Interface.</td>
</tr>
<tr>
<td></td>
<td>Remove the note from Jumbo Frame that Jumbo Frame is not supported on Bond.</td>
</tr>
<tr>
<td></td>
<td>Correct Configuring Chassis High Availability description.</td>
</tr>
<tr>
<td></td>
<td>Add the Section Single SSM to Setting port priority (for each port) under Configuring Chassis High Availability</td>
</tr>
<tr>
<td>18 December 2012</td>
<td>New: VPN LTE</td>
</tr>
<tr>
<td></td>
<td>New: SCTP Acceleration</td>
</tr>
<tr>
<td></td>
<td>Edit: DPD, Inner packet fragmentation, VPN Sticky SA – gateway configuration is automatically configure when configuring LTE.</td>
</tr>
<tr>
<td></td>
<td>Remove the comment section from Time synchronization from NTP server</td>
</tr>
<tr>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3 December 2012</td>
<td>Edit verifying phase of <strong>Upgrading Chassis Software</strong></td>
</tr>
<tr>
<td></td>
<td>New: Replacing the CMM</td>
</tr>
<tr>
<td></td>
<td>Edit asg_ntp_sync_config usage</td>
</tr>
<tr>
<td>19 November 2012</td>
<td>New: <strong>Management Port Speed Configuration</strong></td>
</tr>
<tr>
<td></td>
<td>Edit: Configuring Chassis High Availability – add the section ‘Synchronizing Clusters on a Wide Area Network’</td>
</tr>
<tr>
<td></td>
<td>Clarification: chassis HA in ‘L2 Bridge Mode configuration’.</td>
</tr>
<tr>
<td></td>
<td>Correct ‘Resetting SIC procedure’: run gclish commands.</td>
</tr>
<tr>
<td></td>
<td>Correct the command set snapshot import in ‘Upgrading Chassis Software’ procedure</td>
</tr>
<tr>
<td></td>
<td>New: IPS Bypass under Load</td>
</tr>
<tr>
<td></td>
<td>New: IPS Cluster Failover Management</td>
</tr>
<tr>
<td></td>
<td>New: IPS PacketsCapture</td>
</tr>
<tr>
<td>6 November 2012</td>
<td>Corrected procedure for <strong>Upgrading Chassis Software</strong></td>
</tr>
<tr>
<td></td>
<td>Clarification: Upgrading SSM Firmware – no need to upgrade SSM60 firmware</td>
</tr>
<tr>
<td></td>
<td>New: 61000 Component Versions</td>
</tr>
<tr>
<td></td>
<td>Improved formatting and document layout</td>
</tr>
<tr>
<td>29 October 2012</td>
<td>Added procedure for <strong>Upgrading SSM Firmware</strong></td>
</tr>
<tr>
<td>25 Oct 2012</td>
<td>Improved example output for asg_version</td>
</tr>
<tr>
<td></td>
<td>Fixed asg_dxl calc example</td>
</tr>
<tr>
<td></td>
<td>Updated build numbers in the list of Official the 61000 Security System version names.</td>
</tr>
<tr>
<td>11 Oct 2012</td>
<td>Fixed Typo</td>
</tr>
<tr>
<td>19 Sep 2012</td>
<td>New: Chassis HA – Sync Lost Mechanism, SyncXL.</td>
</tr>
<tr>
<td>26 Aug 2012</td>
<td>asg_drop_monitor</td>
</tr>
<tr>
<td>7 Aug 2012</td>
<td>Fix asg_br_verifier examples</td>
</tr>
<tr>
<td>31 July 2012</td>
<td>Software upgrade procedure , ECMP enhancement , asg_syslog, DHCP relay, port speed configuration, port mirroring, multicast</td>
</tr>
<tr>
<td>18 July 2012</td>
<td>Radius, VPN enhancements, asg diag, SSMs upgrade procedure, Chassis ID configuration, the 61000 Security System LEDs</td>
</tr>
<tr>
<td>1 July 2012</td>
<td>MSS Adjustment, Session Control, VPN enhancements, Reserved connections, Fast Packet Drop, RMA Procedure</td>
</tr>
<tr>
<td>15 May 2012</td>
<td>Jumbo Frames adjustments</td>
</tr>
<tr>
<td></td>
<td>asg stat - Sync health display</td>
</tr>
<tr>
<td>10 May 2012</td>
<td>Monitor Management Interfaces</td>
</tr>
<tr>
<td></td>
<td>asg log ports</td>
</tr>
<tr>
<td></td>
<td>GARP chunk mechanism</td>
</tr>
<tr>
<td>6 May 2012</td>
<td>Fix tcpdump example</td>
</tr>
<tr>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>19 April 2012</td>
<td>Bridge support</td>
</tr>
<tr>
<td></td>
<td>asg utilities name revision</td>
</tr>
</tbody>
</table>
Feedback

Check Point is engaged in a continuous effort to improve its documentation. Please help us by sending your comments (mailto:cp_techpub_feedback@checkpoint.com?subject=Feedback on 61000 Security Systems CLI Reference Guide).
Important Information.............................................................................................................3
Common System Commands ........................................................................................................10
  Showing Chassis and Component State (asg stat) ..................................................................10
  Setting the Required Number of Components per Chassis ...............................................14
  Setting the Unit Weight .....................................................................................................14
  Setting the Bond Min Slaves .............................................................................................15
  Monitoring Chassis and Component Status (asg monitor) ................................................16
  Monitoring Key Performance Indicators and Load Statistics (asg perf) ..........................18
  Showing Hardware Information for Monitored Components (asg hw_monitor) ...........20
  Showing Security Gateway Module Resource Information (asg resource) .................23
  Searching for a Connection (asg search) ..........................................................................24
  Configuring Alerts for SGM and Chassis Events (asg alert) ..........................................25
  Collecting System Diagnostics (asg diag) ........................................................................28
Monitoring and Information Gathering ......................................................................................35
  Networking ..........................................................................................................................35
    Showing Interface Status (asg if) ..................................................................................35
    Showing Traffic Information (asg_ifconfig) ..................................................................36
    Showing the Routing Tables (asg_route) .......................................................................44
    Connecting to a specific SGM (blade command) .........................................................47
    VPN Packet Tracking ....................................................................................................47
    Showing SSM Traffic Statistics (asg_traffic_stats) .....................................................48
    Showing SGM Information (asg_blade_stats) ...............................................................49
    Multi-blade capture (tcpdump –mcap) ...........................................................................49
    Traceroute (asg_tracert) ...............................................................................................50
  Hardware ............................................................................................................................51
    Collecting System Serial Numbers ..............................................................................51
  Security ..................................................................................................................................53
    Showing the Number of Firewall and SecureXL Connections (asg_conns) .........53
    IPS Packet Capture (asg_get_capture) ........................................................................54
    Packet drop monitoring (asg_drop_monitor) ...............................................................56
    Log Utility (asg_log) ......................................................................................................56
  Other ....................................................................................................................................58
    Official 61000 Security System Versions ....................................................................58
    Showing the Auditlog (asg_auditlog) ............................................................................59
    Showing the firewall Database Configuration (asg_config) ....................................61
    Showing Software and Firmware versions (asg_version) .....................................62
    Displaying System Messages (asg_varlog) .................................................................65
    Monitoring the System using SNMP ............................................................................65
System Optimations ..................................................................................................................67
  Firewall connections table size .........................................................................................67
  Reserved connections .......................................................................................................67
  Policy Acceleration – SecureXL Keep Connections .........................................................70
  Extending SecureXL Templates .....................................................................................71
  Configuring DNS Session Rate (cphwd_udp_selective_delay_max) .........................73
  Configuring VLAN performance enhancement (asg_affinity_enhance) ....................75
  Fast packet drop ..............................................................................................................75
  System Under Load .........................................................................................................77
  Jumbo Frames ..................................................................................................................79
    Validation ......................................................................................................................81
    Jumbo Frame configuration on Bond LACP Interface .................................................82
  No Acceleration Disabling On Account Of Traceroute Rule (asg_tmpl_special_srvcs) (Applies to R75.052 only) ..............................................................88
NAT and the Correction Layer .................................................89
Session rate throttling (Applies to R75.052 only) ....................89
System Configuration ............................................................91
  Administration .....................................................................91
gclish .................................................................................91
  CP global commands ..........................................................92
  OS global commands ..........................................................95
  Global Commands Generated by CMM .................................99
  Time synchronization from NTP server (asg_ntp_sync_config) ...106
  Backup and Restore (backup_system) ....................................107
High Availability .................................................................109
  Configuring Chassis High Availability .................................109
  Chassis ID Configuration ...................................................114
  Configuring a Unique IP address per Chassis (UIPC) ............116
  asg_sync_manager ..............................................................117
Monitoring, Logs and Auditing .............................................120
  Redirecting Alerts Messages and Firewall logs to External syslog server (asg_syslog) .................................................................120
  Monitoring Management Interfaces Link State .....................123
  Log Server Distribution (asg_log_servers) ............................125
  Configuring a Dedicated Logging Port ..................................127
  Command Auditing .............................................................128
Security ...............................................................................129
  VPN ..................................................................................129
  DPD ..................................................................................131
  SCTP Acceleration .............................................................133
  Configuring the 6in4 Internet Transition Mechanism .............139
  Generic Routing Encapsulation – GRE (asg_gre) ..................141
  Role Based Administration (RBA) .......................................142
  RADIUS Authentication .....................................................144
Networking ...........................................................................148
  Configuring Link Aggregation (Bonding) ..............................150
  Configuring VLANs ............................................................152
  Configuring Dynamic Routing - Unicast ...............................153
  Changing the Default VMAC (asg_unique_mac_utility) ..........155
  Configuring Source Based Routing ......................................158
  ECMP Configuration ..........................................................159
  Proxy ARP for Manual NAT – (local.arp file) ......................163
  Port speed configuration .....................................................164
  Multicast Configuration ......................................................170
  DHCP relay .......................................................................175
  Configuring IPv6 Support (asg_dxl_ipv6) ............................177
61000 Concepts ..................................................................179
  Policy Installation and the Single Management Object ..........179
  Policy and Configuration Cloning ........................................181
  Cloning the Firewall Policy .................................................181
  Cloning the Configuration ..................................................181
  MAC Addresses and Bit Conventions ..................................182
    Verifying the MAC Address (asg_mac_resolver) ..................183
  SyncXL .............................................................................184
  Security Group (asg_security_group) ....................................185
  Layer-2 Bridge Mode configuration ......................................186
  Distribution Modes ..............................................................191
    Manually Configuring Distribution Modes (asg_dxl_dist_mode)193
    Reconfiguring Distribution Modes ......................................194
  Chassis Control (asg_chassis_ctl) .......................................196
  Hybrid System .....................................................................199
  GARP chunk mechanism ....................................................199

 Protected] For public distribution
Troubleshooting

Licensing and Registration

Software Blades Support

Software Upgrade and Hardware Replacement

Hardware Reference
Chapter 1

Common System Commands

Showing Chassis and Component State (asg stat)

Description
Use this command to show the chassis and component state for single and dual chassis configurations. The command shows System information:

- Up-time
- CPU load: average and concurrent
- Concurrent connections
- System Health
- Hardware component status: the number of components that are up compared to the required number.
- SGM status in terms of (verbose mode):
  - State
  - Policy
  - Process

Syntax

asg stat [-v]

asg stat -i [ tasks | proc | all_ids | all_sync_ips | local_id | active_ids | chassis_monitor

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Shows</th>
</tr>
</thead>
<tbody>
<tr>
<td>(none)</td>
<td>Chassis status</td>
</tr>
<tr>
<td>-v</td>
<td>Verbose chassis information.</td>
</tr>
<tr>
<td>-i tasks</td>
<td>Task distribution on SGMs</td>
</tr>
<tr>
<td>proc</td>
<td>Overall system processes</td>
</tr>
<tr>
<td>all_ids</td>
<td>All SGMs detected since last reboot</td>
</tr>
<tr>
<td>all_sync_ips</td>
<td>Sync IPs of SGMs in the “all_ids” SGM list</td>
</tr>
<tr>
<td>local_id</td>
<td>Local SGM ID</td>
</tr>
<tr>
<td>active_ids</td>
<td>All SGMs whose state is UP</td>
</tr>
</tbody>
</table>
Parameter | Shows
---|---
chassis_monitor | Which SGM handles the `chassis_monitor` process (usually the local SGM)

**Example 1**

```
asg stat
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current CPUs load average</td>
<td>4 %</td>
</tr>
<tr>
<td>Concurrent connections</td>
<td>6</td>
</tr>
<tr>
<td>Health</td>
<td>NORMAL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis</th>
<th>SGMs</th>
<th>Ports</th>
<th>Bonds</th>
<th>Fans</th>
<th>SSMs</th>
<th>CMs</th>
<th>Power Supplies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis 1</td>
<td>2 / 2</td>
<td>3 / 3</td>
<td>1 / 1</td>
<td>4 / 4</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td>6 / 6</td>
</tr>
<tr>
<td>Chassis 2</td>
<td>2 / 2</td>
<td>3 / 3</td>
<td>1 / 1</td>
<td>4 / 4</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td>6 / 6</td>
</tr>
</tbody>
</table>

**Comments**
The output shows that:

- Chassis 1 is in STANDBY state.
- 9 SGMs in Chassis 1 are UP, out of the 12 that are required
- All other components are up and running according to the predefined settings

**Example 2**

```
asg stat -v
```

<table>
<thead>
<tr>
<th>Chassis Status</th>
<th>ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis 1</td>
<td></td>
</tr>
<tr>
<td>SGM ID</td>
<td>State</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1 (local)</td>
<td>UP</td>
</tr>
<tr>
<td>2</td>
<td>UP</td>
</tr>
<tr>
<td>3</td>
<td>UP</td>
</tr>
<tr>
<td>4</td>
<td>UP</td>
</tr>
<tr>
<td>5</td>
<td>UP</td>
</tr>
<tr>
<td>6</td>
<td>UP</td>
</tr>
<tr>
<td>7</td>
<td>UP</td>
</tr>
<tr>
<td>8</td>
<td>UP</td>
</tr>
</tbody>
</table>

<p>| Chassis 2 |</p>
<table>
<thead>
<tr>
<th>SGM ID</th>
<th>State</th>
<th>Process</th>
<th>Policy date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>17Aug11 18:10</td>
</tr>
<tr>
<td>2</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>17Aug11 18:10</td>
</tr>
<tr>
<td>3</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>17Aug11 18:10</td>
</tr>
<tr>
<td>4</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>17Aug11 18:10</td>
</tr>
<tr>
<td>5</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>17Aug11 18:10</td>
</tr>
<tr>
<td>6</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>17Aug11 18:10</td>
</tr>
<tr>
<td>7</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>17Aug11 18:10</td>
</tr>
<tr>
<td>8</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>17Aug11 18:10</td>
</tr>
</tbody>
</table>
Comments

- (local)
  Represents the SGM on which the command `asg stat -v` was run.

- State

<table>
<thead>
<tr>
<th>State</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP</td>
<td>The SGM is processing traffic</td>
</tr>
<tr>
<td>DOWN</td>
<td>The SGM is not processing traffic</td>
</tr>
<tr>
<td>DETACHED</td>
<td>No SGM has been detected in a slot</td>
</tr>
</tbody>
</table>

Note - To manually change the state of an SGM to or from 'administratively down', use: `asg sgm_admin`.

- Process
  The process state of the SGM, whether the SGM is:

  - **Enforcing Security.** The SGM is UP and working properly.
  - **Inactive.** The SGM is inactive because its *State* is: DOWN or DETACHED.
  - **Initial Policy.** The SGM's state is UP but a policy not installed.

- Chassis Grade
  Each component in the chassis, such as a fan or port, has a certain “weight”. The weight is a numerical value which reflects the level of importance you attach to a component. For example if Ports are more important to you than fans user may assign ports a higher value or a greater weight. The chassis grade is the sum of all these component weights.

  In a dual-chassis deployment, the chassis with the higher grade becomes ACTIVE. For example, if ports have a greater weight than fans and many ports go DOWN, this will drop the chassis grade and cause a failover to the STANDBY chassis, which has the higher grade at that point.

  The grade of each component = \((\text{Unit Weight}) \times (\text{Number of components that are UP})\)

  - To reflect the importance of a component in the system, the component's Unit Weight can be configured. For example if you wish to change the weight of SGM from 6 to 12, run:

    `set chassis high-availability factors sgm 12`
If you run `asg stat --v` again, the output shows a greater unit weight per SGM and an higher Chassis Grade than before:

```
<table>
<thead>
<tr>
<th>System Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SGM ID</th>
<th>State</th>
<th>Process</th>
<th>Policy Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (local)</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>06Mar13 04:37</td>
</tr>
<tr>
<td>2</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>06Mar13 04:37</td>
</tr>
</tbody>
</table>

| Chassis 2 | STANDBY |

<table>
<thead>
<tr>
<th>SGM ID</th>
<th>State</th>
<th>Process</th>
<th>Policy Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>06Mar13 04:37</td>
</tr>
<tr>
<td>2</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>06Mar13 04:37</td>
</tr>
</tbody>
</table>

| Chassis Parameters |

<table>
<thead>
<tr>
<th>Unit</th>
<th>Chassis 1</th>
<th>Chassis 2</th>
<th>Unit Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGMs</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td><strong>12</strong></td>
</tr>
<tr>
<td>Ports</td>
<td>Standard</td>
<td>1 / 1</td>
<td>1 / 1</td>
</tr>
<tr>
<td>Bond</td>
<td>1 / 1</td>
<td>1 / 1</td>
<td><strong>11</strong></td>
</tr>
<tr>
<td>Other</td>
<td>0 / 0</td>
<td>0 / 0</td>
<td>6</td>
</tr>
<tr>
<td>Sensors</td>
<td>Fans</td>
<td>4 / 4</td>
<td>4 / 4</td>
</tr>
<tr>
<td>Bond</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td><strong>11</strong></td>
</tr>
<tr>
<td>CSMs</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td>6</td>
</tr>
<tr>
<td>Power Supplies</td>
<td>6 / 6</td>
<td>6 / 6</td>
<td>6</td>
</tr>
</tbody>
</table>

| Chassis Grade | **136 / 136** | **136 / 136** |

| Minimum grade gap for chassis failover | 11 |
| Synchronization | |
| Within chassis | Enabled (Default) |
| Between chassis | Enabled (Default) |
| Exception Rules | |
| Distribution | |
| Control Block | Disabled (Default) |
| Chassis HA mode | Active Up |
| Chassis HA in Freeze | (25 seconds left) |
```

Failure of an SGM with this high unit value will cause a chassis failover, as the minimum default grade gap for chassis failover is 11.

**Minimum grade gap for chassis failover**

Minimum grade gap is the value which determines when a chassis fails over. If the active chassis grade drops by the "minimum grade gap" failover may occur. The active chassis is always the chassis whose grade is higher by at least the minimum grade gap.

- **Synchronization**
  - **Within chassis** Whether synchronization is enabled between SGMs in the same chassis
  - **Between chassis** Whether synchronization is enabled between SGMs in different chassis
  - **Exception Rules** Whether the user has configured any synchronization exception rules using the `asg_sync_manager` commands
  - **Sync Ports** Whether there are link problems on the Sync interfaces (will not appear if there are no issues)
Distribution
Control blade

Whether the control blade feature is enabled. The control blade feature sets the SMO not to handle data traffic, only management traffic. When the feature is enabled, there is always access to the system through an SSH connection.

---

**Setting the Required Number of Components per Chassis**

Running `asg stat` shows the chassis and component state for single and dual chassis configurations. `asg stat` also shows the hardware components status: the number of up/active components compared to the required number.

To change the required number of any component on a chassis, run these commands in `gclish`:

<table>
<thead>
<tr>
<th>Component</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGMs</td>
<td><code>asg security_group</code></td>
</tr>
<tr>
<td>Ports</td>
<td><code>&gt; set interface &lt;port&gt; state on</code></td>
</tr>
<tr>
<td>Fans</td>
<td><code>&gt; set chassis id 1 modules_amount fans 3</code></td>
</tr>
<tr>
<td></td>
<td>Sets the required number of fans on chassis 1 to three</td>
</tr>
<tr>
<td>SSM</td>
<td><code>&gt; set chassis id 1 modules_amount SSM 2</code></td>
</tr>
<tr>
<td></td>
<td>Sets the required number of SSMs on chassis 1 to 2</td>
</tr>
<tr>
<td>CMM</td>
<td><code>&gt; set chassis id 1 modules_amount CMM 2</code></td>
</tr>
<tr>
<td></td>
<td>Sets the required number of CMMs on chassis 1 to 2</td>
</tr>
<tr>
<td>Power Supply</td>
<td><code>&gt; set chassis id 1 modules_amount power_units 3</code></td>
</tr>
<tr>
<td>units</td>
<td>Sets the required number of power units on chassis 1 to 3</td>
</tr>
</tbody>
</table>

---

**Setting the Unit Weight**

Running `asg stat` shows the chassis and component state for single and dual chassis configurations. The command also shows the Unit Weight. To reflect the importance of a component in the system, the component's Unit Weight can be configured.

To change the Unit Weight, run these commands in `gclish`:

- `> set chassis high-availability factors` (or `> set chassis high-availability factors port`)

<table>
<thead>
<tr>
<th>Component</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGMs</td>
<td><code>&gt; set chassis high-availability factors sgm &lt;factor&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA SGM Factor</td>
</tr>
<tr>
<td>bond</td>
<td><code>&gt; set chassis high-availability factors port bond &lt;factor&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA Bond Port Factor</td>
</tr>
<tr>
<td>Mgmt</td>
<td><code>&gt; set chassis high-availability factors port bond &lt;factor&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA Mgmt Port Factor</td>
</tr>
</tbody>
</table>
### Setting the Bond Min Slaves

**Description:**

Running `asg stat` shows the chassis and component state for single and dual chassis configurations. Bond monitoring is one of the chassis components in asg stat. The bond component’s Min Slaves can be configured through gclish.

**Syntax:**

-  > set chassis high-availability bond <bond port> min_slaves

**Example:**

  - set chassis high-availability bond bond1 min_slaves 2

**Note:**

The default value for min slaves is 1, bond will consider down if all its slaves are down.
Monitoring Chassis and Component Status (asg monitor)

**Description**
Use this command to show the chassis and components state for single chassis and dual chassis configurations.

**Syntax**
```
asg monitor [interval][ -v [interval]] [ -all interval]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interval</td>
<td>Monitors SGM state and running processes. Enter a decimal value in seconds, for example: asg monitor 3</td>
</tr>
<tr>
<td>-v interval</td>
<td>Monitors chassis parameters. For example: asg monitor -v 3.</td>
</tr>
<tr>
<td>-all interval</td>
<td>Monitors all SGMs and chassis parameters</td>
</tr>
</tbody>
</table>

**Example 1**
```
asg monitor
```
```
gamin300-ch01-01 > asg monitor
Thu Mar 29 15:58:13 IST 2012

<table>
<thead>
<tr>
<th>Chassis 1</th>
<th>ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGM ID</td>
<td>State</td>
</tr>
<tr>
<td>1 (Local)</td>
<td>UP</td>
</tr>
<tr>
<td>2</td>
<td>UP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis 2</th>
<th>STANDBY</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGM ID</td>
<td>State</td>
</tr>
<tr>
<td>1</td>
<td>UP</td>
</tr>
<tr>
<td>2</td>
<td>UP</td>
</tr>
<tr>
<td>3</td>
<td>UP</td>
</tr>
</tbody>
</table>
Monitoring Chassis and Component Status (asg monitor)

Comments
This shows:
- The date and time when information was last collected
- Chassis 1 is ACTIVE with three Security Gateway Modules up
- Chassis 2 is in STANDBY state with three Security Gateway Modules up
- Security Gateway State is the state of the Security Gateway Module. The state can be
  - Up
  - Down
  - Detached
A state can have one of these Processes:
- Enforcing Security - The SGM is UP and working properly.
- Inactive - The SGM is DOWN, and is experiencing some problem. It is not handling any traffic.
- Initial policy - The policy is not installed on the SGM.
To manually change the state of an SGM, use the asg sgm_admin command. Remember that this command administratively changes the state to up or down. An SGM which is physically down cannot be changed to UP using this command.

(local) - represents the SGM on which you ran the command.

Example 2
asg monitor -v

Output
```bash
$ asg monitor -v
Thu Mar 29 16:06:41 IST 2012

+-----------------+-----------------+-----------------+-----------------+
<table>
<thead>
<tr>
<th>Chassis Parameters</th>
<th>Chassis 1</th>
<th>Chassis 2</th>
<th>Unit Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGMs</td>
<td>3 / 3</td>
<td>3 / 3</td>
<td>6</td>
</tr>
<tr>
<td>Ports Priority</td>
<td>Standard</td>
<td>2 / 2</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>0 / 0</td>
<td>0 / 0</td>
</tr>
<tr>
<td>Sensors</td>
<td>Fems</td>
<td>4 / 4</td>
<td>4 / 4</td>
</tr>
<tr>
<td></td>
<td>SMFs</td>
<td>2 / 2</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>CMFs</td>
<td>2 / 2</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>Power Supplies</td>
<td>6 / 6</td>
<td>6 / 6</td>
</tr>
<tr>
<td>Chassis Grade</td>
<td>130 / 130</td>
<td>130 / 130</td>
<td></td>
</tr>
<tr>
<td>Minimum grade gap for chassis failover:</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronization</td>
<td>Return chassis: Enabled (Default)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Between chassis: Enabled (Default)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Exception Rules: Enabled (Default)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distribution</td>
<td>Control Blade: Disabled (Default)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chassis HI mode: Active Up</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
Monitoring Key Performance Indicators and Load Statistics (asg perf)

**Comments**

- The (number/number) convention presents the number of components actually up set against the number of components required to be up. For example SGMs 3/3 means that 3 SGMs are up and 3 are required to be up.

- **Chassis grade** is the sum of all components grades. The grade of each component = (Unit Weight)x(Number of UP components). The One Unit Weight of each component can be configured to reflect the importance of the component in the system. To configure the One Unit Weight run:
  - `set chassis high-availability factors <sensor name>`

- **Minimum grade gap for chassis failover** - Chassis failover occurs to the chassis with the higher grade only if its grade is greater than the other chassis by more than the minimum gap.

- **Synchronization** - The status of synchronization:
  - Within chassis - between SGMs located in the same chassis
  - Between chassis - between SGMs located in different chassis
  - Exception Rules - user configured exception rules. To configure, use the command `g_sync_exception`

**Monitoring Key Performance Indicators and Load Statistics (asg perf)**

**Description**

Use this command to continuously monitor key performance indicators and load statistics.

**Syntax**

`asg perf [-b blades][-v][-p][-a][-k]`
## Monitoring Key Performance Indicators and Load Statistics (asg perf)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-b blades</td>
<td>List of Security Gateway Modules. For example:</td>
</tr>
<tr>
<td>1_01</td>
<td>Chassis 1 SGM 1</td>
</tr>
<tr>
<td>1_03-1_05</td>
<td>Chassis 1 SGMs 3, 4 and 5.</td>
</tr>
<tr>
<td>1_01,1_03-1_05</td>
<td>Combination of previous two items</td>
</tr>
<tr>
<td>all</td>
<td>All SGMs (including chassis 2, if applicable)</td>
</tr>
<tr>
<td>chassis1</td>
<td>All SGMs in Chassis 1</td>
</tr>
<tr>
<td>chassis2</td>
<td>All SGMs in chassis 2</td>
</tr>
<tr>
<td>chassis_active</td>
<td>All SGMs in the active chassis</td>
</tr>
</tbody>
</table>

- **v**  
  Verbose mode: Per-Security Gateway Module display.  
  Show performance statistics (including load and acceleration load) on the active chassis.

- **p**  
  Show detailed statistics and traffic distribution between these paths on the active chassis:  
  - Acceleration path (Performance Pack).  
  - Medium path (PXL).  
  - Slow path (Firewall).

- **a**  
  Show absolute values.

- **k**  
  Shows peak values for connection rate, concurrent connections and throughput.

- **h**  
  Display usage.

### Example 1

If no SGMs are specified, the following shows performance statistics on the active chassis:

```bash
dg perf -v
```

### Output

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput</td>
<td>3.1 Gbps</td>
</tr>
<tr>
<td>Connection rate</td>
<td>4.9 kbps</td>
</tr>
<tr>
<td>Packet rate</td>
<td>448.9 kbps</td>
</tr>
<tr>
<td>Concurrent connections</td>
<td>26.3 %</td>
</tr>
<tr>
<td>Load average</td>
<td>6.5 %</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Acceleration load (avg/min/max)</th>
<th>%/%/3%/15%</th>
<th>Instances load (avg/min/max)</th>
<th>%/%/3%/10%</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SGM ID</th>
<th>Throughput</th>
<th>Conn. rate</th>
<th>Packet rate</th>
<th>Concurrent Comm.</th>
<th>Accel. Core usage</th>
<th>Instances Core usage</th>
<th>Load (avg/min/max)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_01</td>
<td>15.4%</td>
<td>15.8%</td>
<td>16.6%</td>
<td>15.9%</td>
<td>9/3/20</td>
<td>6/3/10</td>
<td></td>
</tr>
<tr>
<td>1_02</td>
<td>15.6%</td>
<td>15.7%</td>
<td>11.7%</td>
<td>15.2%</td>
<td>8/3/19</td>
<td>5/2/19</td>
<td></td>
</tr>
<tr>
<td>1_03</td>
<td>15.5%</td>
<td>14.8%</td>
<td>11.7%</td>
<td>15.2%</td>
<td>8/3/21</td>
<td>4/2/6</td>
<td></td>
</tr>
<tr>
<td>1_04</td>
<td>15.1%</td>
<td>18.2%</td>
<td>11.2%</td>
<td>18.8%</td>
<td>9/5/22</td>
<td>6/2/16</td>
<td></td>
</tr>
<tr>
<td>1_05</td>
<td>19.0%</td>
<td>19.3%</td>
<td>19.5%</td>
<td>19.3%</td>
<td>11/4/23</td>
<td>5/2/8</td>
<td></td>
</tr>
<tr>
<td>1_06</td>
<td>15.9%</td>
<td>18.5%</td>
<td>11.5%</td>
<td>19.5%</td>
<td>9/3/21</td>
<td>5/3/10</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Comments**  
Load Average = CPU load.
Showing Hardware Information for Monitored Components (asg hw_monitor)

Description
Use this command to show per-chassis hardware information and thresholds for monitored components, including:

- Security Gateway Module: CPU temperatures per CPU socket.
- Chassis fan speeds.
- Security Switch Module: throughput rates.
- Power consumption per chassis.
- Power Supply Unit: Whether installed or not.
- Chassis Management Module: Whether installed or not, and active or standby.

Syntax
asg hw_monitor [-v] [-f filter]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Retrieve hardware monitor information</td>
</tr>
<tr>
<td>-v</td>
<td>Verbose mode</td>
</tr>
<tr>
<td>-f</td>
<td>Show only filtered components (one or more than one filter can be used)</td>
</tr>
<tr>
<td>filter</td>
<td>CMM CPUtemp Fan PowerConsumption PowerUnit SSM</td>
</tr>
</tbody>
</table>

Example
asg hw_monitor
### Chassis 1

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Location</th>
<th>Value</th>
<th>Threshold</th>
<th>Units</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMM</td>
<td>Bay 1</td>
<td>0</td>
<td>0</td>
<td>&lt;S,D&gt;/&lt;A&gt;</td>
<td>1</td>
</tr>
<tr>
<td>CMM</td>
<td>Bay 2</td>
<td>1</td>
<td>0</td>
<td>&lt;S,D&gt;/&lt;A&gt;</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 1, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 1, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 2, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 2, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 3, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 3, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 4, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 4, CPU1</td>
<td>42</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 5, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 5, CPU1</td>
<td>42</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 6, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 6, CPU1</td>
<td>42</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 7, CPU0</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 7, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 8, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 8, CPU1</td>
<td>44</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 9, CPU0</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 9, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 10, CPU0</td>
<td>47</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 10, CPU1</td>
<td>44</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 11, CPU0</td>
<td>39</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 11, CPU1</td>
<td>42</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 12, CPU0</td>
<td>39</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUtemp</td>
<td>Blade 12, CPU1</td>
<td>43</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis 2</th>
<th>Location</th>
<th>Value</th>
<th>Threshold</th>
<th>Units</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerConsumption</td>
<td>N/A</td>
<td>3950</td>
<td>4050</td>
<td>Watts</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnit(AC)</td>
<td>Bay 1</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnit(AC)</td>
<td>Bay 2</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnit(AC)</td>
<td>Bay 3</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnit(AC)</td>
<td>Bay 4</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnit(AC)</td>
<td>Bay 5</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 1</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 1</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 1</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 1</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 2</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 2</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 2</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 2</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 3</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 3</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 3</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 3</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 4</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 4</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 4</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 4</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 5</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 5</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 5</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 5</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 6</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 6</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 6</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 6</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 7</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 7</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 7</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 7</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 8</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 8</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 8</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 8</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 1, Fan 9</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 2, Fan 9</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 3, Fan 9</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>Bay 4, Fan 9</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>0</td>
</tr>
</tbody>
</table>

### Common System Commands

- **show hardware information**
  - Command: `show hardware information`
  - Output:
    ```plaintext
    Showing Hardware Information for Monitored Components (asg hw_monitor)
    ```

- **Common System Commands**
  - Page: 21

---

*Protected* For public distribution
### Comments

<table>
<thead>
<tr>
<th>Column</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>To identify the location, see the 61000 Security Systems Front Panel.</td>
</tr>
<tr>
<td>Value</td>
<td>Most components have a defined threshold value. The threshold gives an indication of the health and functionality of the component. When the value of the resource is greater than the threshold, an alert is sent (&quot;Configuring Alerts for SGM and Chassis Events (asg alert)&quot; on page 56).</td>
</tr>
<tr>
<td>Threshold</td>
<td></td>
</tr>
<tr>
<td>Units</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>0 means the component does not exist.</td>
</tr>
</tbody>
</table>
Showing Security Gateway Module Resource Information (asg resource)

**Description**  
Shows the Security Gateway Module (SGM) resource usage and thresholds for the entire 61000 Security Systems.

**Syntax**  
`asg resource [-b sgm]`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-b sgm</code></td>
<td>List of Security Gateway Modules. For example:</td>
</tr>
<tr>
<td></td>
<td>1_01</td>
</tr>
<tr>
<td></td>
<td>1_03-1_05</td>
</tr>
<tr>
<td></td>
<td>1_01,1_03-1_05</td>
</tr>
<tr>
<td></td>
<td><code>all</code></td>
</tr>
<tr>
<td></td>
<td><code>chassis1</code></td>
</tr>
<tr>
<td></td>
<td><code>chassis2</code></td>
</tr>
<tr>
<td></td>
<td><code>chassis_active</code></td>
</tr>
</tbody>
</table>

| `-h`       | Shows usage and exits |

**Example**  
`asg resource`
Output

gsm1030-ch01-01 > asg resource

<table>
<thead>
<tr>
<th>Resource Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Name</td>
</tr>
<tr>
<td>----------------</td>
</tr>
</tbody>
</table>

1. Chassis 1

<table>
<thead>
<tr>
<th>Resource</th>
<th>Location</th>
<th>Usage</th>
<th>Threshold</th>
<th>Total</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>SGM 1</td>
<td>88%</td>
<td>80%</td>
<td>1.0</td>
<td>G</td>
</tr>
<tr>
<td>Memory</td>
<td>SGM 2</td>
<td>96%</td>
<td>80%</td>
<td>915.6</td>
<td>M</td>
</tr>
<tr>
<td>Memory</td>
<td>SGM 3</td>
<td>96%</td>
<td>80%</td>
<td>915.6</td>
<td>M</td>
</tr>
<tr>
<td>HD: /</td>
<td>SGM 1</td>
<td>92%</td>
<td>80%</td>
<td>3.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /</td>
<td>SGM 2</td>
<td>80%</td>
<td>80%</td>
<td>3.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /</td>
<td>SGM 3</td>
<td>80%</td>
<td>80%</td>
<td>3.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /var/log</td>
<td>SGM 1</td>
<td>18%</td>
<td>80%</td>
<td>2.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /var/log</td>
<td>SGM 2</td>
<td>5%</td>
<td>80%</td>
<td>2.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /var/log</td>
<td>SGM 3</td>
<td>5%</td>
<td>80%</td>
<td>2.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /boot</td>
<td>SGM 1</td>
<td>14%</td>
<td>80%</td>
<td>151.3</td>
<td>M</td>
</tr>
<tr>
<td>HD: /boot</td>
<td>SGM 2</td>
<td>14%</td>
<td>80%</td>
<td>151.3</td>
<td>M</td>
</tr>
<tr>
<td>HD: /boot</td>
<td>SGM 3</td>
<td>14%</td>
<td>80%</td>
<td>151.3</td>
<td>M</td>
</tr>
</tbody>
</table>

2. Chassis 2

<table>
<thead>
<tr>
<th>Resource</th>
<th>Location</th>
<th>Usage</th>
<th>Threshold</th>
<th>Total</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>SGM 1</td>
<td>95%</td>
<td>80%</td>
<td>1.0</td>
<td>G</td>
</tr>
<tr>
<td>Memory</td>
<td>SGM 2</td>
<td>91%</td>
<td>80%</td>
<td>915.6</td>
<td>M</td>
</tr>
<tr>
<td>Memory</td>
<td>SGM 3</td>
<td>92%</td>
<td>80%</td>
<td>915.6</td>
<td>M</td>
</tr>
<tr>
<td>HD: /</td>
<td>SGM 1</td>
<td>80%</td>
<td>80%</td>
<td>3.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /</td>
<td>SGM 2</td>
<td>80%</td>
<td>80%</td>
<td>3.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /</td>
<td>SGM 3</td>
<td>80%</td>
<td>80%</td>
<td>3.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /var/log</td>
<td>SGM 1</td>
<td>20%</td>
<td>80%</td>
<td>2.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /var/log</td>
<td>SGM 2</td>
<td>5%</td>
<td>80%</td>
<td>2.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /var/log</td>
<td>SGM 3</td>
<td>5%</td>
<td>80%</td>
<td>2.1</td>
<td>G</td>
</tr>
<tr>
<td>HD: /boot</td>
<td>SGM 1</td>
<td>14%</td>
<td>80%</td>
<td>151.3</td>
<td>M</td>
</tr>
<tr>
<td>HD: /boot</td>
<td>SGM 2</td>
<td>14%</td>
<td>80%</td>
<td>151.3</td>
<td>M</td>
</tr>
<tr>
<td>HD: /boot</td>
<td>SGM 3</td>
<td>14%</td>
<td>80%</td>
<td>151.3</td>
<td>M</td>
</tr>
</tbody>
</table>

Comments

1. The **Resource** column identifies the resource. There are 4 kinds of resource:
   - Memory
   - HD – hard drive space (/)
   - HD: /var/log – space on hard drive committed to log files
   - HD: /boot - location of the kernel
2. The **Location** column identifies the SGM with the resource.
3. The **Usage** column shows in percentage terms how much of that resource has been used (hard drive or directory on hard drive) or is in use (memory).
4. The **Threshold** column is also expressed as a percentage. The threshold gives an indication of the health and functionality of the component. When the value of the resource is greater than the threshold, an alert is sent.
5. The **Total** column is the total absolute value in units
6. The **Units** column shows the measurement type, Megabytes (M) or Gigabytes (G).

For example, the first row shows that SGM1 on Chassis 1 has 11.6 Gigabyte of memory, 38% of which is used. An alert will be sent if the usage exceeds 80%.

Searching for a Connection (asg search)

**Description**

Use this command to search for a connection, and find out which SGM handles the connection (actively or as backup), and which chassis.
Configuring Alerts for SGM and Chassis Events (asg alert)

**Description**
Configure alerts for SGM and chassis events. Event types include hardware failure, recovery, and performance related events. General events can be monitored as well.

An alert is sent when an event occurs. For example, when an hardware resource value is greater than the threshold. The alert message includes chassis ID, SGM ID and/or unit ID, as applicable.

This is a menu-based tool.

**Syntax**
```
asg alert
```

---

**Example 1**
```
asg search <source IP> <destination IP>
```

**Output**
```
14.14.14.1, 38110, 24.24.24.1, 22, tcp -> [ch01b03 s, ch02b01 b, ch02b03 a]

Legend:
A = Active SGM
B = Backup SGM
```

**Comments**
Searching for connections from 14.14.1.1 to 24.24.24.1 shows one SSH connection:
```
```
This connection is handled by SGM 3 in chassis 1. The connection has a backup on SGM 1 and another backup in chassis 2 on SGM 3.

---

**Syntax**
asg search
asg search <src> <dst> <dport> <ipp> <sport>
asg search -v
asg search -help

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asg search</td>
<td>Run in interactive mode. In this mode you are asked to enter the 5 tuples of the connection parameters. Each parameter can be a wildcard. Press enter for wildcard.</td>
</tr>
<tr>
<td>asg search &lt;src&gt; &lt;dst&gt; &lt;dport&gt; &lt;ipp&gt; &lt;sport&gt;</td>
<td>Run in command line. Each parameter can be replaced by * for wildcard. If you specify only few parameters, the wildcard is used for the others. For example: asg search 192.0.2.44 * * * 4555 is translated as: &lt;192.0.2.44, 4555, any, any, any&gt;</td>
</tr>
<tr>
<td>-v</td>
<td>Verbose mode</td>
</tr>
<tr>
<td>-help</td>
<td>Display usage</td>
</tr>
</tbody>
</table>

---

```
Common System Commands Page 25
```
Output

(Main Menu)

Choose one of the following options:
-----------------------------
1) Full Configuration Wizard
2) Edit Configuration
4) Show Configuration
5) Run Test
  e) Exit

>
Option | Description repeat header rows
--- | ---
1. Full Configuration Wizard | 1. Choose an alert type (SMS, email, SNMP trap or SmartView Tracker log).

2. Configure the properties of each alert type:
   - SMS alert configuration:
     - Full URL that is used to send SMS by your SMS provider
     - HTTP proxy on given port (Optional) – should be configured if your gateway requires a proxy to reach the URL
     - SMS Rate Limit - Limit the number of SMSes sent per hour
     - SMS User Text - Custom prefix for the SMS messages
   - Email alert configuration:
     - SMTP Server IP/s - Configure one or more SMTP servers to which the email alerts will be sent
     - Email recipient address/es - Configure one or more addresses on each SMTP server to send the email alerts to
     - SMTP connectivity check - Configure whether you want the system to check connectivity to each defined SMTP server, and in case there is no connectivity to aggregated all the email alerts that are about to be sent and send them in an aggregated email once connectivity is restored
     - Sender Email address - Configured the sender address for the email alerts
     - Subject Text - Configure the text that will appear in the subject field of each email alert
     - Email body user text - Configure a custom prefix for the email alerts body messages
   - SNMP alert configuration:
     - SNMP Managers - Configure one or more SNMP managers which will receive the SNMP traps sent from the gateway. For each manager the following parameters need to be configured:
       - SNMP manager name - Configure a name for your SNMP manager (unique)
       - SNMP manager IP - Configure the manager IP address (trap receiver)
       - SNMP community string - Configure the community string for the SNMP manager
       - SNMP version - Configure the SNMP version to use (v2c/v3)
       - SNMP v3 user name - Used for SNMP v3 authentication. Needs to be configured in case the SNMP version chosen is v3
       - SNMP user text - Custom prefix for the SNMP trap messages
     - Note: It is recommended to refer to SNMP configuration section in this guide
   - Log (SmartView Tracker) alert configuration
     - These alerts don’t require specific configuration.
     - Log alerts are enabled by default.
     - Whenever log is issued, its message is also sent to syslog. In order to redirect alerts syslog messages to an external syslog server, refer to asg syslog section

3. Configure the events for which to send the alert:

System event types are:
------------------------------------
1 | SGM State
2 | Chassis State
3 | Port State
4 | Pingable Hosts State
Collecting System Diagnostics (asg diag)

Description
Use this command to display system diagnostics.
Upon execution, the command iterates over a predefined list of diagnostics tools. The output of each diagnostics tool is written to a log file. In addition a success/fail summary is presented at the end of the command execution.

Syntax: asg diag {list | verify | print | purge}

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>[[TestNum1],[TestNum2]...] Displays a list of verifiers</td>
</tr>
<tr>
<td>verify</td>
<td>[[TestNum1],[TestNum2]...] Iterates over a list of verifiers and prints a summary to the screen</td>
</tr>
<tr>
<td>print</td>
<td>[[TestNum1],[TestNum2]...] Iterates over a list of verifiers and prints verbose output to the screen</td>
</tr>
<tr>
<td>Purge</td>
<td>[number of logs to keep] Remove all the logs which were created by 'asg diag' except the newest &quot;number of logs to keep&quot;. Default number of logs to keep is 5.</td>
</tr>
</tbody>
</table>

Usage examples:
1. asg diag list – shows a list of all possible tests
2. asg diag list 1,2 – shows a list containing only tests with IDs 1 and 2.
3. asg diag verify – runs a full system test and prints the summary to the screen.
4. asg diag verify 1,2 – runs only tests 1 and 2 and prints the summary to the screen.
5. `asg diag print` – runs a full system test and prints verbose output as well as test's summary to the screen.

6. `asg diag print 1,2` – runs only tests 1 and 2 and prints verbose output as well as test's summary to the screen.

7. `asg diag purge 5` – removes all asg diag's log files except for the newest five.
Example 1  asg diag list

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Health</td>
<td>asg_stat –d</td>
</tr>
<tr>
<td>2</td>
<td>Hardware</td>
<td>asg_hw_monitor –q</td>
</tr>
<tr>
<td>3</td>
<td>Resources</td>
<td>asg_resource –q</td>
</tr>
<tr>
<td>4</td>
<td>Software Versions</td>
<td>asg_version verify –v</td>
</tr>
<tr>
<td>5</td>
<td>CPU Type</td>
<td>cpu_socket_verifier –v</td>
</tr>
<tr>
<td>6</td>
<td>Distribution Mode</td>
<td>dist_verify –d</td>
</tr>
<tr>
<td>7</td>
<td>Policy</td>
<td>asg_policy verify –a</td>
</tr>
<tr>
<td>8</td>
<td>Installation</td>
<td>installation_verify</td>
</tr>
<tr>
<td>9</td>
<td>Security Group</td>
<td>asg_security_group diag</td>
</tr>
<tr>
<td>10</td>
<td>Cores Distribution</td>
<td>cores_verifier</td>
</tr>
<tr>
<td>11</td>
<td>SPI Affinity</td>
<td>spi_affinity_verifier –v</td>
</tr>
<tr>
<td>12</td>
<td>Clock</td>
<td>clock_verifier –v</td>
</tr>
<tr>
<td>13</td>
<td>Mgmt Monitor</td>
<td>mgmt_monitor snmp_verify</td>
</tr>
<tr>
<td>14</td>
<td>MAC Setting</td>
<td>mac_verifier –v</td>
</tr>
<tr>
<td>15</td>
<td>Interfaces</td>
<td>interface_verifier –q</td>
</tr>
<tr>
<td>16</td>
<td>Bond</td>
<td>asg_bond_verifier –v</td>
</tr>
<tr>
<td>17</td>
<td>Bridge</td>
<td>asg_br_verifier –v</td>
</tr>
<tr>
<td>18</td>
<td>IPv4 Route</td>
<td>asg_route –q</td>
</tr>
<tr>
<td>19</td>
<td>IPv6 Route</td>
<td>asg_route_ipv6 –q</td>
</tr>
<tr>
<td>20</td>
<td>Dynamic Routing</td>
<td>asg_dr_verifier</td>
</tr>
<tr>
<td>21</td>
<td>Local ARP</td>
<td>asg_local_arp_verifier –v</td>
</tr>
<tr>
<td>22</td>
<td>Port Speed</td>
<td>asg_port_speed verify</td>
</tr>
<tr>
<td>23</td>
<td>Core Dumps</td>
<td>core_dump_verifier –v</td>
</tr>
<tr>
<td>24</td>
<td>Syslog</td>
<td>asg_syslog verify</td>
</tr>
</tbody>
</table>

The command “asg diag list” shows all available tests and their corresponding commands. For example, “System Health” test runs the command “asg stat –d” in order to receive its status. “asg diag list” can also be invoked by specifying test IDs, for example “asg diag list 1,6,7”. This command would only list the requested tests.
Example 2  

```bash
asg diag verify
```

<table>
<thead>
<tr>
<th>Tests Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>System Components</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>Policy and Configuration</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>Networking</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>Misc</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>Tests Summary</td>
</tr>
<tr>
<td>Passed: 18/24 tests</td>
</tr>
<tr>
<td>Run: &quot;asg diag list 1,2,3,4,5,20&quot; to view a complete list of failed tests</td>
</tr>
</tbody>
</table>

---
Running the command suggested by the diagnostic tool shows the list of commands that have failed.

```
asg diag list 1,2,3,4,5,20 prints the following:
```

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>System Health</td>
<td>asg stat -d</td>
</tr>
<tr>
<td>2</td>
<td>Hardware</td>
<td>asg hw_monitor -q</td>
</tr>
<tr>
<td>3</td>
<td>Resources</td>
<td>asg resource -q</td>
</tr>
<tr>
<td>4</td>
<td>Software Versions</td>
<td>asg_version verify -v</td>
</tr>
<tr>
<td>5</td>
<td>CPU Type</td>
<td>cpu_socket_verifier -v</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Networking</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
</tr>
</tbody>
</table>

In order to check the reason for “System Health” failure, “asg stat -d” or “asg diag print 1” commands should be invoked. This is a sample output of “asg stat -d”:

```
<table>
<thead>
<tr>
<th>System Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SGM ID</th>
<th>State</th>
<th>Process</th>
<th>Policy Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 (local)</td>
<td>UP</td>
<td>Enforcing Security</td>
<td>01Jul12 14:54</td>
</tr>
<tr>
<td>3</td>
<td>DOWN (Admin)</td>
<td>Inactive</td>
<td>NA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>SGMs</td>
</tr>
<tr>
<td>Ports</td>
</tr>
<tr>
<td>Standard</td>
</tr>
<tr>
<td>Other</td>
</tr>
<tr>
<td>Sensors</td>
</tr>
<tr>
<td>Fans</td>
</tr>
<tr>
<td>SSNs</td>
</tr>
<tr>
<td>CMMs</td>
</tr>
<tr>
<td>Power Supplies</td>
</tr>
<tr>
<td>Chassis Grade</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Synchronization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Within chassis:</td>
</tr>
<tr>
<td>Exception Rules:</td>
</tr>
<tr>
<td>Distribution</td>
</tr>
<tr>
<td>Control Blade:</td>
</tr>
</tbody>
</table>
```

The chassis grade is 118/124 because one of the SGMs is in DOWN (Admin) state. Bringing the SGM up will solve the problem. Alternatively, removing the SGM from the security group will suppress the alert.
Another debugging possibility is to open the output file in /var/log/. Every time “asg diag verify” or “asg diag print” are invoked, a log file is created which includes verbose output of each test the diagnostic tool ran.

Sample “CPU Type” test verbose output in the log file:

```
Non-compliant cpu models found:
------------------------------------
model name      : Intel(R) Xeon(R) CPU           E5530  @ 2.40GHz
Refer to /proc/cpuinfo for more information
```

This log file indicates that E5530 is not recognized by the “CPU Type” verifier as compliant with the current system. However, if this CPU type should be recognized as compliant, it can be configured inside asg_diag_config under $FWDIR/conf. In order to set another CPU type is compliant, a “Certified cpu=” line should be added where the RValue should indicate the CPU type.

After fixing the problems, it is possible to run a subset of the tests that failed, in order to verify that all the problems have actually been fixed. The method for invoking only a subset of the tests is show in example 3.

**Example 3**  
```
  asg diag verify 1,2,3,4,5,20
```

<table>
<thead>
<tr>
<th>Tests Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>Tests Summary</td>
</tr>
<tr>
<td>Passed: 6/6 tests</td>
</tr>
<tr>
<td>Output file: /var/log/verifier_sum.1-5.20.2012-07-02_12-23-53.txt</td>
</tr>
</tbody>
</table>

**Comments**
The command iterates over the list of verifiers.
Error Highlights

System health

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis &lt;X&gt; error</td>
<td>General error indicating that chassis X grade is not perfect.</td>
</tr>
</tbody>
</table>

Hardware

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Component&gt; is missing</td>
<td>The component is not found in the chassis.</td>
</tr>
<tr>
<td>&lt;Component&gt; is down</td>
<td>The component is found in the chassis but is inactive.</td>
</tr>
</tbody>
</table>

Resources

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Resource&gt; capacity</td>
<td>Indicates that the specified resource capacity is not as expected. Expected</td>
</tr>
<tr>
<td></td>
<td>capacity can be tuned.</td>
</tr>
<tr>
<td>&lt;Resource&gt; exceed threshold</td>
<td>The resource’s usage exceeds the configured watermark.</td>
</tr>
</tbody>
</table>

CPU type

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non compliant CPU type</td>
<td>At least one SGM CPU type is not configured in the list of compliant CPUs.</td>
</tr>
<tr>
<td></td>
<td>Compliant CPU types can be configured</td>
</tr>
</tbody>
</table>

Security group

<table>
<thead>
<tr>
<th>Error</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Source&gt; error</td>
<td>The information gathered from this source is different between the SGMs.</td>
</tr>
<tr>
<td>&lt;Sources&gt; differ</td>
<td>The information gathered from several sources is different.</td>
</tr>
</tbody>
</table>
Configuration file

asg diag includes the configuration file $FWDIR/conf/asg_diag_config to control acceptable values for several resources.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td>The valid RAM memory capacity in GB.</td>
</tr>
<tr>
<td>HD: /</td>
<td>The valid capacity in GB for “HD: /” partition.</td>
</tr>
<tr>
<td>HD:/var/log</td>
<td>The valid capacity in GB for “HD: /var/log” partition.</td>
</tr>
<tr>
<td>HD: /boot</td>
<td>The valid capacity in GB for “HD: /boot” partition.</td>
</tr>
<tr>
<td>Skew</td>
<td>The maximum clock difference between the SGMs and SSMs in seconds that is acceptable.</td>
</tr>
<tr>
<td>Certified cpu</td>
<td>Each line represents a compliant CPU type</td>
</tr>
</tbody>
</table>

Monitoring and Information Gathering

Networking

Showing Interface Status (asg if)

**Description**

Use this command to show information for interfaces on the appliance. Running the command shows:

- MAC hardware address, IP address, Info, State
- When invoked with the Performance mode parameter (-p) the command shows all the previous data, and also traffic statistics over the last 5 seconds in terms of:
  - Packets
  - Bytes per second
- When invoked with the Error mode parameter (-e) the command shows:
  - Errors
  - Drops
  - IP stack Drops
  - TX restart queue counter and interface state

**Syntax**

```
asg if [-i interface | -a] [-l] (normal mode)
asg if -p [-i interface | -a][-l] (performance mode)
asg if -e [-i interface | -a] (error mode)
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>Displays the interface status table</td>
</tr>
<tr>
<td>-a</td>
<td>Displays all interfaces</td>
</tr>
<tr>
<td>-i</td>
<td>Displays interface status for the specified interface</td>
</tr>
<tr>
<td>-l</td>
<td>Displays interface status of local SGM only. <strong>Note:</strong> -l can be used only when it’s the only flag chosen (aka: asg if -l)</td>
</tr>
<tr>
<td>-e</td>
<td>Display local SGM error mode</td>
</tr>
</tbody>
</table>

**Example**

`asg if`

**Output**

<table>
<thead>
<tr>
<th>Interface</th>
<th>MAC Address</th>
<th>Info</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sync</td>
<td>00:1c:7f:01:04:fe</td>
<td>Bond Master</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td>192.0.2.1/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eth1-01</td>
<td>00:1c:7f:81:01:fe</td>
<td>Ethernet</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td>5.5.6.10/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eth1-Mgt4</td>
<td>00:0c:29:03:aa:e2</td>
<td>Ethernet</td>
<td>up</td>
</tr>
<tr>
<td></td>
<td>172.16.6.151/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eth1-Sync</td>
<td>00:1c:7f:01:04:fe</td>
<td>Bond Slave</td>
<td>up</td>
</tr>
<tr>
<td>(Sync)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eth2-01</td>
<td>00:1c:7f:82:01:fe</td>
<td>Ethernet</td>
<td>ch1: up</td>
</tr>
<tr>
<td></td>
<td>15.15.15.10/24</td>
<td></td>
<td>ch2: down</td>
</tr>
<tr>
<td>eth2-Sync</td>
<td>00:1c:7f:01:04:fe</td>
<td>Bond Slave</td>
<td>up</td>
</tr>
<tr>
<td>(Sync)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Comments**

From the interface table, we learn that:

- Sync is a BOND-Master, with eth1-Sync and eth2-Sync as BOND-Slaves
- Interface eth2-01 is UP on Chassis1 and DOWN on Chassis2
- Interface eth2-Sync is a Bond Slave interface of Bond Master (Sync)

**Showing Traffic Information (asg_ifconfig)**

**Description**

The `asg_ifconfig` command collects statistics from all or a specified range of SGMs, processes them, and shows the combined output. The combined output shows traffic distribution between SGMs and their interfaces (calculated during a certain period).

The command has three modes:

- **Native**
  Default setting. When the analyze or banalyze option is not specified the command behaves similar to the native Linux `ifconfig` command, except that the output shows statistics for all interfaces on all SGMs and shows statistics for interfaces on the local SGM.

- **Analyze**
Shows accumulated traffic information and traffic distribution between SGMs.

- **Banalyze**
  Shows accumulated traffic information and traffic distribution between interfaces.

  **Note** - Analyze and Banalyze parameters cannot be used together.

**Syntax**

```
asg_ifconfig [-b SGMs][interface][analyze][-d][-v][-a]
asg_ifconfig [-b SGMs][interface][banalyze][-d][-v][-a]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>The name of the interface</td>
</tr>
</tbody>
</table>
| -b SGMs   | SGM values in one of these formats:  
  - 1_1_1_4, or 1_1-1_4, or 1_1_1_3-1_7, 1_10  
  - all (the default option)  
  - chassis1  
  - chassis2  
  - chassis_active |
| -d delay  | Delay between data samples - default: 5 seconds |
| -v        | Verbose mode: show detailed information for each interface |
| -a        | Show absolute values (default: rate values) |
| -h        | Show help information and exit |
| analyze   | Show accumulated traffic information.  
  Add [-v] [-a] [-d delay] parameters to show traffic distribution between SGMs. |
### Parameter | Meaning
--- | ---
banalyze | - Shows accumulated traffic information  
- Can be used with these parameters to sort the traffic distribution table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-rp</td>
<td>RX packets</td>
</tr>
<tr>
<td>-rb</td>
<td>RX bytes</td>
</tr>
<tr>
<td>-rd</td>
<td>RX dropped packets</td>
</tr>
<tr>
<td>-tp</td>
<td>TX packets</td>
</tr>
<tr>
<td>-tb</td>
<td>TX bytes</td>
</tr>
<tr>
<td>-td</td>
<td>TX dropped packets</td>
</tr>
</tbody>
</table>

For example if you sort according to the -rb option, then the higher values appear at the top of the RX bytes column in the traffic distribution table:

<table>
<thead>
<tr>
<th>SGM ID</th>
<th>RX packets</th>
<th>RX bytes</th>
<th>RX dropped</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_03</td>
<td>70%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1_02</td>
<td>20%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1_01</td>
<td>10%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The traffic distribution table shows as part of the command output, but unsorted by default.

**Native (asg_ifconfig)**

**Syntax**

```bash
asg_ifconfig [-b] [SGMs] [interface]
```

**Example**

```bash
asg_ifconfig -b chassis1 eth2-01
```
The output shows totals for traffic passed through interface `eth2-01` on each SGM of chassis1.

**Analyze (asg_ifconfig analyze)**

**Description**

By default, this command shows accumulated statistics (rates) for each interface. If the:

- (-a) option is specified, the totals for all statistics are displayed instead of the rates
- (-b) isn't specified, statistics are calculated on the active chassis only.

**Syntax**

- `asg_ifconfig [interface] [analyze [-d delay]]`
  
  Displays accumulated traffic information

- `asg_ifconfig [interface] [analyze [-v][-d delay][-a]]`
  
  Displays accumulated traffic information and traffic distribution between SGMs

**Example 1**

`asg_ifconfig analyze`
Example 2

```bash
asg_ifconfig eth1-01 analyze -v
```

Output

```
# asg_ifconfig eth1-01 analyze -v
Command is executed on SAME: chassis_active
Processing system statistics for 5 seconds...

1_01:
eth1-01  Link encap:Ethernet  HWaddr 00:3c:7f:81:01:0d
inet addr:10.33.86.1  Bcast:10.33.86.255  Mask:255.255.255.0  UP BROADCAST RUNNING MULTICAST MTU:1500  Metric:1
RX: packets:4 bytes:108 (365 bps) dropped:0
TX: packets:0 bytes:0 (0 bps) dropped:0

1_02:
eth1-01  Link encap:Ethernet  HWaddr 00:3c:7f:81:01:0d
inet addr:10.33.86.1  Bcast:10.33.86.255  Mask:255.255.255.0  UP BROADCAST RUNNING MULTICAST MTU:1500  Metric:1
RX: packets:0 bytes:0 (0 bps) dropped:0
TX: packets:0 bytes:0 (0 bps) dropped:0

1_03:
eth1-01  Link encap:Ethernet  HWaddr 00:3c:7f:81:01:0d
inet addr:10.33.86.1  Bcast:10.33.86.255  Mask:255.255.255.0  UP BROADCAST RUNNING MULTICAST MTU:1500  Metric:1
RX: packets:0 bytes:0 (0 bps) dropped:0
TX: packets:0 bytes:0 (0 bps) dropped:0

1_04:
eth1-01  Link encap:Ethernet  HWaddr 00:3c:7f:81:01:0d
inet addr:10.33.86.1  Bcast:10.33.86.255  Mask:255.255.255.0  UP BROADCAST RUNNING MULTICAST MTU:1500  Metric:1
RX: packets:0 bytes:0 (0 bps) dropped:0
TX: packets:0 bytes:0 (0 bps) dropped:0

**= Accumulative ==
eth1-01  Link encap:Ethernet  HWaddr 00:3c:7f:81:01:0d
inet addr:10.33.86.1  Bcast:10.33.86.255  Mask:255.255.255.0  UP BROADCAST RUNNING MULTICAST MTU:1500  Metric:1
RX: packets:4 bytes:108 (365 bps) dropped:0
TX: packets:4 bytes:108 (365 bps) dropped:0

**= Traffic Distribution ==

<table>
<thead>
<tr>
<th></th>
<th>RX packets</th>
<th>RX bytes</th>
<th>RX dropped</th>
<th>TX packets</th>
<th>TX bytes</th>
<th>TX dropped</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_01</td>
<td>100.0%</td>
<td>51.2%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>1_02</td>
<td>0.0%</td>
<td>16.3%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1_03</td>
<td>0.0%</td>
<td>16.3%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1_04</td>
<td>0.0%</td>
<td>16.3%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
```

Comments

Shows accumulated statistics (rates) for the specified interface (verbose mode). If the SGM option (-b) isn’t specified, these statistics are calculated on the active chassis only.

Example 3

```bash
asg_ifconfig eth1-01 analyze -v -a
```
Output

command is executed on SGMs: chassis_active

1_01:
eth1-01. Link encap:Ethernet  Hwaddr 00:11:7f:81:01:16
 inet6 addr: 21c:7fff:fe81:016 Scope:link
 UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
 RX: packets:6272 bytes:334393 (326.8 Kib)  dropped:0
 TX: packets:351 bytes:41550 (40.7 Kib)  dropped:0

1_02:
eth1-02. Link encap:Ethernet  Hwaddr 00:11:7f:81:01:16
 inet6 addr: 21c:7fff:fe81:016 Scope:link
 UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
 RX: packets:6305 bytes:336408 (328.9 Kib)  dropped:0
 TX: packets:334 bytes:41028 (40.1 Kib)  dropped:0

1_03:
eth1-01. Link encap:Ethernet  Hwaddr 00:11:7f:81:01:16
 inet6 addr: 21c:7fff:fe81:016 Scope:link
 UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
 RX: packets:5997 bytes:292191 (285.3 Kib)  dropped:0
 TX: packets:335 bytes:41084 (40.0 Kib)  dropped:0

1_04:
eth1-01. Link encap:Ethernet  Hwaddr 00:11:7f:81:01:16
 inet6 addr: 21c:7fff:fe81:016 Scope:link
 UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
 RX: packets:5831 bytes:300413 (295.2 Kib)  dropped:0
 TX: packets:805 bytes:61404 (60.0 Kib)  dropped:0

="v= Accumulative ="="
eth1-01. Link encap:Ethernet  Hwaddr 00:11:7f:81:01:16
 inet6 addr: 21c:7fff:fe81:016 Scope:link
 UP BROADCAST RUNNING SLAVE MULTICAST MTU:1500 Metric:1
 RX: packets:24403 bytes:1249400 (1.2 Mib)  dropped:0
 TX: packets:1323 bytes:135166 (130.8 Kib)  dropped:0

="v= Traffic distribution =="

<table>
<thead>
<tr>
<th>Blade ID</th>
<th>RX packets</th>
<th>RX bytes</th>
<th>RX dropped</th>
<th>TX packets</th>
<th>TX bytes</th>
<th>TX dropped</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_01</td>
<td>25.7%</td>
<td>26.0%</td>
<td>0.0%</td>
<td>19.2%</td>
<td>22.5%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1_02</td>
<td>23.8%</td>
<td>25.2%</td>
<td>0.0%</td>
<td>18.9%</td>
<td>22.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1_03</td>
<td>24.0%</td>
<td>25.4%</td>
<td>0.0%</td>
<td>18.4%</td>
<td>22.2%</td>
<td>0.0%</td>
</tr>
<tr>
<td>1_04</td>
<td>23.9%</td>
<td>24.5%</td>
<td>0.0%</td>
<td>44.1%</td>
<td>33.2%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Comments Shows accumulated statistics (absolute values) for the specified interface (verbose mode). If the SGM option (-b) isn't specified, these statistics are calculated on the active chassis only.

Banalyze (asg_ifconfig banalyze)

Description
By default this command shows the accumulated statistics (rates) for each SGM. If the:
- (-a) option is specified, the totals for all statistics are displayed instead of the rates
- (-b) isn't specified, statistics are calculated on the active chassis only.

Syntax

asg_ifconfig [interface] [banalyze]

Shows accumulated traffic information and traffic distribution between interfaces

asg_ifconfig [interface] [banalyze] [-v][-d delay][-a][-r][r][-d delay][-r][-td]

Shows accumulated traffic information and traffic distribution between interfaces and sorts the traffic distribution table according to the specified parameters.

Example 1

asg_ifconfig banalyze
Example 2

```
asg_ifconfig -b 1_01,1_02 eth1_mgmt1 banalyze -v
```

Output

```
Processing system statistics for 5 seconds...

1_01:
  RX: packets:501 bytes:363301 (354.8 Kbps) dropped:0
  TX: packets:136 bytes:163026 (159.2 Kbps) dropped:0

1_02:
  RX: packets:508 bytes:378863 (370.0 Kbps) dropped:0
  TX: packets:104 bytes:98314 (96.0 Kbps) dropped:0

1_03:
  RX: packets:506 bytes:379201 (370.3 Kbps) dropped:0
  TX: packets:91 bytes:78610 (76.8 Kbps) dropped:0

== All Blades ==
  RX: packets:1515 bytes:1121365 (1.1 Mbps) dropped:0
  TX: packets:331 bytes:339950 (332.0 Kbps) dropped:0
```

Comments

Shows detailed and accumulated statistics (rates) for the management interface of specified SGMs (1_01 and 1_02). If the SGM option (-b) isn’t specified, these statistics are calculated on the active chassis only.
**Example 3**

```
> asg_ifconfig -b 1_02-1_04 eth2-01 banalyze -v -a
```

**Output**

```
1_02:
eth2-01 Link encap:EthernetHWaddr 00:1c:7f:82:01:16
    inet6 addr: fe80::2a0:1aff:fe01:201/64 Scope:link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:5381 bytes:533854 (326.0 KByte) dropped:0
    TX packets:7 bytes:550 (350.0 b) dropped:0

== Accumulative ==
    RX packets:5381 bytes:533854 (326.0 KByte) dropped:0
    TX packets:7 bytes:550 (350.0 b) dropped:0

== Traffic Distribution ==

<table>
<thead>
<tr>
<th>Interface</th>
<th>RX packets</th>
<th>RX bytes</th>
<th>RX dropped</th>
<th>TX packets</th>
<th>TX bytes</th>
<th>TX dropped</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth2-01</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
```

```
1_03:
eth2-01 Link encap:EthernetHWaddr 00:1c:7f:82:01:16
    inet6 addr: fe80::2a0:1aff:fe01:201/64 Scope:link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:5326 bytes:330384 (322.6 KByte) dropped:0
    TX packets:7 bytes:550 (350.0 b) dropped:0

== Accumulative ==
    RX packets:5326 bytes:330384 (322.6 KByte) dropped:0
    TX packets:7 bytes:550 (350.0 b) dropped:0

== Traffic Distribution ==

<table>
<thead>
<tr>
<th>Interface</th>
<th>RX packets</th>
<th>RX bytes</th>
<th>RX dropped</th>
<th>TX packets</th>
<th>TX bytes</th>
<th>TX dropped</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth2-01</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
```

```
1_04:
eth2-01 Link encap:EthernetHWaddr 00:1c:7f:82:01:16
    inet6 addr: fe80::2a0:1aff:fe01:201/64 Scope:link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:5165 bytes:323756 (316.2 KByte) dropped:0
    TX packets:174 bytes:7564 (7.4 KByte) dropped:0

== Accumulative ==
    RX packets:5165 bytes:323756 (316.2 KByte) dropped:0
    TX packets:174 bytes:7564 (7.4 KByte) dropped:0

== Traffic Distribution ==

<table>
<thead>
<tr>
<th>Interface</th>
<th>RX packets</th>
<th>RX bytes</th>
<th>RX dropped</th>
<th>TX packets</th>
<th>TX bytes</th>
<th>TX dropped</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth2-01</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>
```

**Comment**

Shows detailed and accumulated statistics (absolute values) for the specified interface of specified SGMs (1_01-1_04). If the SGM option (-b) isn't specified, these statistics are calculated on the active chassis only.
**Internal interfaces**

To show traffic statistics for internal interfaces:

- Sync
- Sync1
- Sync2
- CIN

Use the `-v` (verbose) option while running `asg_ifconfig` in the analyze or banalyze mode.

**Showing the Routing Tables (asg_route)**

**Description**

Use this command to show the routing tables on all SGMs. This command shows routes unique to specified SGMs, routes configured on all SGMs, or source-based routes.

**Syntax**

```
asg_route [-b blade_string] [ipv6] [inactive] [filter]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-b blade_string</code></td>
<td>Specify SGM in one of these ways:</td>
</tr>
<tr>
<td></td>
<td>- 1_1,1_4, or 1_1-1_4, or 1_1,1_3-1_7, 1_10</td>
</tr>
<tr>
<td></td>
<td>- all (default)</td>
</tr>
<tr>
<td></td>
<td>- chassis1 (SGMs on Chassis1)</td>
</tr>
<tr>
<td></td>
<td>- chassis2 (SGMs on Chassis2)</td>
</tr>
<tr>
<td>ipv6</td>
<td>Shows IPv6 routes (IPv4 is the default)</td>
</tr>
<tr>
<td>inactive</td>
<td>Shows inactive routes</td>
</tr>
<tr>
<td></td>
<td>Only these filters can be used with the inactive parameter:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Filter</th>
<th>Shows inactive:</th>
</tr>
</thead>
<tbody>
<tr>
<td>aggregate</td>
<td>Aggregate routes</td>
</tr>
<tr>
<td>bgp</td>
<td>BGP routes</td>
</tr>
<tr>
<td>direct <code>&lt;address&gt;</code></td>
<td>Directly connected routes</td>
</tr>
<tr>
<td>ospf</td>
<td>Routes received via OSPF</td>
</tr>
<tr>
<td>static</td>
<td>Static routes</td>
</tr>
<tr>
<td>Filter</td>
<td>Shows active</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>aggregate</td>
<td>Aggregate routes</td>
</tr>
<tr>
<td>bgp</td>
<td>BGP routes</td>
</tr>
<tr>
<td>destination &lt;address&gt;</td>
<td>Route(s) to a specific destination</td>
</tr>
<tr>
<td>direct &lt;address&gt;</td>
<td>Directly connected routes</td>
</tr>
<tr>
<td>exact &lt;address&gt;</td>
<td>Specific route from a given address</td>
</tr>
<tr>
<td>less-specific &lt;address&gt;</td>
<td>Less specific routes from a given address</td>
</tr>
<tr>
<td>more-specific &lt;address&gt;</td>
<td>More specific routes from a given address</td>
</tr>
<tr>
<td>ospf</td>
<td>Routes received via OSPF</td>
</tr>
<tr>
<td>static</td>
<td>Static routes</td>
</tr>
<tr>
<td>sbr</td>
<td>Source-based routes</td>
</tr>
<tr>
<td>Summary</td>
<td>Summarizes the routing table</td>
</tr>
</tbody>
</table>

**Example**  
`asg_route -b 1_01,1_02`
Networking

Output

1. Fetching Routes info from SIDs:
   1.0.1.0.2

Status: DB Routes info is NOT identical on all SIDs
DB Routes info is NOT identical on all SIDs

Identical DB Routes: (6 records)
- 127.0.0.1/32 is directly connected, lo
- 161.0.0.0/24 is directly connected, eth1-CEN
- 172.15.6.0/24 is directly connected, eth1-Mgt4
- 192.0.2.0/24 is directly connected, Sync
- 209.7.122.0/24 via 172.15.6.4, eth1-Mgt4, cost 0, age (live)
- 0.0.0.0/0

Inconsistent DB Routes:
- 1.0.1.0.2
  - 15.15.15.0/24 is directly connected, eth1-01
  - 209.7.122.0/24 via 172.15.6.4, eth1-Mgt4, cost 0, age (live)

Types: C - Connected, S - Static, R - RIP, B - BGP,
O - OSPF Intarea, IA - InterArea, E - External, N - NSSA
A - Aggregate, K - Kernel  Command, H - Hidden, P - Suppressed
SBR - Source-Based Routes

Identical OS Routes: (14 records)
- 172.16.0.254 dev eth1-Mgt4 proto kernel scope link src 172.16.0.254
- 5.5.5.0/24 dev eth1-01 proto kernel scope link src 5.5.5.10
- 127.0.0.0/8 dev lo table 254 proto kernel scope link src 127.0.0.1
- 127.0.0.0/8 dev lo table 254 proto kernel scope link src 127.0.0.1
- 172.16.5.0 dev eth1-Mgt4 table 254 proto kernel scope link src 172.16.5.1
- 172.16.6.255 dev eth1-Mgt4 table 254 proto kernel scope link src 172.16.6.151
- 172.16.6.255 dev eth1-Mgt4 table 254 proto kernel scope link src 172.16.6.151
- 5.5.5.0/16 dev eth1-01 table 254 proto kernel scope link src 5.5.5.10
- 15.15.15.0/24 dev eth1-01 table 254 proto kernel scope link src 15.15.15.0
- 15.15.15.0/24 dev eth1-01 table 254 proto kernel scope link src 15.15.15.0
- 15.15.15.0/24 dev eth1-01 table 254 proto kernel scope link src 15.15.15.0
- 209.7.122.0/24 via 172.16.6.4 dev eth1-Mgt4 proto gated

Inconsistent OS Routes:
- 1.0.1.0.2
  - 15.15.15.0/24 dev eth1-01 proto kernel scope link src 15.15.15.0
  - 15.15.15.0/24 dev eth1-01 table 254 proto kernel scope link src 15.15.15.0
  - 15.15.15.0/24 dev eth1-01 table 254 proto kernel scope link src 15.15.15.0

Note: Output can be found under:
'/var/log/seq_route/Files/inconsistent.13245524.tar.gz'
Connecting to a specific SGM (blade command)

**Description:**
When connecting to the system you are communicating with one of the SGMs. To connect to another SGM use the command “blade” which can be executed in bash shell. The command will open an SSH connection to the desired SGM over the Sync interface.

**Syntax:**
blade <SGM>

**Example:**
blade 1_03
use “exit” to return to the previous SGM

**Input:**
SGM is the SGM ID. Should be in the format <SGM#>_<CHASSIS#> and in case only SGM# is specified then <CHASSIS#> gets the value of the current chassis. <SGM#> can be specified with or without the leading zero, i.e. 1_3 or 1_03

**Note:**
Multiple “blade” commands will open multiple SSH sessions.

VPN Packet Tracking

Use these commands to track IPSEC packet flow.

<table>
<thead>
<tr>
<th>To see</th>
<th>Run</th>
</tr>
</thead>
</table>
| Source and destination IP addresses | • g_tcpdump for ip proto 50 (For Site-to-Site VPN)  
  • g_tcpdump for UDP port 4500 (For SecureClient and Endpoint VPN clients) |
| Which SGM receives encrypted traffic | asg_dxl calc <src_ip> 0 <dst_ip> 0 50 2 |
| Which SGM encrypted packets are forwarded to | bcstats vpn -v |
| Which SGM holds the outbound SA | g_fw tab -t outbound_SPI -f |
  • Search for MSPI in the output. MSPI represents the Meta SA, and shows which SGM holds the outbound SA. For example:  
  • The output can include SA's with an MSPI of 0. These are dummy SAs and can be safely ignored. |
Showing SSM Traffic Statistics (asg_traffic_stats)

Description
Use this command to show traffic statistics, in terms of throughput (Bits per second) and Packet rate (packets per second), for SSM ports during a specified time period.

Packet rate statistics are divided to four categories:

- Unicast
- Multicast
- Broadcast
- Total packets per second.

Syntax
asg_traffic_stats <SSM ID|interface name> [delay, default: 5]

<table>
<thead>
<tr>
<th>Option</th>
<th>Parameter and Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSM ID</td>
<td>SSM ID: 1 or 2</td>
</tr>
<tr>
<td></td>
<td>Shows the total traffic statistics for a specified SSM</td>
</tr>
<tr>
<td>Interface name</td>
<td>The interface name: eth1-04 or eth1-Sync</td>
</tr>
<tr>
<td></td>
<td>Shows the total traffic statistics for a specified SSM</td>
</tr>
<tr>
<td>delay</td>
<td>Time in seconds (optional, default equals 5). Traffic statistics are divided by the delay interval to show the average per second.</td>
</tr>
</tbody>
</table>

Example1

```bash
asg_traffic_stats eth1-04
```

Output

```
eth1-04 statistics
---------------------
Incoming traffic:
Throughput: 273.8 kbps
Packet rate: [total: 1009 pps], [unicast: 328 pps], [multicast: 347 pps], [broadcast: 224 pps]
Outgoing traffic:
Throughput: 261.4 kbps
Packet rate: [total: 952 pps], [unicast: 272 pps], [multicast: 341 pps], [broadcast: 339 pps]
```

Comments
Shows traffic passing through eth1-04.

Example2

```bash
asg_traffic_stats 1
```

Output

```
Summary on SSM1
---------------------
Incoming traffic:
Throughput: 273.8 kbps
Packet rate: [total: 1009 pps], [unicast: 328 pps], [multicast: 347 pps], [broadcast: 224 pps]
Outgoing traffic:
Throughput: 261.4 kbps
Packet rate: [total: 952 pps], [unicast: 272 pps], [multicast: 341 pps], [broadcast: 339 pps]
```

Comments
Shows traffic passing through SSM1.
Showing SGM Information (asg_blade_stats)

**Description:**
Use this command to display various packets forwarding statistics in the system.

**Syntax:**
```
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>corr [-a]</td>
<td>Display correction layer statistics for each SGM.</td>
</tr>
<tr>
<td>corr -p [-v] [-a]</td>
<td>Display correction layer statistics per service (for predefined services) for each SGM.</td>
</tr>
<tr>
<td>corr -reset</td>
<td>Reset correction layer statistics</td>
</tr>
<tr>
<td>corr_online</td>
<td>Display current correction layer information for each SGM</td>
</tr>
<tr>
<td>iterator</td>
<td>Display information about the last iterator process</td>
</tr>
<tr>
<td>smo</td>
<td>Display statistics on SMO task and logs for each SGM</td>
</tr>
<tr>
<td>vpn [-v]</td>
<td>Display statistics on VPN forwarded packets</td>
</tr>
<tr>
<td>6in4 [-v]</td>
<td>Display statistics on 6in4 forwarded packets</td>
</tr>
<tr>
<td>gre [-v]</td>
<td>Display statistics on GRE forwarded packets</td>
</tr>
<tr>
<td>icmp_error [-v]</td>
<td>Display statistics on ICMP ERROR forwarded packets</td>
</tr>
<tr>
<td>all</td>
<td>Display all correction layer statistics mentioned above</td>
</tr>
<tr>
<td>help</td>
<td>Display help information</td>
</tr>
</tbody>
</table>

**Multi-blade capture**
(tcpdump –mcap)

**Description**
Two new command line options were added to tcpdump:
- `tcpdump –mcap` - supports capturing of packets from multiple blades and saving them into a single capture file.
- `tcpdump –view` – reads packets from the file saved by `tcpdump –mcap` and displays the id of the blade on which the packet was captured

1. **tcpdump –mcap**

**Syntax**
```
tcpdump [-b blade string] -mcap -w <capture file fullpath> [tcpdump cmdline]
```

Note: in order to stop the capture process and to merge the capture from all SGMs the “stop” command need to be written.

**Output**
The output file specified in the `-w` command line switch.
In addition to the merged capture file, per blade capture files are created in the same directory, suffixed by their blade id.
gcpmodule-ch01-01 > tcpdump -mcap -w /tmp/capture -nnni eth1-Mgmt4
Capturing packets...

Write "stop" and press enter to stop the packets capture process.

1_01: tcpdump: listening on eth1-Mgmt4, link-type EN10MB (Ethernet), capture size 96 bytes
stop
Received user request to stop the packets capture process.

Copying captured packets from all blades...
Merging captured packets from blades to /tmp/capture...
Done.
gcpmodule-ch01-01 > shell

[Expert@cpmodule-ch01-01]# ls -l /tmp/capture*
-rw-r----- 1 admin root 46285 Nov 27 14:12 /tmp/capture
-rw-r----- 1 admin root 9500 Nov 27 14:12 /tmp/capture_1_1
-rw-r----- 1 admin root 6996 Nov 27 14:12 /tmp/capture_1_2
-rw-r----- 1 admin root 7541 Nov 27 14:12 /tmp/capture_1_3
-rw-r----- 1 admin root 7541 Nov 27 14:12 /tmp/capture_2_1
-rw-r----- 1 admin root 7286 Nov 27 14:12 /tmp/capture_2_2
-rw-r----- 1 admin root 7541 Nov 27 14:12 /tmp/capture_2_3

[Expert@cpmodule-ch01-01]#

Examples
- Simple usage: tcpdump -mcap -w /tmp/capture
- On selected blades: tcpdump -b 1_1,1_3,2_1 -mcap -w /tmp/capture -nnni eth1-Mgmt4
- On specific interface: tcpdump -mcap -w /tmp/capture -nnni eth1-Mgmt4
- With filter: tcpdump -mcap -nnni eth1-Mgmt4 -w /tmp/capture proto http

2. tcpdump -view

Syntax
 tcpdump -view -r <capture file fullpath> [tcpdump cmdline]

Output
Regular tcpdump output, prefixed by blade ID of the processing blade

Example
tcpdump -view -r /tmp/capture port http

- Comments
  1. Run tcpdump -mcap -w /tmp/capture and wait few seconds.
     Write stop and press Enter.
     Check the existence of file /tmp/capture*.
  2. Run tcpdump -view -r /tmp/capture.cap to display the captured packets.
     The packets should be prefixed with the blade id of the blade on which the packet was captured.

Traceroute (asg_tracert)

Description:
Native tracert tool that runs locally from blade's shell (e.g., tracert <IP>) does not work properly on 61000.
The reason is that tracert is probing the requests in high rate and due to stickiness mechanism in the firewall
these packets are being dropped. Thus, asg_tracert replaces tracert and it limits the probing rate by pausing
0.5 seconds between probes. Actually asg_tracert runs tracert with “–z 500” option by force to slow down tracert probing. Note that asg_tracert can be used also with other tracert options.

Syntax:

```plaintext
asg_tracert <IP Address> <tracert options>
```

Example:

```plaintext
asg_tracert 100.100.100.99
asg_tracert 100.100.100.99 --udp
```

Output:

```plaintext
gesx-ch01-01 > asg_tracert 100.100.100.99
traceroute to 100.100.100.99 (100.100.100.99), 30 hops max, 40 byte packets
  1  (20.20.20.20)  0.722 ms  0.286 ms  0.231 ms
  2  (100.100.100.99)  1.441 ms  0.428 ms  0.395 ms
gesx-ch01-01 >

gesx-ch01-01 > asg_tracert 100.100.100.99 --udp
traceroute to 100.100.100.99 (100.100.100.99), 30 hops max, 40 byte packets
  1  (20.20.20.20)  0.998 ms  0.677 ms  0.554 ms
  2  (100.100.100.99)  1.679 ms  1.042 ms  1.134 ms
gesx-ch01-01 >
```

Explanation

“asg_tracert 100.100.100.99”

runs the following command:

```
traceret –z 500 100.100.100.99
```

“asg_tracert 100.100.100.99 --udp”

runs the following command:

```
traceret –z 500 100.100.100.99 --udp
```

Hardware

Collecting System Serial Numbers

Description:

The following two commands are designed to extract serial numbers from hardware components of the 61000 Security System:

1. asg_sgm_serial – extract SGM serial numbers
2. asg_serial_info – extract CMM, SSM and Chassis serial numbers
Note: these commands are also part of the asg_info script which collects configuration and logging files on the system. Serial information can be found under gasginfo output compressed file. Both commands can only be executed from Expert shell.

**asg_sgm_serial**
This command extracts serial numbers from UP SGMs, which belong to the security group. In order to apply the command on all SGMs in the security group, use [-a] parameter.

**Example:**
```
> asg_sgm_serial
1_01:
  Board Serial : AKO0769153
1_02:
  Board Serial : AKO0585533
2_01:
  Board Serial : AKO0462069
2_02:
  Board Serial : AKO0447878
```

**asg_serial_info**
This command extracts serial numbers from CMMs, SSMs and chassis. In case of dual chassis system, the information will be extracted from both units.

**Example:**
```
> asg_serial_info
chassis 1 CMM1 serial: 1163978/005
chassis 1 CMM2 serial: 1157482/001
chassis 1 SSM1 serial: 0011140011
chassis 1 SSM2 serial: 0011140012
chassis 1 serial: 1159584/016
chassis 2 CMM1 serial: 1163090/041
chassis 2 CMM2 serial: 1155519/014
chassis 2 SSM1 serial: 0311310621
chassis 2 SSM2 serial: 0311310626
chassis 2 serial: 0831232/001
```

**Note:** To extract CMM, SSM and chassis serial numbers one of the SGMs on each chassis must be up and running (i.e. if no SGM is found on chassis#2, the serial numbers of the components, associated with this chassis, will neither be extracted nor displayed).
Chapter 2

Security

Showing the Number of Firewall and SecureXL Connections (asg_conns)

**Description**  
Use this command to show the number of firewall and SecureXL connections on each SGM.

**Syntax**  
```
asg_conns [-b <SGM>]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-b &lt;SGMs&gt;</td>
<td>The ID of the SGM. Use a comma to separate specified SGMs, for example: <code>asg_conns 1_1, 1_3</code>. When this parameter is not specified, only connections on the active chassis are shown.</td>
</tr>
</tbody>
</table>

**Example**  
asg_conns

```
1_01:
 #VALS 285 #PEAK 1004 #SLINKS 389
1_02:
 #VALS 53 #PEAK 227 #SLINKS 57
1_03:
 #VALS 53 #PEAK 230 #SLINKS 57
1_04:
 #VALS 281 #PEAK 861 #SLINKS 285
1_05:
 #VALS 54 #PEAK 198 #SLINKS 56
1_06:
 #VALS 53 #PEAK 147 #SLINKS 57
```

**Output**

```
Total (fwl connections table): 879 connections
1_01:
 There are 180666 connections in securexl connections table
1_02:
 There are 165256 connections in securexl connections table
1_03:
 There are 169212 connections in securexl connections table
1_04:
 There are 146598 connections in securexl connections table
1_05:
 There are 162450 connections in securexl connections table
1_06:
 There are 124090 connections in securexl connections table
Total (securexl connections table): 952272 connections
```
IPS Packet Capture (asg_get_capture)

Description

IPS packet captures are saved locally on each SGM. IPS packets which were saved on SGMs that are not the SMO, cannot be obtained via SmartView Tracker. Use asg_get_capture command to obtain IPS packet captures from the SGMs via shell.

Usage

Run from expert shell:

```
asg_get_capture [-b <SGMs>] [-u <Packet Capture UID>] [-p <Number of latest captures to keep>]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-b &lt;SGMs&gt;</td>
<td>List of Security Gateway Modules (default: all)</td>
</tr>
<tr>
<td>-u &lt;Packet Capture UID&gt;</td>
<td>Specify the Packet Capture UID you want to obtain The Packet Capture UID can be retrieved from the information field on the SmartView Tracker log record.</td>
</tr>
<tr>
<td>-p &lt;Number of latest captures to keep&gt;</td>
<td>Specify the number of latest obtained captures to keep while purging old obtained captures from the SGMs.</td>
</tr>
</tbody>
</table>
Example 1

```
$ asg_get_capture -u {0x50aa2392,0x2,0x1a0200c0,0x6ffff}
```

**SmartView Tracker log record:**

![SmartView Tracker log record](image)

**Output:**

```
# asg_get_capture -u {0x50aa2392,0x2,0x1a0200c0,0x6ffff}
Pkt captures are located at:
/var/log/cpmodule___0x50aa2392_0x2_0x1a0200c0_0x6ffff_.cap
```

**Comments:**

The above example obtains IPS packet capture based on SmartView Tracker log record. The capture file is in snoop format and can be opened via Wireshark.
Example 2

capture

Output

# asg_get_capture -p 0
Purging /var/log/cpmodule___0x50aa2392_0x2_0x1a0200c0_0x6ffff_.cap

Comments

The above example purges all obtained captures from all SGMs.

Packet drop monitoring (asg_drop_monitor)

Description

Real time drop monitoring across all SGMs.
Drop statistics are taken from several modules: NIC, OS, CoreXL, PSL, Ppak

Syntax

asg_drop_monitor [-r]

Parameters

reset shown statistics to 0

Example

capture

Output

NICs drops (Rx):
0
IP Stack qdisc drops (Tx):
0
CoreXL queue drops (F2F):
0
CoreXL queue drops (PXL F2P)
0
PSL drops (total):
0
PSL drops (udp):
0
PSL rejects:
0
Ppak drops:
Displaying aggregated data from blades: all
Reason Value Reason Value
-------------------------------------------------------------------------------
genral reason 0 PXL decision 0
fragment error 0 hl - spoof viol 0
F2F not allowed 0 hl - TCP viol 0
corrupted packet 0 hl - new conn 0
clr pkt on vpn 0 partial conn 0
encrypt failed 0 drop template 0
decrypt failed 0 outb - no conn 9
interface down 0 cluster error 0
XMT error 0 template quota 0
anti spoofing 0 Attack mitigation 0
local spoofing 0 sanity error 0
monitored spoofed 0 Conns limit. Exceed 0
Conns limit. Add fail 0

Log Utility (asg log)

Description
Use the asg log utility to show different types of logs, sorted by time and date, collected from the various SGMs.

**Syntax**  
```bash
asg log [-b <SGMs>] <log_name> [-tail [number]] [-f filter]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-b &lt;SGMs&gt;</code></td>
<td>A list of SGMs to show logs for. List the SGMs in one of these formats:</td>
</tr>
</tbody>
</table>
|               | • 1_1,1_4  
|               | The first and fourth SGMs on chassis 1  
|               | • 1_1-1_4  
|               | SGMs 1-4 on chassis 1  
|               | • 1_1,1_3-1_7,2_08  
|               | First SGM on chassis1, SGMs 3-7 on chassis 1, SGM 8 on chassis 2.  
|               | • all  
|               | All SGMs  
|               | • chassis1  
|               | All SGMs on chassis 1  
|               | • chassis2  
|               | All SGM on chassis 2  
|               | • chassis_active  
|               | All SGMs on the active chassis |
| `<log_name>`  | Enter one of these log types: |
|               | • audit  
|               | Shows the audit logs in /var/log, for example:  
|               | /var/log/asgaudit.log.1  
|               | • smd  
|               | Shows the System Monitor Daemon logs in /var/log, for example:  
|               | /var/log/sdm.log.2  
|               | • ports  
|               | Shows the ports logs in /var/log, for example:  
|               | /var/log/ports  
| `-tail [number]` | Shows the last lines of the log file for each SGM. When no number is specified, the last 10 lines of the log are shown. A parameter such "-tail 3" limits the output to the last three lines of the log file. |
| `-f filter`   | Word or phrase to filter by. For example: `-f debug` |
Example 1:

```bash
# asg log audit
Aug 01 08:53:43 1_02 WARNING: sgm_admin down on SGM: 1_02, User: susan, Reason: Maintenance
Aug 02 08:54:21 1_01 WARNING: Reboot on blades: 1_01, User: susan, Reason: Maintenance
Aug 04 08:55:33 2_01 WARNING: sgm_admin up on SGMs: 1_02, User: susan, Reason: Maintenance
Aug 06 11:48:30 2_01 CRITICAL: Sync turn off between chassises on blades: all, User: ms, Reason: Maintenance
Aug 08 11:49:17 2_01 CRITICAL: Sync turn off within chassis on blades: all, User: Tom, Reason: testing sync
Aug 08 11:49:43 2_01 CRITICAL: Sync turn on within chassis on blades: all, User: Peter, Reason: Maintenance
Aug 09 12:38:24 2_01 CRITICAL: Reboot on Blades: all, User: ms, Reason: Maintenance
```

Example 2:

```bash
# asg log smd -tail 3
Aug 03 09:07:03 1_03 check_dxl_consistency_with_pf OK
Aug 03 09:07:03 1_03 check_dxl_consistency_with_pf OK
Aug 03 09:10:03 1_03 check_dxl_consistency_with_pf OK
Aug 03 09:11:03 1_03 check_dxl_consistency_with_pf OK
Aug 03 09:11:03 1_03 check_dxl_consistency_with_pf OK
Aug 03 09:14:05 1_03 check_dxl_consistency_with_pf OK
Aug 03 09:14:08 1_03 check_date_time_configuration OK
Aug 03 09:14:09 1_03 check_dxl_configuration_consistency OK
Aug 03 09:14:09 1_03 check_dxl_configuration_consistency OK
Aug 03 09:14:09 1_03 check_dxl_configuration_consistency OK
Aug 03 09:14:09 2_01 check_dxl_configuration_consistency OK
Aug 03 09:14:09 2_01 check_dxl_configuration_consistency OK
Aug 03 09:15:00 2_01 check_dxl_configuration_consistency OK
Aug 03 09:15:00 2_01 check_dxl_configuration_consistency OK
Aug 03 09:15:00 2_01 check_dxl_configuration_consistency OK
Aug 03 09:15:00 2_01 check_dxl_configuration_consistency OK
Aug 03 09:15:37 2_03 check_date_time_configuration OK
Aug 03 09:15:38 2_03 check_dxl_configuration_consistency OK
Aug 03 09:15:38 2_03 check_dxl_configuration_consistency OK
```

Example 3:

```bash
# asg log ports -f routed
Apr 15 13:31:28 1_01 cpmodule-ch01-01 routed[6396]: CHANGE 3.3.3.0/24 gw (null) eth1-01 <Inactive|Active|Retain>
Apr 15 13:31:28 1_01 cpmodule-ch01-01 routed[6396]: CHANGE 4.4.4.0/24 gw (null) eth1-02 <Inactive|Active|Retain>
Apr 15 13:31:28 1_01 cpmodule-ch01-01 routed[6396]: CHANGE 5.5.5.0/24 gw (null) eth2-01 <Inactive|Active|Retain>
Apr 15 13:31:28 1_01 cpmodule-ch01-01 routed[6396]: CHANGE 70.0.0.0/24 gw (null) eth1-CIN <Inactive|Active|Retain>
Apr 15 13:31:28 1_01 cpmodule-ch01-01 routed[6396]: CHANGE 127.0.0.1/32 gw (null) lo <NoAdvise|Active|Retain>
```

Official 61000 Security System Versions

For the list of official 61000 version names, see the Check Point Data Center Security Appliances R75.40VS for 61000 home page (http://supportcontent.checkpoint.com/solutions?id=sk65305)

To verify which version is installed on 61000 Gateway use gclish ‘ver’ command.
Showing the Auditlog (asg_auditlog)

Description

Use this command to see the contents of the auditlog.

- Auditlog records changes made to the SGM database in memory or the SGM database on the hard disk.
- Entries to these databases are added or deleted using the set, add, or delete commands embedded in the gclish shell.
- When the SGM database changes, the action (set, delete, or add) is recorded in a dedicated auditlog file.
- The auditlog file is in the /tmp directory on each SGM.
- When the asg_auditlog is run, the dedicated auditlog files are collected from the specified SGMs (or all SGMs by default) and merged into one output file.
- If two changes (activities) are made on different SGM databases in a period of n seconds they are considered parallel and shown as one activity.
- The output file groups together actions that were done on different SGMs at the same time.
- The auditlog distinguishes between Permanent (p) and Transient (t) actions.
- The auditlog indicates if those actions added (+) or removed (-) entries from the SGM database.

<table>
<thead>
<tr>
<th>Action</th>
<th>Action Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permanent</td>
<td>p +</td>
<td>Action was followed by SAVE CONFIG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Entry added to the SGM database on the hard disk</td>
</tr>
<tr>
<td></td>
<td>p -</td>
<td>Action was followed by SAVE CONFIG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Entry deleted from the SGM database on the hard disk</td>
</tr>
<tr>
<td>Transient</td>
<td>t +</td>
<td>Action was not followed by SAVE CONFIG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Entry added to the SGM database in memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change does not survive reboot</td>
</tr>
<tr>
<td></td>
<td>t -</td>
<td>Action was not followed by SAVE CONFIG</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Entry deleted from the SGM database in memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change does not survive reboot</td>
</tr>
</tbody>
</table>

Example

| Nov 22 08:41:30 | admin localhost | t + | derived:machine:hostname cpmodule-0w [1 Blades: 1_01] |
| Nov 22 08:41:30 | admin localhost | t - | hosts:4:ge-31ida7:address [1 Blades: 1_01] |
| Nov 22 08:41:31 | admin localhost | p + | machine:hostname cpmodule-chu1-01 [1 Blades: 1_01] |
| Nov 22 08:41:31 | admin localhost | p - | hosts:4:ge-31ida7:1 [1 Blades: 1_01] |

Syntax:  
asg_auditlog [-b blades] [-d <n>] [-t <n>] [filter <filter>]
### Parameter | Meaning
--- | ---
-b <blades> | Specifies SGMs
-d <n> | Delta in seconds for the merging process.
- If two changes (activities) are made on different SGM databases within a period of n seconds they considered parallel and shown as one activity
- 5 seconds is the default
-tail <n> | Number of lines taken from the end of each SGM auditlog file, for example: "-tail 3" takes the last three lines. Default is 10
-filter <filter> | Word or phrase used to filter the auditlog

#### Example 1

```bash
> asg_audilog
```

<table>
<thead>
<tr>
<th>Time</th>
<th>User</th>
<th>Host</th>
<th>Command</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul 11 12:09:06</td>
<td>admin localhost</td>
<td>t</td>
<td>-chassis:high-availability: factors: sensor: fan 1</td>
<td>[9 blades: 1_01,1_02,1_03,1_04,1_05,2_02,2_03,2_04,2_05] 1</td>
</tr>
<tr>
<td>Jul 11 12:09:10</td>
<td>admin localhost</td>
<td>p</td>
<td>-chassis:high-availability: factors: sensor: fan 11</td>
<td>[9 blades: 1_01,1_02,1_03,1_04,1_05,2_02,2_03,2_04,2_05] 2</td>
</tr>
<tr>
<td>Jul 11 12:09:10</td>
<td>admin localhost</td>
<td>p</td>
<td>-chassis:high-availability: factors: sensor: fan 1</td>
<td>[9 blades: 1_01,1_02,1_03,1_04,1_05,2_02,2_03,2_04,2_05] 3</td>
</tr>
</tbody>
</table>

**Comments**

- The administrator set the fan factor to 1 and that the change was transient (1), to memory only
- Then the administrator made the change permanent by doing a SAVE CONFIG action that:
  - Deleted the old fan factor of 11 from the SGM database on hard disk. (2)
  - Added the new fan factor of 1 to the database (3)
- All the actions happened to the same nine SGMs

#### Example 2

```bash
> asg_audilog -b 1_3-1_4 -d 50 -f cpu_load
```

<table>
<thead>
<tr>
<th>Time</th>
<th>User</th>
<th>Host</th>
<th>Command</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jul 10 10:21:17</td>
<td>admin localhost</td>
<td>t</td>
<td>-chassis: alert_threshold: cpu_load_threshold, parc_low_ratio 80</td>
<td>[2 blades: 1_03,1_04]</td>
</tr>
<tr>
<td>Jul 10 10:21:17</td>
<td>admin localhost</td>
<td>t</td>
<td>-chassis: alert_threshold: cpu_load_threshold, parc_high 75</td>
<td>[2 blades: 1_03,1_04]</td>
</tr>
<tr>
<td>Jul 10 10:21:18</td>
<td>admin localhost</td>
<td>p</td>
<td>-chassis: alert_threshold: cpu_load_threshold, parc_high 75</td>
<td>[2 blades: 1_03,1_04]</td>
</tr>
<tr>
<td>Jul 10 10:21:18</td>
<td>admin localhost</td>
<td>p</td>
<td>-chassis: alert_threshold: cpu_load_threshold, parc_low_ratio 80</td>
<td>[2 blades: 1_03,1_04]</td>
</tr>
</tbody>
</table>

**Comments**

- Auditlog file was collected from SGMs 1_03 and 1_04
- Events that occurred in 50 seconds of each other are considered parallel: they occurred at the same time.
- Only records with the cpu_load phrase are shown
Showing the firewall Database Configuration (asg_config)

**Description**
- Use this command to show the configuration of the firewall database, or save the configuration to a file.
- The output shows the configuration for all SGMs.
- Use this command to:
  - Replicate a firewall configuration between systems. For example if you deploy a new 61000 Security System, you can quickly configure the new system by copying the saved configuration from a system already deployed.
  - Quickly configure a system that has been reverted to factory defaults. Before reverting to the factory default image, save the existing configuration then use it to override the factory settings.

**Syntax**
```
asg_config  show|save [-t] <path> <filename>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>show</td>
<td>Shows the existing database configuration</td>
</tr>
<tr>
<td>save [-t] &lt;filename&gt;</td>
<td>• Saves the configuration to a file</td>
</tr>
<tr>
<td></td>
<td>• Use -t to include a timestamp</td>
</tr>
<tr>
<td></td>
<td>For example: asg_config save -t myconfig</td>
</tr>
<tr>
<td></td>
<td>Note: If you do not include a path, the file is saved to: /home/admin</td>
</tr>
</tbody>
</table>

**Example**
```
asg_config show
> asg_config show
# Configuration of 61K-ch01-01
# Language version: 10.0v1
# Exported by admin on Mon Jul 11 05:56:58 2011
set interface eth1-01 state on
set interface eth1-01 ipv4-address 11.11.11.10 mask-length 24
set interface eth1-02 state on
set interface eth1-02 ipv4-address 2.2.2.10 mask-length 24
set static-route default nexthop gateway address 192.168.18.1 on
```
Showing Software and Firmware versions (asg_version)

Description
Use this command to:

- Retrieve system configuration
- Retrieve software versions:
  - Check Point software (firewall and Performance Pack versions)
  - Firmware versions for SGMs, SSMs, and CMMs
- Make sure system hardware components are running approved software and firmware versions

Syntax
asg_version [-b <blades>] [verify -v | -h]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>[-b &lt;blades&gt;]</td>
<td>Specifies SGMs</td>
</tr>
<tr>
<td></td>
<td>Use a comma to separate specified SGMs, for example: asg_version -b 1_01</td>
</tr>
<tr>
<td>verify [-v]</td>
<td>Makes sure system hardware components are running approved software and firmware versions</td>
</tr>
<tr>
<td></td>
<td>Use -v for verbose mode</td>
</tr>
<tr>
<td></td>
<td>Verbose mode also shows hardware components running approved software/firmware versions</td>
</tr>
<tr>
<td>-h</td>
<td>Shows usage options</td>
</tr>
</tbody>
</table>

Example 1
asg_version

Output
The first part of the output shows software and firmware versions installed on SSMs and CMMs.

```
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| Hardware Versions                                                                                                                   |
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| Component       | Type        | Configuration | Firmware |
+--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| Chassis 1       |             |               |          |
| SSM1            | SSM160      | N/A           | 2.4.811  |
| SSM2            | SSM160      | N/A           | 2.4.811  |
| CMM             | N/A         | N/A           | 2.74     |
| Chassis 2       |             |               |          |
| SSM1            | SSM160      | N/A           | 2.4.811  |
| SSM2            | SSM160      | N/A           | 2.4.811  |
| CMM             | N/A         | N/A           | 2.74     |
```

The second part shows:

- software versions installed on SGMs
- Internal firmware versions, such as BIOS.
- CPU related information, such as frequency and CoreXL allocation
blades
-------
OS version
--------
-** 4 blades: 1_02 1_03 2_02 2_04 -*-
OS build 67. OS kernel version 2.6.18-92cpx86_64. OS edition 64-bit

FireWall-1 version
-----------------
-** 4 blades: 1_02 1_03 2_02 2_04 -*-
This is Check Point VPN-1(TM) & FireWall-1(R) 61000 R75.051 - Build 107
kernel: 61000 R75.051 - Build 107

Performance Pack version
--------------------------
-** 4 blades: 1_02 1_03 2_02 2_04 -*-
This is Check Point Performance Pack version: 61000 R75.051 - Build 041
Kernel version: 61000 R75.051 - Build 041

Hardware
-------
-** 2 blades: 1_02 2_04 -*-
IPMC: 1.52 BL: 1.42 FPGA: 2.40 FPGAR: 2.38 BIOS: 1.20
-** 1 blade: 1_03 -*-
IPMC: 1.52 BL: 1.42 FPGA: 2.40 FPGAR: 2.38 BIOS: 1.30
-** 1 blade: 2_02 -*-
IPMC: 1.52 BL: 1.42 FPGA: 2.40 FPGAR: 2.40 BIOS: 1.30

SSD
---
-** 2 blades: 1_02 1_03 -*-
Firmware Version: 2CV102M3
-** 1 blade: 2_02 -*-
Firmware Version: 4PC10362
-** 1 blade: 2_04 -*-
Firmware Version: 4PC10302

Number of cores
-----------------
-** 4 blades: 1_02 1_03 2_02 2_04 -*-
12

Number of CoreXL instances
--------------------------
-** 4 blades: 1_02 1_03 2_02 2_04 -*-
8

CPUs frequency
---------------
-** 4 blades: 1_02 1_03 2_02 2_04 -*-
2.4GHz
Example 2  

```
> asg_version verify
Chassis1
------
SSMI Firmware version: database and hardware do not match:
database - 7.5.20, hardware - 7.5.19

blades
------
Hardware
-------
-**- 1 blade: 1_04 -*-

BIOS version: database and hardware do not match:
database - 1.20, hardware - 1.10
```

Comments

- Using the verify option, the command identifies firmware that is not up to
date, and prints the required version. (Database refers to an internal
database of approved versions) In the example, the firmware on SSM1 does
not match the version recorded in the internal predefined database of
approved versions. SSM1 has firmware version 7.5.19. The internal
database lists the required version as 7.5.20.

- Run in verbose (-v) mode, all hardware components are shown, including
those which have up-to-date firmware.
Displaying System Messages (asg_varlog)

**Description**
Use this command to show system messages written to message files stored in the /var/log directory on SGMs.
- The output is in chronological order
- Each line identifies which SGM logged the system message
- Run `asg_clear_messages` to clear the system buffer of messages

**Syntax**
```
asg_varlog [-b <blades>] [-t <number>] [-f <filter>]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>-b &lt;blades&gt;</td>
<td>Specifies SGMs from which to collect /var/log/messages*.</td>
</tr>
<tr>
<td>-t &lt;number&gt;</td>
<td>Prints the last 10 lines of input to /var/log/messages* from each SGM. When the number of lines is not specified, the last 10 lines of input print to the standard output.</td>
</tr>
<tr>
<td>-f &lt;filter&gt;</td>
<td>Enter a word or phrase to filter by. Only lines that contain the word or phrase are printed to the standard output.</td>
</tr>
</tbody>
</table>

**Example**
```
asg_varlog -f chassis
```

**Output**
```
Nov 10 09:17:14 1_03 61000-ch01-03 kernel: [fw_1];FW-1: [CHASSIS_MGR]: Number of active blades in Local Chassis has changed (prev: 1, current: 2)
```

**Comments**
- 1_03 is the SGM id
- Text that follows "61000-ch01-3 kernel" is the message logged by the specified SGM

Monitoring the System using SNMP

SNMP can be used to monitor various aspects of the 61000 Security System, including:
- Software versions
- Hardware status
- Key performance indicators
- Chassis high availability status

**To monitor the system using SNMP**
1. Upload the MIB to your third-party SNMP monitoring software.
   The SNMP MIB is located on each SGM under: `$CPDIR/lib/snmp/chkpnt.mib`
   For monitoring the 61000 Security System, the only supported OIDs are under
   `iso.org.dod.internet.private.enterprise.checkpoint.products.asg` (OID 1.3.6.1.4.1.2620.1.48)
2. Enable the SNMP agent on the 61000 Security System.
   In `gclish`, run: `set snmp agent on`

**SNMP Traps**
The 61000 Security System supports SNMP traps.
- The SNMP traps MIB is located on each SGM under: `$CPDIR/lib/snmp/chkpnt-trap.mib`
  `iso.org.dod.internet.private.enterprise.checkpoint.products.chkpntTrap`
(OID 1.3.6.1.4.1.2620.2000)
iso.org.dod.internet.private.enterprise.checkpoint.products.asgTrap
(OID 1.3.6.1.4.1.2620.2001)

- To learn more about SNMP, see:
  - Configuring asg alerts (see "Configuring Alerts for SGM and Chassis Events (asg alert)"
System Optimizations

Firewall connections table size

Description:
Firewall connections table default size is 3.5M entries per SGM, regardless of its configuration in SmartDashboard.
This behavior aims to minimize the additional settings, required by customer before deployment.

Configuration:
In order to set a different value, instead of 3.5M, run:
# fw ctl set int fwconn_tab_limit_user <new value, e.g. 4000000>
# update_conf_file fwkern.conf fwconn_tab_limit_user=<new value, e.g. 4000000>
# Install policy

Deactivation:
In order to restore legacy behavior and configure firewall connections table size from SmartDashboard->Gateway Properties->Capacity Optimization->Maximum concurrent connections, run:
# update_conf_file fwkern.conf fwconn_tab_limit_from_policy=1
# reboot -b all

Verification:
To verify firewall connections table size run:
# fw tab -t connections -m 1
And check limit attribute in each blade.
Example:
gcp-ch01-01 > fw tab -t connections -m 1
1_01:
localhost:
-------- connections --------
dynamic, id 8158, attributes: keep, sync, aggressive aging, kbufs 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35, expires 25, refresh, limit 3500000, hashsize 4194304

1_02:
localhost:
-------- connections --------
dynamic, id 8158, attributes: keep, sync, aggressive aging, kbufs 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35, expires 25, refresh, limit 3500000, hashsize 4194304

Reserved connections

Description
Normally, when the connection table limit is reached, no more connections are allowed, even ones critical for operating and managing the gateway. The reserved connections feature allows the gateway to process these critical connections, even after the connections table limit is reached. There is a user defined amount of space that is reserved in the connections table for these critical connections. If the rulebase permits these connections, they are allowed even if no other connections can be accepted.

For example, when the connections table limit is reached, the administrator may not be able to install a new policy that increases the connections limit or open other essential connections, such as SSH to the gateway.
**Notes**

Enforcing the reserved connections limit

The connections table limit is defined in the Capacity Optimization tab, but a certain amount of connections table space is always available for reserved traffic. By default, the number of reserved connections is limited to 2000 and the actual limit of the connections table is increased by this amount.

Before a new connection is recorded, the system verifies that there is enough space in the connections table. If connections table limit is reached, the connection is recorded if it satisfies these conditions:

- The limit is below the limit sum of 'connections table limit' and 'reserved connections limit'
- Connection matches one of the rules in the reserved connections table

Otherwise the connection recording fails.

**Syntax**

\[ \text{asg\_reserved\_conns} \]

**Example 1**

To display the initial list of connections which are allowed to be recorded in the connections table, even if it has reached its defined capacity, run the `asg_reserved_conns` command and choose 1) Print reserved connections table.

**Output**

<table>
<thead>
<tr>
<th>Idx</th>
<th>Source</th>
<th>Mask</th>
<th>Destination</th>
<th>Mask</th>
<th>DPort</th>
<th>Ipp</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>1129</td>
<td>0</td>
<td>Sync</td>
</tr>
<tr>
<td>2)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>130</td>
<td>0</td>
<td>Sync</td>
</tr>
<tr>
<td>3)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>4444</td>
<td>0</td>
<td>Sync</td>
</tr>
<tr>
<td>4)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>Sync</td>
</tr>
<tr>
<td>5)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>8888</td>
<td>0</td>
<td>Sync</td>
</tr>
<tr>
<td>6)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>2010</td>
<td>0</td>
<td>Sync</td>
</tr>
<tr>
<td>7)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>1131</td>
<td>0</td>
<td>Sync</td>
</tr>
<tr>
<td>8)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>256</td>
<td>0</td>
<td>Sync</td>
</tr>
<tr>
<td>9)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Sync</td>
</tr>
<tr>
<td>10)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>eth1-CIN</td>
</tr>
<tr>
<td>11)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>22</td>
<td>8</td>
<td>eth1-CIN</td>
</tr>
<tr>
<td>12)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>23</td>
<td>6</td>
<td>eth1-CIN</td>
</tr>
<tr>
<td>13)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>161</td>
<td>17</td>
<td>eth1-CIN</td>
</tr>
<tr>
<td>14)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>623</td>
<td>17</td>
<td>eth1-CIN</td>
</tr>
<tr>
<td>15)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>eth2-CIN</td>
</tr>
<tr>
<td>16)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>22</td>
<td>6</td>
<td>eth2-CIN</td>
</tr>
<tr>
<td>17)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>23</td>
<td>6</td>
<td>eth2-CIN</td>
</tr>
<tr>
<td>18)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>161</td>
<td>17</td>
<td>eth2-CIN</td>
</tr>
<tr>
<td>19)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>623</td>
<td>17</td>
<td>eth2-CIN</td>
</tr>
<tr>
<td>20)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>22</td>
<td>6</td>
<td>Any</td>
</tr>
<tr>
<td>21)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>256</td>
<td>6</td>
<td>Any</td>
</tr>
<tr>
<td>22)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>18191</td>
<td>6</td>
<td>Any</td>
</tr>
<tr>
<td>23)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>18192</td>
<td>6</td>
<td>Any</td>
</tr>
</tbody>
</table>

Press enter to continue
The first column ‘Idx’ shows the rule number. The second and third columns ‘Source’ and ‘Mask’ show the source IP address and mask. The IP address ‘0.0.0.0’ stands for ‘Any’.

The fifth and sixth columns ‘Destination’ and ‘Mask’ show the destination IP address and mask. The ‘Dport’ column specifies the service number. In case of non-TCP/UDP protocol (6/17) it should be ignored. The ‘Ipp’ column specifies the IP protocol number – 6 for TCP, 17 for UDP, 1 for ICMP and so on. The ‘Interface’ column specifies to which interface the rule applies.

### Example 1

Adding new reserved connection rule:

Run the command `asg_reserved_conns` and choose 2) Add new reserved connection rule

```plaintext
Output

Enter source IP [0.0.0.0]:
>10.10.10.0
Enter source IP mask length [0]:
>24
Enter destination IP [0.0.0.0]:
>20.20.20.0
Enter destination IP mask length [0]:
>24
Enter destination port [0]:
>0
Enter IP protocol number (for example: tcp = 6, udp = 17): >6
Enter interface number [0 – Any]:
0: Any
1: eth1-Mgmt4
2: eth2-Mgmt4
3: BPECH0
4: BPEth1
5: eth1-04
6: eth2-01
7: eth1-Mgmt1
8: eth1-CIN
9: eth2-Mgmt1
10: eth2-CIN
11: Sync
>0
Ok to insert new reserved conn rule: <10.10.10.0/24, 20.20.20.0/24, 0, 6, Any> ? (y/n) y
Entry inserted, rule will apply when new connection will be opened
Press enter to continue
```

The feature works after installation without additional configuration. Use the `asg_reserved_conns` CLI to manage the reserved connections rules.

The rules are recorded in the ‘`reserved_conns_table`’ kernel table.

### Kernel global variables:

- `fwconn_reserved_conn_active`: (type int) enables or disables the feature. Default 1 (enabled).
- `fwconnReserved_limit`: (type int) contains the number of entries in the `reserved_conns_table` kernel table. Default is 2000
Verification

To make sure the feature is configured properly do the following:

1. Check that the value of the kernel global parameter
   `fwconn_reserved_conn_active` is set to 1.

2. Run the command `asg_reserved_conns` and choose 1)
   Print reserved connections table.

3. Run `fw tab -t reserved_conns_table` and make sure that the table
   contains the entries for the rules above

4. Check the contents of the file `$FWDIR/bin/reserved_conns_tab` and
   make sure it contains the rules above

Debugging

To enable reserved connections debugging, set the following kernel global
parameter and use the CONN kernel debug flag to see reserved connections
related debugs.

`fwreserved_conns_debug`: (type int) used to enable reserved connections
debug prints. Default 0 (disabled)

Troubleshooting

1. Run `fw tab -t reserved_conns_table` and make sure that the table
   contains the entries for the rules above

2. Check the contents of the file `$FWDIR/bin/reserved_conns_tab` and
   make sure it contains the rules above. This file is not intended to be
   edited directly.

3. Run the `asg_reserved_conns -f` command to delete all current rules
   from kernel and reload the reserved rules table from the file
   `$FWDIR/bin/reserved_conns_tab` to kernel. It is useful if there were
   changes in network interface names or when
   the `$FWDIR/bin/reserved_conns_tab` file was edited directly.

Policy Acceleration – SecureXL Keep Connections

Description

Allow flow acceleration while pushing policy to the system.

Configuration

Select “Keep all connections” in SDB gateway’s properties->other->connection persistence

Note: Feature is enabled only if:

- SecureXL is enabled
- Firewall blade only is enabled
Extending SecureXL Templates

Description
To enhance connection rate and throughput in a SecureXL enabled environment, the firewall groups together packets of a connection that share the same service (same source port). The first packets of the first connection are handled by the firewall. The firewall then offloads the connection to SecureXL (acceleration hardware or software) for processing.

SecureXL creates a connection template that matches the accept rule in the firewall rulebase, but with a wildcard replacing the source port. New connections that match the template are processed by SecureXL.

On a busy network, repeated connections to the same DNS server clearly benefit from SecureXL acceleration, where the DNS source port (53) is replaced by a wildcard. However, multiple IP addresses can resolve to the same DNS name. In such an environment, replacing the source IP address with a second wildcard decreases the number of connections processed by the firewall.

To replace source IP addresses with a second wild card, you must extend the existing SecureXL templates.

Note - By default, SecureXL template extension is disabled.

To enable SecureXL template extension for accelerated DNS connections:
On the SMO:

1. Exit gclish
   (To exit gclish, enter: shell.)
2. Open: /etc/ppk.boot/boot/modules/simkern.conf for editing.
   If the file does not exist, create it.
3. Add sim_use_srcipWildcard_for_template=1 to the file.
4. Copy the file to all SGMs by running:

Legacy mode:
To allow “Keep all connections” while disabling “SecureXL keep connections” set cphwd_policy_accel=0 in fwker.conf

Verification:
After policy installation, templates of the old policy should be deleted. This can be tracked in the following way:

- Run g_fwaccel stats
- Save the old value of the “Policy deleted tmpl” statistic
- Install policy
- Run g_fwaccel stats again
- Make sure that templates were deleted
Extending SecureXL Templates

5. Open: /etc/fw.boot/modules/fwkern.conf for editing
6. Add cphwd_src_ip_template_enabled=1 to the file.
7. Copy the file to all SGMs by running:
   g_cp2blades -a /etc/fw.boot/modules/fwkern.conf
8. Reboot all SGMs.

In the SecureXL acceleration template, the source IP address and source port are replaced with wildcards.

**Note** - Traffic is only accelerated if DNS is the destination port (53).

To add other services to the template (for example HTTP and Telnet), on the SMO:

1. Exit gclish
   (To exit gclish, enter: shell.)
2. Open: /etc/fw.boot/modules/fwkern.conf for editing
3. Add cphwd_use_srcip wildcard_for_template=80,23 to the file.
   This adds ports 80 and 23 to the list of permitted destination ports.
   - Separate each port number with a comma
   - Do not add more than 4 port numbers
   For UDP services, add: cphwd_src_ip_tmpl_udp_ports= <UDP port numbers>.
4. Copy the file to all SGMs by running:
   g_cp2blades -a /etc/fw.boot/modules/fwkern.conf
5. Open /etc/ppk.boot/boot/modules/simkern.conf for editing.
6. Add sim_src_ip_tmpl_tcp_ports=80,23 to the file.
   For UDP services, add sim_src_ip_tmpl_udp_ports=<UDP port numbers>
7. /etc/ppk.boot/boot/modules/simkern.conf on all blades
8. Copy the file to all SGMs by running:
   g_cp2blades -a /etc/ppk.boot/boot/modules/simkern.conf
9. Reboot all SGMs.

**Verification**

To make sure extended SecureXL templates are being used:

1. In gclish, run: fwaccel templates.
2. Examine the output.

```
> fwaccel templates

1.0.1.1  Source Destination OPort PR  Flags Conns LTC DLY C25 i/f S2C i/f Trst Identity
     ---------- ---------- ---------- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ---- ----
   11.11.11.100 22 6 ...... 1 2 1 2/5 3/2 1 0

1 Total number of templates: 1
   1 blade: 1.0.1.1
   0 BPECHO
   1 BEFNL
   2 eth1-0L
   3 eth2-Ngml
   5 eth2-0L
   6 eth2-Ngml
   8 Sync
   9 eth1-CIN
  10 eth2-CIN

An asterisk (*) in the **Source** column and an increasing **Conns** counter means the extended template is being utilized.
```
Configuring DNS Session Rate (cphwd_udp_selective_delay_ha)

Description
To improve the DNS session rate, the 61000 Security Systems implements two enhancements:

- **Delayed Connection**
  When a DNS connection matches a SecureXL template, the 61000 Security Systems firewall is not immediately notified. The notification is delayed using the global parameter: `cphwd_udp_selective_delay_ha`. After a delay is set, the connection is handled completely by the acceleration device.

  **Note** - If the connection is not completely handled (and closed) by the acceleration device during the set delay period, then the firewall is notified in the usual manner.

- **Delete on Response**
  After the DNS response is received, the connection is immediately deleted from the gateway instead of being kept for an additional 60 seconds (the UDP connection default timeout).

Syntax
From gclish, run these commands in this order:

1. `>fw ctl set int cphwd_udp_selective_delay_ha <delay in seconds>`
2. `>fwaccel off`
3. `>fwaccel on`

Verification
To make sure that DNS connections are delayed by the set value:

1. Open several DNS connections from the same client to the same server
2. Run: `fwaccel templates`

<table>
<thead>
<tr>
<th>Source</th>
<th>SPort</th>
<th>Destination</th>
<th>DPort</th>
<th>Flags</th>
<th>Conn</th>
<th>LCT</th>
<th>DLY</th>
<th>Cls 1/f</th>
<th>S2C f/</th>
<th>Inst Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.10.10.1</td>
<td>53</td>
<td>127.0.0.1</td>
<td>53</td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
<td>0x00</td>
</tr>
</tbody>
</table>

   The delay you see for the DNS template (under DLY field) should match the value specified for `cphwd_udp_selective_delay_ha`.

   **Note** - The default value for this parameter is 30 seconds. The maximum value is 60.

To make the Enhancements Permanent:

Update `fwkern.conf` by running:

```
> update_conf_file fwkern.conf cphwd_udp_selective_delay_ha=<delay>
```

To turn off the Enhancements:

To turn off Delayed Connection and Delete on Response:

- `Set cphwd_udp_selective_delay_ha` to zero, or
- Remove all services from `cphwd_delayed_udp_ports`.

  **Note** - this disables both enhancements.

Extending Session Rate Enhancements to other UDP Services

By modifying the value of `cphwd_delayed_udp_ports` in `fwkern.conf`, you can extend the benefits of these two DNS session rate enhancements to other services. For example, to add UDP service 100 to the list, from gclish run:

```
> update_conf_file fwkern.conf cphwd_delayed_udp_ports=53,100,0,0,0,0,0,0
```
**Note -**

- The number of services is limited to 8.
- The command must contain 8 values. If you configure less than 8 services, enter 0 for the others.
- Directly updating `fwkern.conf` is the only way to extend DNS session rate enhancements to other UDP services (`fw ctl set int` is not supported).
- The configuration takes effect only after reboot.
Configuring VLAN performance enhancement (asg_affinity_enhance)

Description
By default VLAN traffic goes only to single receive queue on the network interface card (NIC), thus only 1 core can be used per interface (BPEth). The reason for that is that RSS (Receive Multi-Queue feature in the NIC) by default does not work on packets with double vlan header (the 61k switch adds the extra vlan header). Thus, we have added a feature to enable RSS for double vlan packets (what we call "vlan traffic") to utilize 4 cores (instead of 2) and thus improve vlan packet rate significantly.
Note: this mode causes ~18% degradation for clear packet rate (from 2.4Mpps to 2Mpps on single blade).

Syntax
asg_affinity_enhance [ -s | -u | -v | -d | -h ]

Options:
- s : turn on multi-queue for vlan (for improved vlan packet-rate)
- u : turn off multi-queue for vlan (for improved clear packet-rate)
- v : show current setting
- d : restore default setting (off)
- h : show this help

Example
To enable VLAN performance enhancement run:
gperf-ch01-02 > asg_affinity_enhance -s
-- 1 blade: 1_02 --
VLAN performance enhancement has been Enabled

To disable VLAN performance enhancement:
gperf-ch01-02 > asg_affinity_enhance -u
-- 1 blade: 1_02 --
VLAN performance enhancement has been disabled

Fast packet drop

Description
Fast packet drop can be used in situations, such as when under DoS attack, to drop unwanted packets as early as possible in the packet processing path. This makes the gateway’s resources available to process legitimate traffic. The rulebase is in a configuration file that defines which packets should be dropped.

Syntax
sim dropcfg < -l | -f <file> | -r | -y | -h >
Parameter  | Description
---|---
-l | Show current configuration
-f <file> | Set configuration file name
-r | Reset drop rules
-y | Do not require confirmation
-h | Show usage information

### Configuration

Configuration Steps:

1. Create the rulebase configuration file (see details below)
2. Copy the configuration file to all SGMs [run from gclish: `asg_cp2blades <file-name-full-path>`]
3. Apply Fast packet drop [ run: `sim dropcfg -f <configuration file>`]

The rulebase configuration is specified using the `-f` CLI option. It contains drop rules, and each line should contain a single rule.

Each rule line must contain one or more of the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>src &lt;source ip&gt;/&lt;subnet&gt;</td>
<td>Source IP address and subnet. Subnet is optional</td>
</tr>
<tr>
<td>dst &lt;destination ip&gt;/&lt;subnet&gt;</td>
<td>Destination IP address and subnet. Subnet is optional</td>
</tr>
<tr>
<td>dport &lt;destination port&gt;</td>
<td>Destination port.</td>
</tr>
<tr>
<td>proto &lt;ip protocol&gt;</td>
<td>IP Protocol (e.g. TCP=6,UDP=17,ICMP=1)</td>
</tr>
</tbody>
</table>

### Notes

If subnet is not specified, a single IP address is used.

Use `*` to specify 'Any'. It is the same as not specifying the parameter.

Use '#' at the beginning of the line to add comments.

Empty lines are ignored.

### Examples

Example configuration file:

```
src 1.1.1.1
dport 80 proto 6
src 1.1.0/24 dst 2.2.0.0/16 dport 53 proto 17
```
Verification

To make sure fast packet drop rules are being enforced, run the command:

```
sim dropcfg -l
```

The output shows list of active drop rules:

```
Drop rules (Match after conn lookup):
Source             Destination        DPort PR
------------------ ------------------ ------ ----
  1.1.1.1/32                  *     *   *
      *                  *    80   6
  1.1.1.0/24         2.2.0.0/16    53  17
```

Disabling Fast Packet Drop

If there are drop rules defined, run the following command to clear the fast packet drop rulebase:

```
sim dropcfg -r
```

System Under Load

**Description**

System Under Load feature (SUL) enables the Gateway to monitor high CPU load and also suspends setting remote SGMs to DOWN state when cannot receive CCP packets for a timeout of `BLADE_DEAD_INTERVAL` (default is 3 sec) and when SUL state ON.

It enables every SGM to act differently when they/others SGM are under load.

Being under load (SUL state ON) meaning at least one SGM has reported Kernel CPU Usage above threshold of 80% by default (`CPU threshold`)

Highest average Kernel CPU usage of a single core is being calculated locally and is published via CCP packets to remote SGMs

The average is based on 5 samples by default (`Number of sample`) – sample is taken every 2 HA Time Units (HTU = 0.1s)

Every SGM calculates its own Kernel High CPU

Local Kernel High CPU usage and remote usage have almost the same handler with minor changes

1. Local or Remote Kernel High CPU will set SUL state ON
2. Local User space + Kernel High CPU will triggers PNOTE timeout postponer to all user-space PNOTES (etc fwd) on local SGM
### SUL state change

#### Feature flow

- SUL set to ON - if reported high CPU

- SUL will set to OFF if no report has been received for at least 10 seconds by default from the last report (*short timeout*)

  if system is continually under load (high CPU report gap is less then *short timeout*, SUL will stay ON for up to 3 minutes by default (*Long interval*)

**When / why SUL is ON?**

1. Every SGM calculates CPU usage on all cores, picking the highest and stores in memory.
2. on every CPU state check (called periodically) we take the average of recent 5 highest samples (*Number of sample*) and publish via CCP
3. by receiving CCP with SGM CPU:
   a. if > threshold (*CPU threshold*) --> toggle SUL ON
4. by calculating locally:
   a. if > threshold (*CPU threshold*) --> toggle SUL ON
   b. --> local load is ON (for local user-space PNOTEs

  SUL ON mode will be delayed for a fixed timeout (*Start timeout*) (default=0) if at least one SGM continually reports high CPU more than 3min (*Long interval*) and the reason for setting OFF from the begging was the *long-timeout* expiration

**When / why SUL is OFF?**

SUL can be toggle OFF after one of the following scenarios:

1. System is idle - no SGM reported High CPU usage for at least 10 seconds (default timeout of *Short timeout*)
2. System is Under Load for too long - after a fixed watermark of 3 minutes (*Long interval*), the SUL in ON, it will be forced to toggle OFF, even if SGMs still reporting High CPU. SUL will be ON again if they will keep reporting high CPU after the shutdown but only after fix timeout – 0 by default is over (*Start timeout*)
3. User decided to manually disable the feature while SUL was ON

#### Syntax

```
fw ctl set int fwha_pnote_timeout_mechanism_monitor_cpu <value>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Turns SUL mechanism ON</td>
</tr>
<tr>
<td>1</td>
<td>Turns SUL mechanism OFF</td>
</tr>
</tbody>
</table>

#### Example

Enabling SUL feature: (SUL is enabled by default)

```
fw ctl set int fwha_pnote_timeout_mechanism_monitor_cpu 1
```

#### Output

Every state change (ON/OFF) is logged via SVT & /var/log/messages (dmesg), when (only SMO sends the SVT messages)

Log Example via SVT:

```
cluster_info : (Bladed System) FW: [SUL] Changing SUL state to OFF due to High CPU usage (6%) on remote Blade 1,3, threshold = 1%, local CPU usage is 13% fw_message [1,1]
cluster_info : (Bladed System) FW: [SUL] Changing SUL state to ON due to High CPU usage (8%) on remote Blade 1,3, threshold = 1%, local CPU usage is 13% fw_message [1,1]
cluster_info : (Bladed System) FW: [SUL] Changing SUL state to ON due to High CPU usage (6%) on remote Blade 1,3, threshold = 1%, local CPU usage is 13% fw_message [1,1]
cluster_info : (Bladed System) FW: [SUL] Changing SUL state to ON due to High CPU usage (8%) on remote Blade 1,3, threshold = 1%, local CPU usage is 13% fw_message [1,1]
```

---

**System Under Load**

**System Optimizations**

Page 78
**Tuning feature**  
SUL feature can be modified and tuned to meet user specific needs.

**Parameters**

**Syntax**

```
fw ctl set int <parameter> <numerical value>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fwha_pnote_timeout_mechanism_cpu_load_limit</td>
<td>(CPU threshold) (highest average CPU usage of a single core) default = 80</td>
</tr>
<tr>
<td>fwha_sul_num_sample_cpu_check</td>
<td>(Number of sample) (on how many samples the CPU avg will be based on; sample is taken every 2 HTUs) default = 5 HTU - HA Time Unit (=0.1s)</td>
</tr>
<tr>
<td>fwha_pnote_timeout_mechanism_disable_feature_timeout</td>
<td>(Long interval) (maximum continues time allowed for SUL ON state) default = 1800 HTU (3 minutes) HTU - HA Time Unit (=0.1s)</td>
</tr>
<tr>
<td>fwha_system_under_load_short_timeout</td>
<td>(Short timeout) (low CPU usage period for setting SUL OFF) default = 100 HTU (10 seconds) HTU - HA Time Unit (=0.1s)</td>
</tr>
<tr>
<td>fwha_system_under_load_start_timeout</td>
<td>(Start timeout) (delay time between next SUL ON, if last ON period interrupted by Long interval) default = 0 HTU (0 seconds) HTU - HA Time Unit (=0.1s)</td>
</tr>
</tbody>
</table>

**Notes**

In order for the modified SUL parameters, including state (ON/OFF) to survive reboot  
Add them to the **fwkern.conf** file using **g_update_conf_file** utility

---

**Jumbo Frames**

**Description**

61000 Security System has the capability to support Jumbo Frames but it is still not fully integrated. For that reason the configuration is not trivial and requires several steps:

1. Configuration of Jumbo Frames on the SSM.  
2. Enabling Jumbo Frames on the gateway.  
3. Configuring Jumbo Frames on the gateway interfaces.
The maximum MTU supported is 9146 for SSM60 systems and 12288 for SSM160 systems. 

**Note:** Jumbo Frame configuration is not supported on Bond LACP interfaces.

### Configuration

**Configuration of Jumbo Frames on the SSM**

In order to allow Jumbo Frames on the SSM we need to modify the MTU of the ports leading to the blades (downlinks) and of the front panel ports we wish to allow Jumbo Frames on.

**Instructions for SSM60:**

1. Connect to the SSM with telnet:
   
   Use `show chassis id <chassis_ID> module SSM<SSM ID> ip` to verify the SSM IP. Password is admin.

2. Issue "en" to enter "enable" mode.

3. Issue "conf t" to enter the Configuration terminal.

4. Select to configure all the downlinks interfaces ("#interface range 1/2-1/14/1")

5. Set the required MTU (#packet-size-limit 9146).

6. Select to configure the required front panel ports. Interfaces 1/15/1 – 1/15/5 represents ports 1-5 of the SSM. ("#interface range 1/15/1-1/15/5" for all of them).

7. Set the required MTU (#packet-size-limit 9146).

8. Exit Configuration terminal ("#end") and save configuration ("#write").

```
# telnet 198.51.100.32
Trying 198.51.100.32...
Connected to 198.51.100.32.
Escape character is '^]'.

User Access Verification

Password:
FI_cp>en
FI_cp#conf t
FI_cp(config)#interface range 1/2/1-1/14/1
FI_cp(config-if-group)#packet-size-limit 9146
FI_cp(config-if-group)#interface range 1/15/1-1/15/5
FI_cp(config-if-group)#packet-size-limit 9146
FI_cp(config-if-group)#end
FI_cp#write
```

**Instructions for SSM160**

1. Use `asg_chassis_ctrl` to enable/disable Jumbo frames on the SSMs.

2. Use `asg_chassis_ctrl` to set the MTU of the SSM ports.
# asg_chassis_ctrl jumbo_frames enable 1
Jumbo frames are enabled on SSM1
# asg_chassis_ctrl jumbo_frames enable 2
Jumbo frames are enabled on SSM2
# asg_chassis_ctrl set_port_mtu 1 1 9000
MTU of port 1 on SSM1 was set to 9000

Enabling Jumbo Frames on the gateway

The utility asg_jumbo_conf allow us to enable and disable Jumbo frames on the gateway. This utility is available only from BASH shell, use "shell" command to move to BASH shell from gclish. In order to enable Jumbo Frames it should be issued with the flag “enable”.

# asg_jumbo_conf enable
Jumbo frames enabled. Don't forget to set the MTU of relevant interfaces in gclish.

Note: In SSM160 systems this action will also enable Jumbo frames on the SSMs, but only for the local chassis.

Configuring Jumbo Frames on the gateway interfaces

The pseudo interfaces configuration is done via gclish.

1. Enter gclish and set the required MTU on the relevant interface (set interface eth2-04 mtu 9000 for example).
2. Save the new configuration.

> set interface eth2-04 mtu 9000
2_01: success
2_02: success
2_03: success
> save config
2_01: success
2_02: success
2_03: success

Validation

Before you start transmitting jumbo frames via the gateway it is recommended to verify your Jumbo Frames configuration

SSM60 Configuration Validation

1. Connect to the SSM with telnet:
   Use show chassis id <chassis ID> module SSM<SSM ID> ip to verify the SSM IP. Password is admin.
2. Issue "en" to enter "enable" mode.
3. Issue "show run" to display the running configuration.
4. Verify that under the relevant interfaces (downlinks and front panel ports) the required packet size limit appears.
# telnet 198.51.100.32
Trying 198.51.100.32...
Connected to 198.51.100.32.
Escape character is '^]'.

User Access Verification

Password:
FI_cp>en
#show run
!
interface 1/2/1
flow-control disable
packet-size-limit 9146

**SSM160 Configuration Validation**

1. **Use** `asg_chassis_ctrl jumbo_frames` **to display the current Jumbo frames configuration on the SSMs.**
2. **Use** `asg_chassis_ctrl get_port_mtu` **to verify the MTU of specific ports on the SSMs.**

```bash
# asg_chassis_ctrl jumbo_frames show 1
Jumbo frames are enabled on SSM1
# asg_chassis_ctrl get_port_mtu 1 1
MTU of port 1 on SSM1 is 9000
```

**SGM Configuration Validation**

The `asg_jumbo_conf` utility has a “show” flag which allows us to view the current setting. It also has a `verbose` flag, `-v`, which supplies additional information.

```bash
# asg_jumbo_conf show
Jumbo frames are enabled (SSM1 max MTU: 9146, SSM2 max MTU: 9146)

# asg_jumbo_conf show -v
Jumbo frames are enabled (SSM1 max MTU: 9146, SSM2 max MTU: 9146)
Current interfaces MTU configuration:
interface:BPEth0:mtu 9146
interface:BPEth1:mtu 9146
interface:eth1-01:mtu 3500
interface:eth1-02:mtu 6500
interface:eth1-03:mtu 9146
interface:eth2-01:mtu 9146
interface:eth2-02:mtu 9000
interface:eth2-03:mtu 9146
```

The MTU of all the interfaces which are not in the list is 1500.

**Jumbo Frame configuration on Bond LACP Interface**

**Note:** Before you can configure support for Jumbo Frames on a bond LACP interface you need to install a driver hotfix. Contact Check Point support to get the driver hotfix.

**Configuration:**

To configure support for Jumbo Frames on bond interfaces, follow the Jumbo Frames configuration instructions, with the following differences for bond interfaces:

1. **Apply the SSM configuration to all slave interfaces.**
2. Apply the SGM configuration to the Bond interface (no need to configure it on the slaves)

Example:

This example shows configuration of Jumbo Frames on Bond interface bond0 which has two slaves, eth1-01 and eth2-01
The configuration includes 2 SSM160.

Enabling Jumbo Frames on the gateway
To enable Jumbo frames:
# asg_jumbo_conf enable
Don't forget to set the MTU of relevant interfaces in gclish.

Set the MTU of the slaves ports on the SSM
To set the MTU of port 1 on SSM1 to 9000
# asg_chassis_ctrl set_port_mtu 1 1 9000
To set the MTU of port 1 on SSM2 to 9000
# asg_chassis_ctrl set_port_mtu 2 1 9000

Configuring Jumbo Frames on the Bond interface on the SGM
> set interface bond0 mtu 9000
  2_01: success
  2_02: success
  2_03: success
TCP MSS Adjustment

Description:

The TCP MSS Adjustment feature allows MSS (Maximum Segment Size) clamping of TCP traffic. It enables the configuration of the MSS that is part of the OPTIONS in the TCP header.

This feature lets you prevent fragmentation when the MTU value on the communication path is lower than the MSS value.

Configuration:

From gclish run:

**Syntax:** `fw ctl set int <parameter> <numerical value>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fw_clamp_tcp_mss</code></td>
<td>Enable or Disable MSS Adjustment.</td>
</tr>
<tr>
<td></td>
<td>0 – Disable (default)</td>
</tr>
<tr>
<td></td>
<td>1 – Enable</td>
</tr>
<tr>
<td><code>fw_tcp_mss_value &lt;value&gt;</code></td>
<td>Set the MSS value. If value is set to 0, the MSS value is taken from the interface MTU</td>
</tr>
</tbody>
</table>

**Note:** In order for the modified parameters, including state (ON/OFF), to survive reboot - add them to the fwkern.conf file using g_update_conf_file utility from Expert shell.

Verification

Monitoring can be done using Packet Sniffers to verify that indeed MSS is clamped when the feature is enabled according to configuration.

**Note:** MSS value is applied on all interfaces, including Management

Debug

- Enable SIM debug using the command: `sim dbg -m pkt + pkt`.
- Start fw debugging using the command: `fw ctl zdebug + packet`.
- Look for prints that contain the string 'MSS'.
Session Control

Description:
The Session Control feature sets an upper limit for the allowed concurrent connections created by the same source IP address.

Feature capabilities:
1. Enable / Disable the feature on a specific Gateway.
2. Set the global limit.
3. Enable / Disable logging when the limit on specific source IP address is exceeded.
4. Ability to define exceptions to the general limitations:
   - Set the IPv4 or IPv6 address of the exception source IP address.
   - Set the specific limit for the exception.
   - Ability to set exception as unlimited.

Configuration:
The initial configuration must be done on the management side using GuiDBEdit editing tool.

After the initial configuration has been installed to the Security Gateway, the Administrator can edit the configuration from the CLI on the Security Gateway.

GuiDBEdit Configuration
1. Open the GuiDBEdit tool.
2. In "Network Objects" select the required Gateway object.
3. Go to the "firewall_settings" attribute.
4. Go to the "conns_limit_by_ip" attribute.
5. Enable the session control feature by setting the "conns_limit_by_ip_enable" attribute to true.
6. Set the "conns_limit_by_ip_global" to the default upper limit for concurrent connections. (Zero value is not allowed.)
7. Set the "conns_limit_by_ip_log" to enable or disable logs when the limit on specific source IP address is exceeded.
8. Double-click the "conns_limit_by_ip_excpts" attribute. A new exception entry is created. (Repeat this step for each required exception.)
   - Optional: set the exception entry name.
   - Set one source IP address. Either in the "ipv4_addr" attribute for IPv4 or in the "ipv6_addr" attribute for IPv6. Do not configure both attributes.
   - Set the connection direction: set <0> for Client to server.
   - Set the specific connections limit for this exception in the "max_conns" attribute. (Zero value stands for allow unlimited connection for this source IP.)
9. Save and close the GuiDBEdit tool.
10. Install the policy on the target Gateways.
Gateway Configuration

The following CLI commands are valid only after you install the policy with the session control feature at least once.

**To set/get Session Control capabilities run from gclish:**

**Syntax:** `fw ctl set int <parameter> <numerical value>`

```
fw ctl set int
```

```
fw ctl get int
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conns_limit_by_ip_global</td>
<td>Set/Get upper limit for the allowed concurrent connections created by the same source IP address</td>
</tr>
</tbody>
</table>

Note: To apply the settings run: "fwaccel off" and "fwaccel on"
conns_limit_by_ip_log

Enable or Disable logs when allowed concurrent connection limit on specific source IP address is exceeded.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Disable</td>
</tr>
<tr>
<td>1</td>
<td>Enable</td>
</tr>
</tbody>
</table>

Note: To apply the settings run: "fwaccel off" and "fwaccel on"

conns_limit_by_ip_enable <0>

Disable the source control feature immediately.

Note: To enable the feature again run: "waccel off" and "fwaccel on"

Note: In order for the modified parameters to survive reboot, add them to the fwkern.conf file using g_update_conf_file utility from Expert shell.

To print the connections limit table run:

Syntax: fwaccel conns_limit [\-s | \-m <max entries>] [\-h]

Description: Prints the whole connections limit table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s</td>
<td>Print only number of connections limits</td>
</tr>
<tr>
<td>-m &lt;max entries&gt;</td>
<td>Maximum number of entries to print</td>
</tr>
<tr>
<td>-h</td>
<td>Show this help</td>
</tr>
</tbody>
</table>

Connections Limit Table fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Address</td>
<td>Source IP Address</td>
</tr>
<tr>
<td>Direction</td>
<td>C2S for client to server and S2C for server to client</td>
</tr>
<tr>
<td>Flags</td>
<td>x - Entry was inserted by the administrator as exception. u - Entry was signed by the administrator as unlimited. I - A log was sent once for this source IP.</td>
</tr>
</tbody>
</table>
No Acceleration Disabling On Account Of Traceroute Rule (asg_tmpl_special_svcs)
(Appplies to R75.052 only)

Validation

1. Send traffic through the Gateway from different sources.
2. From the CLI, run "fwaccel conns_limit" to see the state of the connections limit table for the source IP addresses.
3. When the number of allowed concurrent connections for specific source IP address is reached, the next connection is dropped. A log message is created based on the logging configuration.

Notes

1. By default the feature is disabled.
2. connection_limit table is not synchronized between blades.
3. The expected value for the global limit or exception's limit should be above connections. Less can cause failures in simple WEB operations like browsing.
4. While installing the policy, the system generates "Concurrent Connections limit exceptions" signed as unlimited for all the Gateway interfaces. This prevents the system from blocking Check Point’s management transportation as a result of exceeding the limit.
5. Feature configuration requires Management Server hotfix. Contact Checkpoint Support.

No Acceleration Disabling On Account Of Traceroute Rule (asg_tmpl_special_svcs)
(Appplies to R75.052 only)

Description
This feature safely prevents security policy rules with the service "traceroute" from disabling acceleration for all subsequent rules.

Syntax
asg_tmpl_special_svcs [on|off]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>Enable – acceleration will not be disabled on account of traceroute rules.</td>
</tr>
<tr>
<td>off</td>
<td>Disable</td>
</tr>
</tbody>
</table>

Example
asg_tmpl_special_svcs on

Output
Configuration successful. Install policy to apply changes.

Comments
- This functionality requires a patch on the Management side, which can be acquired by contacting Check Point Support.
- For this feature to function correctly, the traceroute service object in SmartDashboard must remain with default settings and not customized.
NAT and the Correction Layer

For optimum system performance, a session from start to finish should be handled by the same SGM. With NAT, a connection from the same session might be distributed to a different SGM. The system Correction Layer then has to forward the connection to the correct SGM.

Correctly configuring distribution modes keeps corrections situations to a minimum and optimizes system performance. To achieve optimal distribution between SGMs on the gateway:

- **When not using NAT rules**
  Set the General distribution mode.

- **When using NAT rules**
  Set the hidden network(s) to User Mode, and the destination network(s) to Network Mode.

Session rate throttling *(Applies to R75.052 only)*

**Description**  
The purpose of session rate throttling is to limit connection establishment rate according to a pre-defined set of rules. The feature is controlled through the “asg_session_control” command, which is only available from the “Expert” shell.

The feature is disabled by default – no session rate throttling will take place.

**Syntax:**

```
asg_session_control <apply | disable | stats | verify>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>apply</td>
<td>Applies the rules from &lt;$FWDIR/conf/control_rules&gt; to all of the blades in the security group.</td>
</tr>
<tr>
<td>disable</td>
<td>Removes all rules from all of the blades in the security group. No connection per second limit will be enforced. The control_rules file is not deleted, it is only ignored.</td>
</tr>
<tr>
<td>stats</td>
<td>Shows drop statistics</td>
</tr>
<tr>
<td>verify</td>
<td>Verifies that the file &lt;$FWDIR/conf/control_rules&gt; and that the loaded rules are identical on all of the blades.</td>
</tr>
</tbody>
</table>

**Abbreviation:** asg_session_control uses the following abbreviations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source, src</td>
<td>The source address or network.</td>
</tr>
<tr>
<td>Destination, dst</td>
<td>The destination address or network.</td>
</tr>
<tr>
<td>DPort, dport</td>
<td>The destination port.</td>
</tr>
<tr>
<td>PR, proto</td>
<td>The Protocol number.</td>
</tr>
<tr>
<td>Limit</td>
<td>The maximum session establishment rate.</td>
</tr>
</tbody>
</table>
Rules configuration

The session throttling rules configuration is done by editing a rules file on one of the SGMs and then applying the rules by running "asg_session_control apply". Upon rules application, the configuration file is copied and applied on all of the SGMs.

Location: The rules file location is <$FWDIR/conf/control_rules>.

Format:
Each line represents a session control rule and should specify the following:

src <ip/mask> dst <ip/mask> dport <port> proto <protocol_id> limit <session_establishment_rate>

At least one of the parameters <src, dst, dport, proto> must be specified.
The parameter <limit> must be specified.
If a parameter is omitted, “any” is assumed. Use * in order to explicitly configure “any”.

Examples:
1. src * dst 1.1.1.0/24 dport 67 proto 17 limit 20
   This rule specifies that a maximum of 20 new connections on destination port 67 and protocol 17 will be established each second to any destination in subnet 1.1.1.0. Additional connections will be dropped while other ports or protocols will behave normally.
2. dst 2.2.2.2/32 dport 80 proto 6 limit 13
   This rule limits the amount of new connection to destination server 2.2.2.2 on destination port 80 and protocol 6 to 13.
Example 1  asg_session_control apply

```
-** 2 blades: 1_01 1_02  **-

<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Source</th>
<th>Destination</th>
<th>DPort PR Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>*</td>
<td>1.1.1.0/24</td>
<td>67 17 20</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>2.2.2.2/32</td>
<td>80 6 13</td>
</tr>
</tbody>
</table>
```

Session rate entries configured successfully

The “apply” command returns an output indicating which rules each blade loaded. All blades should return the same output.

Example 2  asg_session_control stats

```
1_0:
<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Source</th>
<th>Destination</th>
<th>DPort PR Limit</th>
<th>Drops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>*</td>
<td>1.1.1.0/24</td>
<td>67 17 20</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>2.2.2.2/32</td>
<td>80 6 13</td>
<td>0</td>
</tr>
</tbody>
</table>
```

```
1_02:
<table>
<thead>
<tr>
<th>Rule ID</th>
<th>Source</th>
<th>Destination</th>
<th>DPort PR Limit</th>
<th>Drops</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>*</td>
<td>1.1.1.0/24</td>
<td>67 17 20</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>*</td>
<td>2.2.2.2/32</td>
<td>80 6 13</td>
<td>2</td>
</tr>
</tbody>
</table>
```

The “Drop” column represents the number of packets each blade dropped as a result of each rule matching. As seen in the example, blade 1_01 dropped three packets as a result of rule ID 1, whereas blade 1_02 dropped two packets as a result of rule ID 2.

Example 3  asg_session_control disable

```
-** 2 blades: 1_01 1_02  **-
Resetting session rate entries
```

Session rate entries configured successfully

```
```

System Configuration

Administration

gclish

Description:
The gclish is a command line interface which stands for global clish.
It is used like clish, but the commands are global by default, hence they are performed on all the SGMs, which are part of the security group.

gclish commands are not applied on down SGMs: If a set command is performed while a SGM was down (either administratively or not) it will not be applied on it. The down SGM will sync its database during its startup process. If the database was changed, the SGM will reboot itself in order for the changes to apply.
clish commands are documented in Gaia Admin Guide. Almost all commands are also available in 61000.

Few notes:

1. 61000 introduces chassis feature, which is documented in the hardware monitoring and chassis HA section
2. In 61000 auditlog is enabled by default. All set commands are recorded to log and can be retrieved with `asg_auditlog` (documented separately)
3. config-lock is the lock that protects gclish database. The lock can be held by single SGM per system. When user attempts to perform gclish set operations from specific SGM, he should make sure that this SGM holds the config-lock. In order to acquire config-lock, the command `set config-lock on override` should be executed
4. As mentioned afore, gclish commands are applied on all SGMs, which are part of the security group. Command output will include the list of these SGM and their reply. See examples below
5. gclish traffic runs on Sync interface, port 1129/TCP
6. Similarly to Gaia, gclish is capable of running extended commands. Use `show commands extended` to see the list of extended commands, which can run from gclish
7. In order to run command on specific set of SGMs, use the `blade-range` specification. Once specifying blade-range, all gclish embedded commands will run only on this subset of SGMs. Since all SGMs must have identical configuration, the use of `blade-range` is hardly recommended

**CP global commands**

**Description:**

The global commands are utilities to run certain commands on multiple SGMs. This document is dealing with Check Point products related commands, those utilities are mostly extended wrapper to known Check Point products commands (like `fw`, `sim`, `fwaccel`). And some new utilities that are related to those products (`cpconfig`)

The general global command syntax is shown in “OS global commands” document

The list of available commands is: `sim`, `sim6`, `fwaccel`, `fwaccel6`, `fw`, `fw6`, `cpconfig`

Those commands are available in the gclish in addition they are available in bash with the initial “g_”

Other relevant documents may include “OS global commands” and “General commands”.

**sim, sim6**

**Description:**

When invoked from gclish, `sim/sim6` commands are global wrappers to the known `sim/sim6` command.

`sim/sim6` are, for most parameters, comparison type global commands which shows unified information from all SGMs.

Note: sim affinity is not supported on 61000 Security System, `g_mq_affinity` should be used instead.

**fwaccel, fwaccel6**

**Description:**

When invoked from gclish, `fwaccel/fwaccel6` commands are global wrappers to the known `fwaccel/fwaccel6` command.

`fwaccel/fwaccel6` are, for most parameters, comparison type global commands which shows unified information from all SGMs. “fwaccel stats” and “fwaccel notifstats” commands shows an aggregated statistics from all SGMs
Example:

gdua17-t43-ch02-02 > fwaccel stats
Displaying aggregated data from blades: all

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>accel packets</td>
<td>6518</td>
<td>accel bytes</td>
<td>870476</td>
</tr>
<tr>
<td>connss created</td>
<td>38848</td>
<td>connss deleted</td>
<td>38043</td>
</tr>
<tr>
<td>C total connss</td>
<td>801</td>
<td>C templates</td>
<td>0</td>
</tr>
<tr>
<td>C TCP connss</td>
<td>493</td>
<td>C delayed TCP connss</td>
<td>0</td>
</tr>
<tr>
<td>C non TCP connss</td>
<td>308</td>
<td>C delayed nonTCP conn</td>
<td>0</td>
</tr>
<tr>
<td>connss from templates</td>
<td>0</td>
<td>temporary connss</td>
<td>0</td>
</tr>
<tr>
<td>nat connss</td>
<td>0</td>
<td>C nat connss</td>
<td>0</td>
</tr>
<tr>
<td>dropped packetss</td>
<td>0</td>
<td>dropped bytes</td>
<td>0</td>
</tr>
<tr>
<td>nat templates</td>
<td>0</td>
<td>port alloc templates</td>
<td>0</td>
</tr>
<tr>
<td>connss from nat tmpl</td>
<td>0</td>
<td>port alloc connss</td>
<td>0</td>
</tr>
<tr>
<td>Policy deleted tmpl</td>
<td>0</td>
<td>C Policy deleted tmpl</td>
<td>0</td>
</tr>
</tbody>
</table>

Accelerated VPN Path

| C crypt connss        | 0         | enc bytes             | 0         |
| dec bytes             | 0         | ESP enc pkts          | 0         |
| ESP enc err           | 0         | ESP dec pkts          | 0         |
| ESP dec err           | 0         | ESP other err         | 0         |
| AH enc pkts           | 0         | AH enc err            | 0         |
| AH dec pkts           | 0         | AH dec err            | 0         |
| AH other err          | 0         | espudp enc pkts       | 0         |
| espudp enc err        | 0         | espudp dec pkts       | 0         |
| espudp dec err        | 0         | espudp other err      | 0         |

Medium Path

| PXL packets           | 0         | PXL async packets     | 0         |
| PXL bytes             | 0         | PXL connss            | 0         |
| C PXL connss          | 0         | C PXL templates       | 0         |

Firewall Path

| F2F packets           | 10077862  | F2F bytes             | 118505123  |
| F2F connss            | 38839     | C F2F connss          | 800        |
| TCP violations         | 0         | C partial connss      | 0         |
| C anticipated connss  | 0         |                        |           |

General

| memory used           | 0         | free memory           | 0         |

(*) Statistics marked with C refer to current value, others refer to total value

Monitoring mode: fwaccel_m is an extension to global fwaccel command. It provides constant monitoring on fwaccel output. This extension is useful for acceleration statistics display.

Example: fwaccel_m stats -p

Will constantly monitor the reasons for traffic, which was forwarded from Performance Pack to firewall

fw, fw6

Description:

When invoked from gclish, fw/fw6 commands are global wrappers to the known fw/fw6 command.

fw/fw6 are, for most parameters, comparison type global commands which shows unified information from all SGMs.
Example:

gdual7-t43-ch02-02 > fw ctl

-*. 6 blades: 1_01 1_02 1_03 2_01 2_02 2_03 -*

Usage: fw ctl command args...

Commands: install, uninstall, pstat, iflist, arp, debug, kdebug, bench

chain, conn

fw dbgfile:

Description:

This command is used for easy debugging of the system

fw dbgfile collect collects firewall debugging information (fw ctl debug). User needs to stop its collection manually - by writing stop.

fw dbgfile view shows the collected debugging information

Usage: fw [gexec-flags] dbgfile [collect | view] [fw ctl debug options]

collect - collects debugging information, runs until receiving "stop" command from the user
view - view collected information

Examples:

Debug collection: fw dbgfile collect [-buf BUF_SIZE] -f FILE [FLAGS]

FILE - file to collect the debug information to, full path should be provided
FLAGS - debug flags

For example: fw dbgfile collect -f /home/admin/temp.dbg -buf 2300 -m kiss + pmdump +m fw + xlate

Debug viewing: fw dbgfile view FILE

FILE - file containing debug information collected by the collect option, full path should be provided

For example: fw dbgfile view /home/admin/temp.dbg
OS global commands

**Description**
The global commands are utilities to run certain commands on multiple SGMs. This document is dealing with Operating System related commands, those utilities are mostly an extended wrapper to known UNIX commands (like ls, cp, tcpdump...).

The list of available commands is: arp, cat, cp, dmesg, ethtool, ls, md5sum, mv, netstat, reboot, tail, tcpdump asg_ifconfig, asg_ifconfig_m, top.

<table>
<thead>
<tr>
<th>gclish name</th>
<th>bash name</th>
</tr>
</thead>
<tbody>
<tr>
<td>arp</td>
<td>g_arp</td>
</tr>
<tr>
<td>cat</td>
<td>g_cat</td>
</tr>
<tr>
<td>cp</td>
<td>g_cp</td>
</tr>
<tr>
<td>dmesg</td>
<td>g_dmesg</td>
</tr>
<tr>
<td>ethtool</td>
<td>g_ethtool</td>
</tr>
<tr>
<td>ls</td>
<td>g_ls</td>
</tr>
<tr>
<td>md5sum</td>
<td>g_md5sum</td>
</tr>
<tr>
<td>Mv</td>
<td>g_mv</td>
</tr>
<tr>
<td>Netstat</td>
<td>g_netstat</td>
</tr>
<tr>
<td>Reboot</td>
<td>g_reboot</td>
</tr>
<tr>
<td>Tail</td>
<td>g_tail</td>
</tr>
<tr>
<td>Tcpdump</td>
<td>g_tcpdump</td>
</tr>
<tr>
<td>asg_ifconfig</td>
<td>asg_ifconfig</td>
</tr>
<tr>
<td>asg_ifconfig_m</td>
<td>asg_ifconfig_m</td>
</tr>
<tr>
<td>top</td>
<td>g_top</td>
</tr>
</tbody>
</table>

Other relevant documents may include “CP global commands” and “General commands”.

**Syntax**
```
<global_command> [global_command-flags]
[native_command_args]
```

**Where**
- `<global command>` is a general name for a command, available `<global command>` commands were shown in the description section, other available commands are listed in the documents “CP global commands” and “General commands”.
- Some command comes with the suffix “_m” which notes that this command is used under the “watch” command and hence in “monitor mode”. one example is asg_ifconfig_m
- `[global_command-flags]` are optional flags that determine on which SGMs the command would run. The default behavior is to run on all up SGMs. Optional flags:
  - `-b` SGMs: in one of the following formats
    - A list of comma separated SGMs ids. e.g 1_1,1_4
    - A range of SGM IDs e.g 1_1-1_4
    - A list of SGMs IDs and ranges e.g 1_1,1_3-1_10,1_11
    - all
    - chassis1 – all SGMs on chassis 1
    - chassis2 – all SGMs on chassis 2
    - chassis_active – all SGMs on active chassis
  - `-l` : Execute only on local blade
-r : Execute only on remote blades
-a : Force execution on blades (incl. down SGMs)

One or more flags may be specified, however –l and –r flags should not be specified together.

[native_command args] are optional argument relevant for the running command. For example if the command is the global extension of the UNIX command “ls” then the [cmd arguments] would be the command arguments of the ls command, for example a directory with flags: “/var/log -lrt”
Global command Families:

simple
Utilities of this family will run the command on all selected SGMs and returns the output as is.

Example

```bash
> arp
1_01:
   Address                HWtype  HWaddress             Flags Mask          Iface
   192.0.2.2                ether   00:1C:7F:02:04:FE   C         Sync
   172.23.9.28              ether   00:14:22:09:D2:22   C         eth1-Mgmt4
1_02:
   Address                HWtype  HWaddress             Flags Mask          Iface
   192.0.2.3                ether   00:1C:7F:03:04:FE   C         Sync
1_03:
   Address                HWtype  HWaddress             Flags Mask          Iface
   192.0.2.1                ether   00:1C:7F:01:04:FE   C         Sync
   172.23.9.28              ether   00:14:22:09:D2:22   C         eth1-Mgmt4
```

Comparison
Utilities of this family will run the command on the selected SGM and will try to unify outputs from different SGMs.

Example:

```bash
> md5sum /opt/CPsuite-R75/fw1/modules/fwkern.conf
-*. 6 blades: 1_01 1_02 1_03 2_01 2_02 2_03 -*.
0a3a446b63831e7815f49fd7949b77815f49fd7949b7 /opt/CPsuite-R75/fw1/modules/fwkern.conf
```

Streaming
Utilities of this family will run the command on the selected SGMs in a streaming mode.

Example:

```bash
gdual7-143-ch02-02 > tcpdump -nnni bond1
[1_01]tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
[1_01]listening on bond1, link-type EN10MB (Ethernet), capture size 96 bytes
[1_02]tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
[1_02]listening on bond1, link-type EN10MB (Ethernet), capture size 96 bytes
[1_03]tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
[1_03]listening on bond1, link-type EN10MB (Ethernet), capture size 96 bytes
[2_03]tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
[2_03]listening on bond1, link-type EN10MB (Ethernet), capture size 96 bytes
[2_02]tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
[2_02]listening on bond1, link-type EN10MB (Ethernet), capture size 96 bytes
[2_02]19:23:38.547545 5c:26:0a:9f:07:21 > 01:80:c2:00:00:0e, ethertype Unknown (0x88cc), length 60:
```
Examples:

Running global ls from gclish on SGMs 1_1,1_2,1_3,2_1:
> ls -b 1_1-1_3,2_1 /var/
  `-4 blades: 1_01 1_02 1_03 2_01 `-`
  CPbackup ace crash lib log opt run suroot
  CPSnapshot cache empty lock mail preserve spool tmp
(note the aggregated output)

Running global ls from bash
[Expert@61K]# g_ls /var/
  `-6 blades: 1_01 1_02 1_03 2_01 2_02 2_03 `-`
  CPbackup ace crash lib log opt run suroot
  CPSnapshot cache empty lock mail preserve spool tmp

Global top Syntax:
Global top is a utility for viewing UNIX top output for multiple SGMs.

The global top relies on the user configuration for the local top utility; The global command will use the
local SGM configuration file for configuring the output on the remote SGMs.

Usage:
```
top [local] [-f [-o filename] [-n niter] [-s filename] -h] [global command-flags] [top command line arguments]
```

How to manage g_top display

Top uses a configuration file to manage output display; top by default will copy and use this
configuration file from the local blade (usually located under ~/.toprc). This file will be copied to all SGMs
and will be used when calling top.

To manage g_top display:
1. Run local top (from shell) and set the desired display view
2. Save configuration (shift+w)
3. Run global top

local mode

it is also possible for each blade to display output using its own local configuration file
simply run "top local"
How to send output to a file

at times it is more convenient to send g_top output to a file, for example when there are more blades
then the screen can handle; to enable file mode use the -f flag.

output file

in file mode the output top will be sent to a file (default: /var/log/gtop.<time>). Use --o flag to specify a
different file to save in.

number of iterations

By default top will perform 1 iteration in file mode, use --n to specify a different number

Showing output file

Use "top --s <filename>" to show the content of file <filename>

---

Global Commands Generated by CMM

**Description**  
CMM is in charge of managing and controlling chassis components. Among other, it is capable of powering on and off SGMs and SSMs. User needs to power on/off SGMs in severe situations, for example, when SGM cannot be accessed via Sync interface (in this case, simple reboot command will not suffice).

There are three commands, which control SGMs power from CMM:

1. asg_reboot <global command-flags> – power off and on SGMs
2. asg_hard_shutdown <global command-flags> – power off SGMs
3. asg_hard_start <global command-flags> – power on SGMs

Global commands-flags are described under OS Global commands section. All commands are available from both gclish and Expert shell.

**Example**

> asg_reboot -b 1_03,2_05
You are about to perform hard reboot on blades: 1_03,2_05
It might cause performance hit for a period of time

Are you sure? (Y - yes, any other key - no) Y

Hard reboot requires auditing
Enter your full name: User1
Enter reason for hard reboot [Maintenance]:
WARNING: Hard reboot on blades: 1_03,2_05, User: User1, Reason: Maintenance

Rebooting blades: 1_03,2_05

**Note**

In order to run these commands on SGMs on remote chassis, at least one SGM must be UP and running on the remote chassis.

For instructions on how to restart SSM from CMM, refer to asg_chassis_ctrl section.

---

Configuring a Chassis as Up or Down (asg chassis_admin)

**Description**  
Administer the chassis in a dual-chassis deployment. This command uses administrative privileges to turn a chassis on or off. The command takes a chassis offline (down) or puts a chassis online (up).

When a Chassis is down:

- Backup connections on Chassis Security Gateway Modules are lost
New connections are not synced with the chassis that is down

**Syntax**

```
asg chassis_admin -c <chassis_id> <down|up>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>chassis_id</td>
<td>ID of one chassis to be modified (1 / 2)</td>
</tr>
<tr>
<td>down</td>
<td>up</td>
</tr>
</tbody>
</table>

**Example**

```
asg chassis_admin -c 2 down
```

**Output**

```
You are about to perform chassis_admin down on chassis: 2
Are you sure? (Y - yes, any other key - no) y
Chassis_admin down requires auditing
Enter your full name: John
Enter reason for chassis_admin down [Maintenance]: test
WARNING: Chassis_admin down on chassis: 2, User: John, Reason: test
Chassis 2 is going DOWN...
Chassis 2 state is DOWN
```

**Comments**

- This command is audited. (asg log audit)
- The chassis state can be confirmed by running: `asg stat /monitor`

**Note** - In a two-chassis deployment, changing the chassis state to DOWN degrades system performance.

### Configuring SGMs (asg_blade_config)

**Description**

Use the `asg_blade_config` command for administrative actions such as:

- Pulling the configuration from other (remote) SGMs
- Changing the sync start IP address
- Resetting the system uptime
- Fetching a policy from the Security Management server

**Syntax**

```
asg_blade_config [pull_config]| full_sync <ip_addr> | set_sync_start_ip <start_ip>| reset_uptime | reset_uptime_user | get_smo_ip |is_in_security_group|is_in_pull_conf_group |config fetch_smc]
```

**Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pull_config</td>
<td>Pulls (clones) the configuration from other SGMs (&quot;Manual Policy and Configuration Cloning&quot; on page Error! Bookmark not defined.).</td>
</tr>
<tr>
<td>full_sync</td>
<td>Runs full sync from a remote SGM. The &lt;ip_addr&gt; is the Sync IP of the remote SGM. The full Sync process synchronizes kernel tables between SGMs.</td>
</tr>
</tbody>
</table>
set_sync_start_ip  | Changes the Sync start IP address from the local SGM to the specified address.
reset_uptime  | Resets the system uptime on all SGMs to the current time.
reset_uptime_user  | An interactive command that resets the uptime for all SGMs to a user configured time.
get smo_ip  | Returns the sync IP address of the Single Management Object defined in SmartDashboard. This address is not shown in SmartDashboard.
is_in_security_group  | Checks whether the local SGM is in the security group.
is_in_pull_conf_group  | Check whether the local SGM is in the Pulling Configuration Group (if not, the SGM won’t pull configuration and policy)
cfg fetch_sm  | Fetches the policy from the Security Management server, and distributes it to all SGMs.

**Troubleshooting asg blade_config**

To troubleshoot problems associated with the `asg blade_config` command, examine the logs stored at: `/var/log/blade_config`. For example, if the SGM unexpectedly reboots, you can search the log file for the word `reboot` to learn why.

**System Monitor Daemon (asg system monitor)**

**Description**

By running a series of verification tests from a specified list, the System Monitor Daemon (SMD) makes sure different features of the 61000 Security Systems are working correctly. SMD logs the verification test results to one of two files:

- `/var/log/smd.log`
- `/var/log/smd_smo.log`

**Note** - To see test results, run: `asg log smd`

When enabled, SMD has two modes:

- Enforce
- Monitor Only (non-enforce)

<table>
<thead>
<tr>
<th>SMD Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enforce</td>
<td>Runs verification tests and logs the result. If a feature fails a verification test, the SMD triggers a pre-defined set of actions that attempt to correct the problem. For example, SMD:</td>
</tr>
<tr>
<td></td>
<td>- Reboots a <code>&lt;tp box blade&gt;</code> that is unexpectedly down.</td>
</tr>
<tr>
<td></td>
<td>- Sends SMS, SNMP trap, or email alerts as configured by the <code>asg alert</code> utility (see &quot;Configuring Alerts for SGM and Chassis Events (asg alert)&quot;)</td>
</tr>
<tr>
<td></td>
<td>Corrective actions are specified in: <code>/etc/smd_user.conf</code>.</td>
</tr>
<tr>
<td></td>
<td>Each test has test identifier. For example <code>check_blade_down</code> tests whether the SGM state is DOWN.</td>
</tr>
<tr>
<td>Monitor Only</td>
<td>Runs verification tests and logs the result. But does not take the corrective actions specified in: <code>/etc/smd_user.conf</code>.</td>
</tr>
</tbody>
</table>
To configure verification tests for SMD:

1. Run: `asg_system_monitor config`.
The primary SMD menu opens.
2. Select one of these two options:
   a) **Full Configuration wizard**
   b) **Edit Configuration**

### Full Configuration Wizard

Select **Full Configuration Wizard** to set up verification tests for the first time. The SMD Configuration wizard opens showing a list of supported verification tests:

<table>
<thead>
<tr>
<th>Verification:</th>
<th>Tests if:</th>
<th>Identifier in smd_user.conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blade Down</td>
<td>An SGM or range of SGMs state is DOWN</td>
<td>check_blade_down</td>
</tr>
<tr>
<td>Blade Admin Down</td>
<td>An SGM's administrative state set using the <code>asg sgm_admin</code> command is DOWN</td>
<td>check_blade_admin_down</td>
</tr>
<tr>
<td>SecureXL Down</td>
<td>SxureXL's state set as DOWN</td>
<td>check_sxl_down</td>
</tr>
<tr>
<td>SecureXL Admin Down</td>
<td>SxureXL's administrative state has been set as down</td>
<td>check_sxl_admin_down</td>
</tr>
<tr>
<td>Configuration Consistency</td>
<td>The configuration on the tested SGM is identical with the configuration on the SMO SGM.</td>
<td>check_configuration_consistency</td>
</tr>
<tr>
<td></td>
<td>• The files listed in: <code>/etc/xfer_file_list</code> must be identical on both SGMs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• The <code>asg config show</code> command when run on the tested SGM or the SMO must report the same configuration</td>
<td></td>
</tr>
<tr>
<td>DxI Configuration Consistency</td>
<td>The md5 checksum of the dxl configuration on the local SGM is identical to the md5 checksum of the dxl configuration calculated on the SMO SGM.</td>
<td>check_dxl_configuration_consistency</td>
</tr>
<tr>
<td>DxI Consistency with SSM</td>
<td>Distribution is consistent between SGMs and the SSM. The SSM is distributing data equally between the SGMs.</td>
<td>check_dxl_consistency_with_ssm</td>
</tr>
<tr>
<td>Debug Flags Enabled</td>
<td>One of these debug flags are enabled:</td>
<td>check_debug_flags</td>
</tr>
<tr>
<td></td>
<td>• error</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• info</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• debug</td>
<td></td>
</tr>
<tr>
<td>tcpdump Enabled</td>
<td>The tcpdump utility is running</td>
<td>check_tcpdump_running</td>
</tr>
<tr>
<td>fw monitor Enabled</td>
<td>The fw monitor utility is running</td>
<td>check_fw_monitor_running</td>
</tr>
</tbody>
</table>
### System Configuration

#### Verification:

<table>
<thead>
<tr>
<th>Verification</th>
<th>Tests if</th>
<th>Identifier in smd_user.conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date and Time Configuration</td>
<td>The date and time on the tested SGM and on the SMO SGM differ by more than 5 minutes.</td>
<td>check_date_time_configuration</td>
</tr>
<tr>
<td>SW Version Consistency</td>
<td>Checks Operating system versions, Firewall versions, and other software products.</td>
<td>check_sw_version_setup</td>
</tr>
<tr>
<td>License Validity</td>
<td>Tests whether the license is up to date</td>
<td>check_valid_license</td>
</tr>
</tbody>
</table>

3. Enable all the verification tests or configure each test manually.
4. When prompted, configure:
   - Monitor Only Mode
   - A List of Blades (SGMs)
   - Debug Level (error/info/debug)
   - SMD as enabled

Changes are applied immediately.

### Edit Configuration

Select **Edit Configuration** to edit an existing set of verification tests. Select one of these options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verification tests</td>
<td>Shows a list of verification tests</td>
</tr>
<tr>
<td>Monitor Only Mode</td>
<td>Switches the SMD mode from the (default) enforce to monitor only.</td>
</tr>
<tr>
<td>List of Blades</td>
<td>Shows a list of blades subject to SMD verification tests</td>
</tr>
<tr>
<td>Debug Level</td>
<td>Sets a debug level: error, info, debug.</td>
</tr>
</tbody>
</table>

**To configure an SMD mode:**

5. Run: `asg_system_monitor config`.
   The primary SMD menu opens.
6. Select **Edit Configuration**.
7. Select **Monitor Only Mode**.
   - When promoted, select `y` for **Monitor Only Mode**
   - Select `n` for **Enforce** mode

**Note** - To enable or disable SMD:
   - Run: `asg_system_monitor config`.
   - Select **Enable SMD** or **Disable SMD**.

**To enable or disable an SMD verification test:**

8. Run: `asg_system_monitor config`.
   The primary SMD menu opens.
9. Select **Edit Configuration**.
10. Select **Verification Tests**.
    The Verification Tests menu opens.
11. Select one of the verification tests.
12. Enter `y` or `n` to enable or disable the verification.

**To show the existing SMD configuration:**

13. Run: `asg_system_monitor config`.
The primary SMD menu opens. 

14. Select **Show Configuration**. 

15. A list of verification tests and their configured status shows:
   - **y** means the test is enabled 
   - **n** means the test is disabled
   - Other configurable parameters show along with their values:
     - blades_list=a
     - debug_level=error
     - smd_enabled=y
     - monitor_only_mode=y

**To Show SMD logs:**

SMD logs to two different files:

<table>
<thead>
<tr>
<th>Log file</th>
<th>Location</th>
<th>Max size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smd.log</td>
<td>/var/log/</td>
<td>1MB</td>
<td>- Created on each SGM. &lt;br&gt;- Contains the results of each test verification and the result of any corrective action. &lt;br&gt;- When file the size of smd.log becomes larger than 1MB, it is renamed smd.log.1 and a new smd.log opened. Up to 2 log files of this type can be opened.</td>
</tr>
<tr>
<td>smd_smo.log</td>
<td>/var/log/</td>
<td>500MB</td>
<td>- This log is created on the SMO SGM only. &lt;br&gt;- Contains data about all the activities monitored by SMD. &lt;br&gt;- The log level can be modified using <strong>asg_system_monitor config &gt; Edit Configuration &gt; Debug Level</strong>. Select error, info or debug (default). &lt;br&gt;- When file the size of smd_smo.log becomes larger than 500MB, it is renamed smd_smo.log.1 and a new smd_smo.log opened. Up to 5 log files of this type can be opened. &lt;br&gt;- smd_smo.log is created on all SGMs that have functioned as the SMO SGM.</td>
</tr>
</tbody>
</table>

To see the data collected from these logs, run: **asg log smd**.

<table>
<thead>
<tr>
<th>Log command and options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asg log smd -f fail</td>
<td>Shows validation tests that failed.</td>
</tr>
<tr>
<td>asg log smd -f fix</td>
<td>Shows corrective actions taken by the SMD in enforce mode.</td>
</tr>
<tr>
<td>asg log smd -tail 10</td>
<td>Shows the last 10 logs from each SGM.</td>
</tr>
</tbody>
</table>
Log command and options | Description
---|---
watch asg log smd -tail 5 | Periodically shows the last 5 logs from each SGM.
cat /var/log/smd_smo.log | This log is mostly used for monitoring SGMs state (e.g., UP, DOWN and reboot). For examples, to find out when SGM 1_01 was DOWN run:
cat /var/log/smd_smo.log | grep 1_01 | grep DOWN
To find out when SGM 2_03 was UP run:
cat /var/log/smd_smo.log | grep 2_03 | grep UP
To confirm if SGM 2_08 has been rebooted by the SMD, run:
cat /var/log/smd_smo.log | grep 2_08 | grep reboot

Configuring Security Gateway Modules as Up or Down (asg sgm_admin)

Description: Administer the Security Gateway Modules (blades). Administratively turn the blades on and off.

Syntax: `asg sgm_admin -b blade_string <up|down> [-p]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>blade_string</td>
<td>List of Security Gateway Modules. For example:</td>
</tr>
<tr>
<td></td>
<td>1_01                          Chassis 1 SGM 1</td>
</tr>
<tr>
<td></td>
<td>1_03-1_05                     Chassis 1 SGMs 3, 4 and 5.</td>
</tr>
<tr>
<td></td>
<td>1_01,1_03-1_05                Combination of previous two items</td>
</tr>
<tr>
<td></td>
<td>all                           All SGMs (including chassis 2, if applicable)</td>
</tr>
<tr>
<td></td>
<td>chassis1                      All SGMs in Chassis 1</td>
</tr>
<tr>
<td></td>
<td>chassis2                      All SGMs in chassis 2</td>
</tr>
<tr>
<td></td>
<td>chassis_active                All SGMs in the active chassis</td>
</tr>
</tbody>
</table>

- **p**: Persistent. The setting is kept after reboot.
- **h**: Display usage

Example: `asg sgm_admin -b 2_03 up -p`
### Description
Administer the Security Gateway Modules (blades). Administratively turn the blades on and off.

<table>
<thead>
<tr>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>You are about to perform blade_admin up on blades: 2_03</td>
</tr>
<tr>
<td>Are you sure? (Y - yes, any other key - no) y</td>
</tr>
<tr>
<td>Blade_admin up requires auditing</td>
</tr>
<tr>
<td>Enter your full name: Fred</td>
</tr>
<tr>
<td>Enter reason for blade_admin up [Maintenance]: test</td>
</tr>
<tr>
<td>WARNING: Blade_admin up on blades: 2_03, User: Fred, Reason: test</td>
</tr>
<tr>
<td>Performing blade_admin up on blades: 2_03</td>
</tr>
<tr>
<td>[2_03]Setting blade to normal operation ...</td>
</tr>
<tr>
<td>[2_03]pulling configuration from: 192.0.2.16 (may take few seconds)</td>
</tr>
<tr>
<td>[2_03]Blade current state is ACTIVE</td>
</tr>
</tbody>
</table>

### Comments
- gclish commands do not affect it.
- Traffic is not forwarded to this blade.
- Running asg stat shows the blade is DOWN (admin).

When a blade is administratively down:
- gclish commands do not affect it.
- Traffic is not forwarded to this blade.
- Running `asg stat` shows the blade is DOWN (admin).

When a blade is brought administratively up, the blade imports the configuration from one of the up blades. This makes sure that the system configuration is consistent.

This command is audited. Auditing makes it possible to maintain a log of critical changes made in the system. To show audited activities, run the `asg log audit` command.

This command is useful for debugging. However, we do not recommend using it in production environments because it degrades system performance.

### Time synchronization from NTP server (asg_ntp_sync_config)

#### Description
Blades now can be configured to synchronize their time with NTP server running on the network. This is achieved by periodically performing manual NTP time update by running the command `ntpd -u`. The time on the CMM is also updated.

#### Syntax
```
asg_ntp_sync_config [set <primary | secondary> <NTP Server IP | hostname> [ -v <version> ] [ -r <refresh timeout> ] | show | disable | enable | delete <primary | secondary> ]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTP Server IP</td>
<td>NTP server IP Address or hostname</td>
</tr>
<tr>
<td>version</td>
<td>Server version of the NTP Service</td>
</tr>
<tr>
<td>If NTP version is not specified, NTPv4 will be used</td>
<td></td>
</tr>
</tbody>
</table>
**Backup and Restore (backup_system)**

**Description**

Use this command to:

- **Save a network configuration and security policy**

This saves the SGM configuration and policy to a `.tgz` file and copies it to the other SGMs. The OS database is backed up and the files listed in: `/etc/xfer_file_list`.

An initial backup file (initial.tgz) that contains policy and configuration settings is also automatically created after running the 61000 Security System setup wizard.

- **Restore a saved network configuration and security policy**

The specified configuration and policy backup file is copied and applied to all SGMs in the system.

- **Note** -

  - During the restore procedure, you can select whether to restore:
    - Only the network configuration
    - Network configuration and security policy.

  **Warning:** After reverting to a backed-up policy, SmartDashboard no longer reflects the actual policy settings on the gateway.

  - The `backup_system` command is only available from the bash shell.

  - After restoring a configuration and policy, all SGMs must be rebooted

(Run: `g_reboot -b all`)

---

<table>
<thead>
<tr>
<th>refresh timeout</th>
<th>Timeout in seconds between refresh. If Refresh Timeout is not specified, it defaults to 5 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>show</td>
<td>Show NTP Server configuration</td>
</tr>
<tr>
<td>disable</td>
<td>Disable NTP service</td>
</tr>
<tr>
<td>enable</td>
<td>Enable NTP service</td>
</tr>
<tr>
<td>delete &lt;primary</td>
<td>Delete primary/secondary NTP configuration</td>
</tr>
<tr>
<td>secondary&gt;</td>
<td></td>
</tr>
<tr>
<td>set &lt;primary</td>
<td>Set NTP server as primary/secondary server</td>
</tr>
<tr>
<td>secondary&gt;</td>
<td>Note: This option relevant for version R75.051 and above</td>
</tr>
</tbody>
</table>

**Notes:**

The command updates all blades with the NTP server IP address. Each **Refresh Timeout** seconds, the local time is updated on each blade by running the command 'ntpdate -u'. If timeout is less than 300 seconds (5 minutes), the time on the CMM is updated no more than each 5 minutes.

**Validation**

- Execute `show time` from gclish, validate same time on all SGMs
- Execute `tcpdump` on port 123/UDP on the relevant interface and verify that all SGMs initiate NTP connections
Backup and Restore (backup_system)

Syntax  
backup_system [backup|backup <filename>] | show |restore |[restore <file_path>]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>backup</td>
<td>Creates a backup file with a default name in /var/CPbackup/asg_backup/</td>
</tr>
<tr>
<td>backup &lt;file_name&gt;</td>
<td>Creates a backup file with a unique name in /var/CPbackup/asg_backup/</td>
</tr>
<tr>
<td>show</td>
<td>Shows backup files</td>
</tr>
<tr>
<td>restore</td>
<td>Restores a backup file from in /var/CPbackup/asg_backup/</td>
</tr>
<tr>
<td>restore &lt;file_path&gt;</td>
<td>Restore a backup file from a specified path</td>
</tr>
</tbody>
</table>

Example:

```bash
> backup_system restore
Backup files:
-------------
1) initial.tgz
2) ipv6.tgz
3) normal.tgz
Please select file
>1
  copying /var/CPbackup/asg_backup/initial.tgz to all blades
Would you like to restore policy in addition to configuration? y/n [n]
>y
  copying /var/CPbackup/asg_backup/initial.policy.tgz to all blades
Would you like to backup your system now? y/n [y]
>n
  extracting file initial.tgz
  extracting file initial.policy.tgz
restore completed successfully, please reboot all blades
> g_reboot -b all
```
High Availability

Configuring Chassis High Availability

Chassis HA mechanism allows configuration of two chassis’ (dual box). Only one Chassis process the traffic, this is the chassis which is in state ‘Active’, while the other chassis is in ‘Standby’ state. The chassis which is in ‘Standby’ state is synced with the ‘Active’ chassis so traffic will survive events of Chassis failover.

Each Chassis calculates its own grade. Grade calculation is based on monitoring the following units in the system: SGM, ports, and different HW Sensors (fans, SSMs, CMMs and power supplies). Refer to “Using the set chassis high-availability factors command” for detailed explanation for assigning weights for different factors.

The ‘Active’ chassis is selected to be the chassis with the higher grade. Whenever the other chassis’ grade becomes higher in at least the minimum grade gap for failover, chassis failover occurs. Refer to the section “Setting the minimum gap failover”.

Each chassis has unique MAC value assigned to its ports. These MACs are different between two Chassis of the same 61000 Security System setup. Chassis failover event is performed by sending GARPs packets for each interface in a similar manner as failover event performed by ClusterXL. Please refer to the section “GARP chunk mechanism” for detailed information regarding the GARP sent during failover.

Chassis High Availability section in gclich enable to configure different parameters like: Chassis HA grade factors, failover grade difference for failover, Failover freeze interval, ports factor and Chassis HA Active-Up or Primary Up mode.

Synchronizing Clusters on a Wide Area Network

The synchronization network can be spread over remote sites, which makes it easier to deploy geographically distributed clustering. There are two limitations to this capability:

1. The synchronization network must guarantee no more than 100ms latency and no more than 5% packet loss.
2. The synchronization network may only include switches and hubs. No routers are allowed on the synchronization network, because routers drop Cluster Control Protocol packets.

Using the set chassis high-availability factors command

Each component in a chassis, such as a fan or port, has a “weight”. The weight is a numerical value which reflects the component importance level. Ports might be more important than fans and receive a higher value or a greater weight. The chassis grade is the sum of all these component weights. In a high-availability dual-chassis deployment, the chassis with the higher grade becomes active and processes traffic. The grade of each component = (Unit Weight) X (Number of UP components)

To see the weight of each component, run: asg stat -v.

Use the set chassis high-availability factors command to configure a component's weight.

Syntax

```
set chassis high-availability factors [SGM <factor> | port high <factor> | port standard <factor> | sensor cmm <factor> | sensor fans <factor> | sensor power_supplies <factor> | sensor ssm <factor> | pnote pingable_hosts <factor>]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGM</td>
<td>Sets the weight factor for an SGM</td>
</tr>
<tr>
<td></td>
<td>The weight factor must be between 0 and 1000</td>
</tr>
<tr>
<td></td>
<td>Example: set chassis high-availability factors sgm 100</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>port high</td>
<td>- A port has one of two grades: high or standard. This parameter sets a weight factor for the high grade</td>
</tr>
<tr>
<td></td>
<td>- The weight factor must be between 0 and 1000</td>
</tr>
<tr>
<td></td>
<td>- Example: set chassis high-availability factors Port high 70</td>
</tr>
<tr>
<td></td>
<td>This means that ports set to high grade have a weight of 70.</td>
</tr>
<tr>
<td>port standard</td>
<td>- A port has one of two grades: high or standard. This parameter sets a weight factor for the standard grade</td>
</tr>
<tr>
<td></td>
<td>- The weight factor must be between 0 and 1000</td>
</tr>
<tr>
<td></td>
<td>- Example: set chassis high-availability factors Port standard 50</td>
</tr>
<tr>
<td></td>
<td>This means that ports set to standard grade have a weight of 50.</td>
</tr>
<tr>
<td>Sensor CMMs</td>
<td>- Sets a weight factor for CMMs</td>
</tr>
<tr>
<td></td>
<td>- The weight factor must be between 0 and 99</td>
</tr>
<tr>
<td></td>
<td>- Example: set chassis high-availability factors sensor cmm 40</td>
</tr>
<tr>
<td>Sensor fans</td>
<td>- Sets a weight factor for fan units</td>
</tr>
<tr>
<td></td>
<td>- The weight factor must be between 0 and 99</td>
</tr>
<tr>
<td></td>
<td>- Example: set chassis high-availability factors sensor fans 30</td>
</tr>
<tr>
<td>Sensor Power Supply units</td>
<td>- Sets a weight factor for power supply units</td>
</tr>
<tr>
<td></td>
<td>- The weight factor must be between 0 and 99</td>
</tr>
<tr>
<td></td>
<td>- Example: set chassis high-availability factors sensor power_supplies 20</td>
</tr>
<tr>
<td>SSMs Sensor</td>
<td>- Sets a weight factor for SSMs</td>
</tr>
<tr>
<td></td>
<td>- The weight factor must be between 0 and 99</td>
</tr>
<tr>
<td></td>
<td>- Example: set chassis high-availability factors sensor ssm 45</td>
</tr>
<tr>
<td>pnote pingable_hosts</td>
<td>- Sets a weight factor for pingable hosts, a way of making sure ports are properly connected to their hosts.</td>
</tr>
<tr>
<td></td>
<td>- The weight factor must be between 0 and 99</td>
</tr>
<tr>
<td></td>
<td>- Example: set chassis high-availability factors pnote pingable_hosts 99</td>
</tr>
</tbody>
</table>

**Set the primary Chassis**

Use the `set chassis high-availability primary-chassis <0-2>` command to define which chassis is primary. If both chassis have the same grade, the chassis defined as primary using this command becomes active.

**Syntax:** `set chassis high-availability primary-chassis <0-2>`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No primary Chassis (Active Up Mode)</td>
</tr>
<tr>
<td></td>
<td>In this mode, the chassis which is UP stays up until the other chassis gets a higher grade.</td>
</tr>
</tbody>
</table>
## Setting the minimum gap failover

Use the `set chassis high-availability failover` command to set the minimum grade gap for chassis failover.

**Syntax:** `set chassis high-availability failover <1-1000>

## Setting the freeze interval

Use the `set chassis high-availability freeze_interval` command to set a freeze interval. After a failover, the chassis is prevented or frozen from failing over again until the interval expires.

**Syntax:** `set chassis high-availability freeze_interval <1-1000>

**Note:** When running `asg stat` after chassis failover, you will be notified with the freeze time:

```
gilani-chC2-01 > asg stat -v
```

### Chassis Parameters

<table>
<thead>
<tr>
<th>Unit</th>
<th>Chassis 1</th>
<th>Chassis 2</th>
<th>Unit Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGMs</td>
<td>2 / 3</td>
<td>2 / 3</td>
<td>11</td>
</tr>
<tr>
<td>Ports</td>
<td>Standard</td>
<td>2 / 2</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>Port 1</td>
<td>0 / 1</td>
<td>0 / 1</td>
</tr>
<tr>
<td></td>
<td>Other</td>
<td>0 / 0</td>
<td>0 / 0</td>
</tr>
<tr>
<td>Sensors</td>
<td>Fans</td>
<td>4 / 4</td>
<td>4 / 4</td>
</tr>
<tr>
<td></td>
<td>CSMs</td>
<td>2 / 2</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>CMMs</td>
<td>2 / 2</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>Power Supplies</td>
<td>6 / 6</td>
<td>6 / 6</td>
</tr>
<tr>
<td>Chassis Grade</td>
<td>124 / 141</td>
<td>124 / 141</td>
<td>-</td>
</tr>
</tbody>
</table>

**Minimum grade gap for chassis failover:** 11

### Setting port priority (for each port)

Use the `set chassis high-availability port priority` command to set a port priority (high or standard) for each port
**Syntax:** set chassis high-availability port <interface> priority <1-2>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Standard priority</td>
</tr>
<tr>
<td>2</td>
<td>Other priority</td>
</tr>
</tbody>
</table>

Use this command together with the `set chassis high-availability factors port` command.

1. First set the port grade as standard or high.
   For example:
   
   ```
   set chassis high-availability factors port standard 50
   ```
   
   This sets the standard grade at 50.

2. Then decide which ports have the high grade or the standard grade.
   For example:
   
   ```
   set chassis high-availability port eth1-01 priority 2
   ```
   
   This assigns to `eth1-01` the standard port grade.

**Single SSM**

While working with single SSM is supported, this is a highly not recommended state – it is considered as alert state that must be fixed immediately.

When working with dual chassis, failover is recommended upon single SSM failure (default grades will cause this behavior).

**Verification**

Each of the `set` commands has a corresponding `show` command. For example: `set chassis high-availability primary-chassis <0-2>` can be verified by running: `show chassis high-availability primary-chassis`. 
**Chassis HA - Link Preemption Mechanism**

**Description:**

The Link Preemption Mechanism prevents constant chassis fail-over and failback whenever there is interface link flapping.

When you enable this feature, an interface state that changes from down to up, is only considered in the chassis grade if the link state is up for X seconds (default is 10 sec).

**Configuration:**

The Link Preemption Mechanism is enabled by default with a preemption time of 10 seconds. To configure the preemption time, run these commands from gclish:

```bash
> fw ctl set int fwha_ch_if_preempt_time < preemption time >
> update_conf_file fwkern.conf fwha_ch_if_preempt_time=< preemption time >
```

Sample commands that set the preemption time to 20 seconds:

```bash
> fw ctl set int fwha_ch_if_preempt_time 20
> update_conf_file fwkern.conf fwha_ch_if_preempt_time=20
```

**Deactivation:**

To disable Link Preemption Mechanism, run these commands from gclish:

```bash
> fw ctl set int fwha_ch_if_preempt_time 0
> update_conf_file fwkern.conf fwha_ch_if_preempt_time=0
```

**Verification:**

To check the preemption time value, run this command from gclish:

```bash
> fw ctl get int fwha_ch_if_preempt_time
```
**Chassis HA – Sync Lost Mechanism**

**Description:**

The Sync Lost mechanism handles loss of connectivity between two chassis on the Sync network. To prevent the two chassis changing their states to Active, a special SYNC_LOST CCP packet is sent over non-sync interface (the Data Ports and Mgmt interfaces) to the other chassis. The SYNC_LOST CCP packet causes the two chassis to freeze their current state until connectivity between the two chassis is restored. During the Sync Loss, the Standby chassis, does not change its state to Active until it stops hearing the SYNC_LOST packets from the other chassis.

**Configuration:**

Sync Lost mechanism is enabled by default.

To disable Sync Lost Mechanism, run these commands from gclish:

```bash
> fw ctl set int fwha_ch_sync_lost_mechanism_enabled 0
> update_conf_file fwkern.conf fwha_ch_sync_lost_mechanism_enabled=0
```

To enable Sync Lost Mechanism, run these commands from gclish:

```bash
> fw ctl set int fwha_ch_sync_lost_mechanism_enabled 1
> update_conf_file fwkern.conf fwha_ch_sync_lost_mechanism_enabled=1
```

**Verification:**

To check whether the mechanism is enabled:

```bash
> fw ctl get int fwha_ch_sync_lost_mechanism_enabled
```

(1- enabled, 0- disabled)

**Chassis ID Configuration**

When installing and configuring chassis high availability, you must make sure that chassis ID are different before you start to configure the software. Chassis IDs are configured on the CMM and should be <1> for the first chassis and <2> for the second chassis.

**Note:** In case your 61000 Security System is up and running, change the chassis ID on the **Standby Chassis**, hence you will have to perform chassis failover.

**Procedure**

1. Remove the upper CMM from the chassis
2. Log in to the 61000 Security System CMM.
   
   I. Connect the serial cable to the console port on CMM.
   
   II. Connect to the 61000 Security System CMM using a terminal emulation application such as PuTTY.
       • Make sure the Speed (baud rate) is set to 9600.
       • No IP address is necessary.
III. Log in with username and password: admin/admin.

3. Edit the file `/etc/shmm.cfg` using `vi`. And set the correct ID on the line with the string `SHMM_CHASSID`.

```
# vi /etc/shmm.cfg
# -----------------------------------------------------
# Shelf Manager Config file, template
# <<---- '#' for comment
#
#!/bin/bash
SHMM_IP="10.10.11.35"
SHMM_IP2="10.10.12.35"
SHMM_IPMASK="255.255.255.0"
# power budget setup for each slot, by hw_addr
# format: <hw_addr, fru ID, watts>
# or "board", slot, watts>
# SNMP Credential
SNMP_rwuser="asg1"
SNMP_createUser="asg1 MD5 asg1asg1 DES"
# authentication type
# format is: <callback user op admin oem>
# SNMP_authen="0x04 0x04 0x04 0x04 0"
# Chassis ID
SHMM_CHASSID="1"
```

4. Remove from the chassis the lower CMM that just was reconfigured.
5. Insert the upper CMM to the chassis.
6. Run the step 1 - 3 on the upper CMM.
7. Remove the upper CMM from the chassis.
8. Insert both CMMs into the chassis.
9. Mark the chassis and the CMM with correct stickers.
10. This step is mandatory if the Chassis has already been configured (After FTW)
    Do hard reboot to all SGMs by physically removing all SGMs from the chassis and the reinserting them.
Configuring a Unique IP address per Chassis (UIPC)

Description
In dual-chassis deployment:

- A heavy load on the active chassis can prevent you from making a network connection to the SMO SGM and implementing management tasks.
- You may also require direct access to the standby chassis to trouble-shoot a problem, such as an SGM which is down. (You cannot use the SMO SGM to connect to the standby chassis).

These two scenarios can be solved by assigning a unique IP address to each chassis. Assigning a unique IP address to each chassis adds an extra alias IP to the management interfaces on all SGMs in the chassis.

- If there is a high load on the SMO SGM, connect using the unique IP assigned to the standby chassis. The SGMs on the standby chassis are always UP and available to run `gclish` management commands.
- When you need to connect directly to the standby chassis, use the standby chassis’ unique IP.

Note -
- Similar to the SMO mechanism, only one SGM owns the UIPC task
- The UIPC feature is disabled by default
- If the GW isn’t managed from management port, the unique IP can be added to one of the data ports

To add a unique IP per chassis:
In `gclish`, run:

Syntax
```
set chassis id <1|2> general unique_ip <ip_addr>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>chassis id</td>
<td>The chassis ID, 1 or 2</td>
</tr>
<tr>
<td>general unique_ip</td>
<td>An alias IP address on the same network as one of the SGMs interfaces</td>
</tr>
</tbody>
</table>

Output
```
> set chassis id 1 general unique_ip 172.16.6.106
Adding alias IP: 172.16.6.106 to chassis 1
Alias IP was added successfully
```

To remove the unique IP from a chassis:
In `gclish`, run:

Syntax
```
delete chassis id <1|2|all> general unique_ip
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>chassis id</td>
<td>The chassis ID, 1 or 2 or all</td>
</tr>
<tr>
<td>general unique_ip</td>
<td>The alias IP to remove</td>
</tr>
</tbody>
</table>

Output
```
> delete chassis id 1 general unique_ip
Deleting alias IP 172.16.6.136 of chassis 1
Alias IP was deleted successfully
```

Although the UIPC feature is automatically enabled when you run the configuration commands, you can also manually enable or disable it:

- To manually enable UIPC, run:
  `g_fw ctl set int fwha_uipc_enabled 1`
- To manually disable UIPC run:
  `g_fw ctl set int fwha_uipc_enabled 0`
To show the existing UIPC configuration, run:
show chassis id <1|2|all> general unique_ip

asg_sync_manager

Description
The asg_sync_manager enables the user to define its required synchronization level. The synchronization level is a combination of system synchronization settings (e.g. backup connections to standby chassis) and specific rules (e.g. do not sync HTTP connections). Specific rules are referred to as sync exception table. Connections are serially matched against this table.

In addition to the synchronization settings, this utility also controls SecureXL delayed synchronization parameters: when connection is created within SecureXL (from SecureXL template), asg_sync_manager can set the period until it will be synchronized to firewall.

By default, specific sync exception table consists of a single rule, which is not to synchronize DNS traffic.

Key synchronization properties are also displayed in `asg stat -v`

Syntax:

Usage:
The utility is interactive. The following options are available:
### High Availability

**System Configuration**

<table>
<thead>
<tr>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1) Print sync exceptions table | This view displays the sync exception table. Each entry in this table consists of:  
1. <5-tuple, including wild cards>  
2. synchronization mode (none, within chassis only, between chassis only, both within, between chassis and to all SGMs)  
3. SecureXL delayed synchronization value  
In addition, global synchronization values are displayed |
| 2) Add new sync exceptions rule | Add new rule to the sync exceptions table. The user can hit enter at any stage to apply the default value. Specific rules allow the use of wildcards within 5-tuple. New rule will apply for new connections |
| 3) Delete old sync exception rule | Delete rule from the sync exceptions table |
| 4) Set sync between chassis flag on / off | Global system setting: whether to synchronize connections to backup chassis |
| 5) Set sync within local chassis flag on / off | Global system setting: whether to synchronize connections within active chassis |
| 6) Configure sync between chassis blades ratio | Minimal blades ratio between active and backup chassis for synchronization to occur. If the number of UP SGMs in standby chassis is significantly low, compared to active chassis, synchronization might overload them. Default ratio for synchronization is 70% and it can be re-configured here. After configuration, user can also choose to restore default settings |
| 7) Set default delay notifications | Default delayed synchronization setting are divided to HTTP related services (30) and all other services (5). User can reconfigure these settings here. Note that when configuring service delayed synchronization in SmartDashboard it overrides these settings |
| 8) Enable / Disable unicast sync | The user can enable / disable unicast sync (correction layer will be enabled / disabled accordingly) and return to legacy synchronization scheme (synchronize connections to all SGMs). Changing this setting requires reboot of all SGMs |

**Output:**

This is the main menu of the tool:

```
Please choose one of the following:
-----------------------------------
1) Print sync exceptions table
2) Add new sync exceptions rule
3) Delete old sync exception rule
4) Set sync between chassis flag on / off
5) Set sync within local chassis flag on / off
6) Configure sync between chassis blades ratio
7) Set default delay notifications
8) Enable / Disable unicast sync
9) Exit
```

**Example:**

The following example shows how to add rule which limits the synchronization of HTTP traffic, initiated from network 3.3.3.0/24 to network 4.4.4.0/24 to active chassis only:
Enter source IP [0.0.0.0]: >3.3.3.0
Enter source IP mask length [0]: >24
Enter destination IP [0.0.0.0]: >4.4.4.0
Enter destination IP mask length [0]: >24
Enter destination port [0]: >80
Enter IP protocol number (for example: tcp = 6, udp = 17): >6
Enter the sync exception rule [3 - sync to all chassis]:
0 = no sync
1 = sync only to local chassis
2 = sync only to other chassis
3 = sync to all chassis
4 = sync to all SGMs (without Unicast Sync)
>1
Enter delay notification [30 - http, 5 - other]: >
to insert new exception: <3.3.3.0/24, 4.4.4.0/24, 80, 6> sync rule: 1, delay: 5 ? (y/n)

After adding this rule, sync exception table will be as follows:

<table>
<thead>
<tr>
<th>Idx</th>
<th>Source</th>
<th>Mask</th>
<th>Destination</th>
<th>Mask</th>
<th>DPort</th>
<th>Ipp</th>
<th>Sync</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0.0.0.0</td>
<td>0</td>
<td>53</td>
<td>17</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>2)</td>
<td>3.3.3.0</td>
<td>24</td>
<td>4.4.4.0</td>
<td>24</td>
<td>80</td>
<td>6</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

Sync between chassis flag is: on
Sync within chassis flag is: on
Default delays: http - 30, other - 5

Sync Values:

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sync</td>
<td>Delay</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>0 = no sync</td>
<td>The time it takes for connections created from templates to synchronize</td>
</tr>
<tr>
<td>1 = sync only to local chassis</td>
<td></td>
</tr>
<tr>
<td>2 = sync only to other chassis</td>
<td></td>
</tr>
<tr>
<td>3 = sync to all chassis</td>
<td></td>
</tr>
<tr>
<td>4 = sync to all SGMs</td>
<td></td>
</tr>
</tbody>
</table>

Press enter to continue
Monitoring, Logs and Auditing

Redirecting Alerts Messages and Firewall logs to External syslog server (asg_syslog)

Description:

asg_syslog command should be used in order to redirect alert messages and firewall logs to remote syslog servers.

This command allows configuring the following:

- Remote syslog servers either by IPv4 address or by hostname to log all alert messages.
- Remote syslog servers to log FW logs.
- Disable/Enable firewall logs to be sent to the Log Server. (Log Server is configured from SmartDashboard: Right-click gateway object > Edit > Logs and Masters > Log Servers)
- Verify configuration consistency on all SGMs.
- Recover configuration on all SGMs by forcing current SGM configuration on all SGMs.

Command is only available from Expert shell.

Syntax:

To verify /print/ recover configuration:

Usage: asg_syslog < verify | print [ -v ] | recover >

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;verify&gt;</td>
<td>Verify configuration consistency on all SGMs</td>
</tr>
<tr>
<td>&lt;print&gt; [-v]</td>
<td>Print remote syslog servers configuration</td>
</tr>
<tr>
<td>&lt;recover&gt;</td>
<td>Recover configuration files on all SGMs and restart syslog service</td>
</tr>
</tbody>
</table>
Examples:

```bash
# asg_syslog verify

<table>
<thead>
<tr>
<th>Service</th>
<th>Path</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPLog</td>
<td>/etc/syslog_servers_list.conf</td>
<td>Passed</td>
</tr>
<tr>
<td>Alert</td>
<td>/etc/syslog.conf</td>
<td>Passed</td>
</tr>
</tbody>
</table>

Configuration files on all SGMs are identical

# asg_syslog print

<table>
<thead>
<tr>
<th>Service</th>
<th>Server IP</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>5.5.5.5</td>
<td>disable</td>
</tr>
<tr>
<td>alert</td>
<td>6.6.6.6</td>
<td>enable</td>
</tr>
</tbody>
</table>

* Firewall logging is disabled

Syntax:

Configure remote syslog servers for alerts:

Usage: asg_syslog < disable | enable | set | delete > alert < IP address | hostname >

Configure remote syslog server for firewall logs:

Usage: asg_syslog < disable | enable | set [ -s < status > ]| delete > cplog < IP address >

Note: When configuring alert syslog servers, syslog service is being restarted on all SGMs.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;set&gt;</td>
<td>Set remote syslog server</td>
</tr>
<tr>
<td>-s &lt;status&gt;</td>
<td>Set connection with status &lt;enable&gt; or &lt;disable&gt;</td>
</tr>
<tr>
<td>&lt;disable&gt;</td>
<td>Disable sending Firewall logs / alerts to a remote syslog server defined by IP address or host name. Note: disable operation will not remove the configuration. You can enable it again using the ‘enable’ parameter</td>
</tr>
<tr>
<td>&lt;enable&gt;</td>
<td>Enable sending Firewall logs / alerts to a remote syslog server defined by IP address or host name. This parameter can be used after the remote server has been configure ( see ‘set’ parameter)</td>
</tr>
<tr>
<td>&lt;delete&gt;</td>
<td>Delete remote syslog server.</td>
</tr>
<tr>
<td>&lt;ip address</td>
<td>hostname&gt;</td>
</tr>
</tbody>
</table>
Examples:

```
# asg_syslog set alert 5.5.5.5
Writing new configuration
Updating all SGMs with new configuration
Restarting syslog service on all SGMs
syslog alert server 5.5.5.5 configured successfully
```

<table>
<thead>
<tr>
<th>Service</th>
<th>Server IP</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>5.5.5.5</td>
<td>enable</td>
</tr>
</tbody>
</table>

* Firewall logging is disabled

```
# asg_syslog disable alert 5.5.5.5
Updating all SGMs with new configuration
Restarting syslog service on all SGMs
syslog alert server 5.5.5.5 status changed to disable
```

<table>
<thead>
<tr>
<th>Service</th>
<th>Server IP</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>5.5.5.5</td>
<td>disable</td>
</tr>
</tbody>
</table>

* Firewall logging is disabled

```
# asg_syslog set cplog 6.6.6.6
Writing new configuration
Updating all SGMs with new configuration
syslog cplog server 6.6.6.6 configured successfully
```

<table>
<thead>
<tr>
<th>Service</th>
<th>Server IP</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>5.5.5.5</td>
<td>disable</td>
</tr>
<tr>
<td>cplog</td>
<td>6.6.6.6</td>
<td>disable</td>
</tr>
</tbody>
</table>

* Firewall logging is disabled

Syntax:

To Disable/Enable firewall logs to be sent to Firewall log server (i.e. SmartView Tracker):

Usage: asg_syslog < disable | enable > log_server

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;disable&gt;</td>
<td>Enable sending firewall logs to the log server. (log server is configured in Smart Dashboard)</td>
</tr>
<tr>
<td>&lt;enable&gt;</td>
<td>Disable sending firewall logs to the log server. (log server is configured in Smart Dashboard)</td>
</tr>
</tbody>
</table>
Example:

```bash
# asg_syslog disable log_server
# asg_syslog print -v
```

<table>
<thead>
<tr>
<th>Service</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewall logging is disabled</td>
<td></td>
</tr>
</tbody>
</table>

---

## Monitoring Management Interfaces Link State

**Description**

By Default, 61000 Security Systems monitor link state only on data ports (ethX-YZ).

Mgmt Monitor feature enables monitoring link states of Mgmt ports as well.

Both SSM60 and SSM160 Mgmt ports are monitored periodically by snmp. Their link state is distributed between all SGMs and is integrated as part of the Chassis HA grade mechanism.

Once enabled, new type of Ports will be available in `asg stat -v` view (Mgmt), describing link state of the monitored Mgmt interfaces from the Policy topology section, just like “Standard” and “Other” Ports.

Monitored management ports have their part in the Chassis grade mechanism as mentioned, according to their pre-defined factors (default = 11)

In addition, `asg if` utility will consider this feature and if enabled, will also show the link state of Mgmt interfaces according to the feature mechanism.

**Note** - in SSM60, we also need to pre-configure the base Switch to enable snmp-server before enabling the feature itself (see next: SSM60 snmp-server configuration)" after configuring snmp-server, run the following gclish command (same in SSM60 / 160)

**Syntax**

```bash
set chassis high-availability mgmt-monitoring <on | off>
```
ssm60 snmp-server configuration

On each Chassis, login to every SSM base switch address using telnet.

1. Enter ‘enable’ mode
2. Enter ‘configure terminal’ mode
3. Execute the following 5 commands:
   a. `snmp-server enable`
   b. `snmp-server view myview 1.3 included`
   c. `snmp-server group mygroup v3 auth read myview write myview notify myview`
   d. `snmp-server system-name BI_cp`
   e. `snmp-server user asg1 group mygroup v3 auth md5 asg1asg1`
4. exit ‘configure terminal’ mode
5. execute ‘write’ to save configuration

Validating snmp configuration

After configuring all SSM60, validate configuration by running the following command from shell:

```
mgmt_monitor snmp_verify
```
Monitoring, Logs and Auditing

System Configuration

Page 125

[Protected] For public distribution

Mgmt Port Factors

Mgmt Ports are integrated as part of the Chassis HA grade mechanism therefore; setting Mgmt port factors (for all Mgmt ports) are the same as ‘Standard’ or ‘Other’ data ports factors.

- Use the `set chassis high-availability factors port mgmt` command to change mgmt port factors (default = 11)

Log Server Distribution (asg_log_servers)

Description

In SmartDashboard, multiple log servers can be configured per gateway object. In such an environment, the gateway sends its logs to all of its configured log servers. If the gateway object is a 61000 Security Systems appliance (consisting of many SGMs) each SGM will send its logs to all log servers in the configuration. To reduce the load on the log servers, use the `asg_log_servers` command to enable log distribution (load sharing).

When enabled, each SGM sends its logs to one log server only. The decision which Log Server will be assigned to which SGM is done automatically and cannot be defined by the user.

Example

```
Syntax asg_log_servers
Example asg_log_servers
```
Output

> asg log_servers
+---------------------------------------------
| Log Servers Distribution
+---------------------------------------------

Log Servers Distribution Mode: Disabled

Available Log Servers:
* LogServer
* Gaia
* LogServer2

Logs will be sent to all available servers.

Choose one of the following options:

1) Configure Log Servers Distribution mode
2) Exit

>1

+---------------------------------------------
| Log Servers Distribution
+---------------------------------------------

Log Servers Distribution Mode: Disabled

Choose the desired option:

1) Enable Log Servers Distribution mode
2) Disable Log Servers Distribution mode
3) Back

If log server distribution is already enabled, the command shows which log servers are assigned to each SGM:

+---------------------------------------------
| Log Servers Distribution
+---------------------------------------------

Log Servers Distribution Mode: Enabled

Available Log Servers:
* LogServer
* Gaia
* LogServer2

Log Servers Distribution:

<table>
<thead>
<tr>
<th>SGM id</th>
<th>Chassis 1</th>
<th>Chassis 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gaia</td>
<td>Gaia</td>
</tr>
<tr>
<td>2</td>
<td>LogServer2</td>
<td>LogServer2</td>
</tr>
<tr>
<td>3</td>
<td>LogServer</td>
<td>LogServer</td>
</tr>
<tr>
<td>4</td>
<td>Gaia</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>LogServer</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>Gaia</td>
</tr>
<tr>
<td>9</td>
<td>LogServer</td>
<td>LogServer2</td>
</tr>
<tr>
<td>10</td>
<td>Gaia</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>LogServer2</td>
<td>-</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(‘-’ - SGM is not in Security Group)

Choose one of the following options:

1) Configure Log Servers Distribution mode
2) Exit
Note - You cannot configure an SGM to send its logs to a particular log server. Distribution takes place automatically.

Configuring a Dedicated Logging Port

Description
The 61000 Security Systems logging mechanism lets each SGM forward logs directly to a logging server over the SSM's management ports. However, management ports can experience a high load when a large number of logs are forwarded. Load on the SSM management ports can be significantly reduced by:

- Setting up a dedicated SSM port for logging
- Assigning the dedicated logging port to each SGM

To set up a dedicated logging port:
1. Install a log server and create an object for it in SmartDashboard.
2. Connect the log server directly to a management port on the SSM.

Important - Do not use the same port which connects to the Security Management server.

3. In gclish, run the set interface command to configure the port as a dedicated logging port:

   Syntax
   set interface <interface> <ipv4-address> mask-length

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>The interface that connects directly to the log server.</td>
</tr>
<tr>
<td>ipv4-address</td>
<td>IPv4 address of the logging server</td>
</tr>
<tr>
<td>mask-length</td>
<td>mask length</td>
</tr>
</tbody>
</table>

   Example
   set interface eth1-Mgmt2 ipv4-address 2.2.2.10 mask-length 24

   Output
   > set interface eth1-Mgmt2 ipv4-address 2.2.2.10 mask-length 24
     1_01: SUCCESS
     1_02: SUCCESS
     1_03: SUCCESS
     2_01: SUCCESS
     2_02: SUCCESS
     2_03: SUCCESS
     >

   Comments
   - For each SGM, eth1-Mgmt2 is set as a unique logging port
   - 2.2.2.0/24 is the logging server network or leads to the logs server network.

Connecting to the logging server:
1. Open SmartDashboard.
2. Open the Single Management Object (SMO ) for the 61000 Security Systems.
4. Select the dedicated log server.
5. Install a policy.

- The SMO in SmartDashboard makes sure that return traffic from the logging server, such as ACKS, reaches the correct SGM.
- 61000 Security Systems can be configured to send logs to more than one log server. For more on logging servers, see the R75 documentation [http://supportcontent.checkpoint.com/solutions?id=sk58362](http://supportcontent.checkpoint.com/solutions?id=sk58362).

### Command Auditing

Command auditing is a way of:

- Notifying users about critical actions they are about to take
- Obtaining confirmation for critical actions
- Creating forensic logs

If users confirm the action, they are requested to supply their names and a reason for running the command. If the command affects a critical device or a process (Note) a second confirmation may be required.

For example, if you use administrative privileges to change the state of an SGM to DOWN the output looks like this:

```
> asg_sgm_admin -b 2_01 down
You are about to perform sgm_admin down on blades: 2_01
Are you sure? (Y - yes, any other key - no) y
sgm_admin down requires auditing
Enter your full name: John Smith
Enter reason for sgm_admin down [Maintenance]:
WARNING: sgm_admin down on SGM: 2_01, User: John Smith, Reason: Maintenance
```

To view the audit logs, run `asg log audit`:

```
# asg log audit
Aug 01 08:53:45 1_01 WARNING: sgm_admin down on sgm: 1_02, User: susan, Reason: Maintenance
Aug 02 08:54:21 1_01 WARNING: Reboot on blades: 1_01, User: susan, Reason: Maintenance
Aug 04 08:55:33 2_01 WARNING: sgm_admin up on sgm: 1_02, User: susan, Reason: Maintenance
Aug 06 11:48:30 2_01 CRITICAL: sync turn off between chassis on blades: all, User: ms, Reason: Maintenance
Aug 07 11:49:02 2_01 CRITICAL: sync turn on between chassis on blades: all, User: Paul, Reason: increase performance
Aug 08 11:49:17 2_01 CRITICAL: sync turn off within chassis on blades: all, User: Tom, Reason: testing sync
Aug 08 11:49:43 2_01 CRITICAL: sync turn on within chassis on blades: all, User: Peter, Reason: Maintenance
Aug 09 12:38:24 2_01 CRITICAL: Reboot on blades: all, User: ms, Reason: Maintenance
```
Security

VPN

VPN LTE

Description

LTE VPN configuration consists of hundreds or thousands of eNodeB's VPN peers. Each eNodeB has its own IPSec tunnel against the 61K. eNodeB actually encrypts the GTP traffic (GTP is encapsulation over http, etc.) from the mobile clients behind the eNodeB.

Configuration:

Management Configuration

1. LTE requires DPD Hotfix. Contact Checkpoint Support.
2. Configure DPD (management configuration only)
3. Configure Inner packet fragmentation (management configuration only)

Gateway Configuration

1. Run asg_lte_config enable
   Note: if not all blades are in UP state while running this command (e.g., not all blades are present in the chassis) make sure to copy $CPDIR/tmp/.CPprofile.sh from the blade you ran the command on to newly added blades
2. Change distribution mode to General:
   Run from gclish asg dxl dist_mode set and choose General (option 2).
3. Accelerate SCTP
4. Reboot all blades: run from gclish reboot -b all

Inner packet fragmentation

Description

This feature prevents packet fragmentation after encryption. If the encrypted packet exceeds the MTU of the output interface, then the clear packet is pre-fragmented.

We recommended that you use this feature when there are some VPN peers that cannot handle IPSEC fragments.

In Performance Pack, these candidate packets are not fragmented but are forwarded to the Firewall to be processed.

Configuration

Management Configuration

11. Enable the Inner packet fragmentation:
   - Open the GuiDBEdit tool.
   - In "Network Objects", select the required Gateway object.
   - Go to "VPN" attribute.
11. Set the `ipsec_fragment_inner` attribute to true.

12. Install the policy on the Gateway

Gateway Configuration

Gateway side configuration is automatically configured when running the script `asg_lte_config enable` as part of the LTE configuration.

If you are configure this feature without configuring LTE follow these steps:

1. Access the 61000 Security System and run:
   ```
   update_conf_file $PPKDIR/boot/modules/simkern.conf vpn_f2f_for_fragmentation=1
   ```

2. Reboot all blades `[reboot –b all]`

Verification:

Run `tcpdump` to verify that there is no traffic with fragmented IPSec packets.

```
[ tcpdump -i <interface> ‘ip proto 50 and not ip[7] = 0’]
```
DPD

Description

DPD (Dead Peer Detection) detects a malfunctioning tunnel, and re-establishes a tunnel through an available peer. This functionality is essential for an LTE IPsec based solution between the eNodeB and the Security Gateway protecting the EPC. DPD lets you maintain IPsec tunnels to the EPC, and even if the tunnel goes down it is possible to connect through a different peer.

DPD (Dead Peer Detection) is defined in RFC 3706, and is an alternative mechanism that is more scalable solution for detecting dead IPsec peers than the IKE keep-alive. DPD does not send periodic messages to check liveliness of a peer, as the IPsec traffic serves as implicit proof of the availability of the peer. The lack of data or a response from the remote peer causes the local peer to clear the local IPsec connection and recover the security association resources assigned. After clearing the resources, it may attempt to reestablish a connection with the same peer or with an alternate peer.

Note: Feature configuration requires Management Server hotfix. Contact Checkpoint Support.

Configuration

To configure DPD on the database:

1. Open the GuiDBEdit tool.
2. In "Network Objects" select the required peer object.
3. Go to the "VPN" attribute.
4. Set the `tunnel_keepalive_method` attribute to `dpd` or `passive`

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>passive</td>
<td>DPD requests are not sent to peers. Tunnel with such peers are monitored according to incoming IPsec traffic as well as incoming DPD requests.</td>
</tr>
<tr>
<td>dpd</td>
<td>DPD requests are sent at regular intervals (every 10 seconds, by default), unless an IPsec traffic has been recently received from the peer.</td>
</tr>
</tbody>
</table>
5. Install the policy

Gateway side configuration is automatically configured when running the script
\texttt{asg\_lte\_config enable} as part of the LTE configuration

**VPN Sticky SA (for LTE)**

**Description**

To support LTE environments, you must enable the VPN sticky SA feature. Using this feature ensures that an LTE device holds only one outgoing SA against the 61000 Gateway which is usually a requirement for an LTE device.

**How does it work?**

First packet of clear connection is forwarded to firewall which calculates which blade will hold the SA against the relevant remote peer (LTE device). The packet is then forwarded to this blade. In addition a "correction" entry is added to hold the cached decision so the next packets will be forwarded directly from Performance Pack to the blade that holds the SA against the LTE device.

**Limitations:**

- Connection are synched to all blades (instead synchronizing only to the backup blade).
- Correction layer with NAT is not supported for VPN traffic.

**Configuration**

Gateway side configuration is automatically configured when running the script \texttt{asg\_lte\_config enable} as part of the LTE configuration.

If you are configure this feature without configuring LTE follow these steps:

1. Access 61000 Gateway and run from shell: \texttt{g\_update\_conf\_file $FWDIR/modules/fwkern.conf fwha\_vpn\_sticky\_tunnel\_enabled=1}
2. Reboot all the blades. [\texttt{reboot –b all}]

**Verification:**

Assuming SecureXL is enabled, check if 'VPN Sticky Tunnel Enabled' is 'yes' in \texttt{/proc/ppk/conf file}.

[From Expert Shell run: \texttt{g\_cat /proc/ppk/conf | grep VPN}]
SCTP Acceleration

Smart Dashboard Configuration:

1. Create SCTP as “other” service using IP protocol 132

2. Enable “Accept Replies” property in the advanced tab of the created SCTP service
3. Install the policy on the Gateway

Gateway Configuration
1. Connect to the SMO expert mode. Run `shell`.
2. Open: `$FWDIRR/boot/modules/fw kern.conf` for editing. If the file does not exist, create it.
   a. Add `sxl_accel_proto_list=132` to the file.
3. Open: `/$PPKDIRR/boot/modules/simkern.conf` for editing. If the file does not exist, create it.
   a. Add `sim_accel_non_tcpudp_proto=1` to the file.
4. Copy the file to all SGMs by running:
   a. `g_cp2blades $FWDIRR/boot/modules/fw kern.conf`
   b. `g_cp2blades $PPKDIRR/boot/modules/simkern.conf`
5. Reboot all blades. Run `reboot -b all`
VPN Performance Enhancements

The following performance enhancements have been applied:

1. **SPI Based Traffic Distribution in SSM160**: Utilizing all SGMs to handle VPN traffic by distributing the packets according to the SPI field.

2. **SPI Affinity**: Improve network performance of VPN traffic by distributing packet processing across cores using Intel’s NIC Multi-Queue technology.

3. **VPN Templates**: Accelerate session rate by adding VPN Templates to the SecureXL technology.

**SPI Based Traffic Distribution in SSM160**

**Description:**

The default distribution method of SSM is to distribute traffic to blades according to the IP address field in the packet header. In Site-To-Site topology when there is a low number of remote peers, using this method to distribute VPN traffic might be insufficient. The distributors will only see the tunnel IP address and not the inner connection’s IP address. This causes most of the traffic to be handled on a few blades instead of using the entire system.

In SSM160, an administrator can enable the SPI distribution method for VPN traffic.

**Configuration**

To enable distribution by SPI on encrypted traffic, from gclish run:

**Syntax**

```
asg dxl spi <disable | enable | verify [-v]>
```

<table>
<thead>
<tr>
<th>option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>disable</td>
<td>Disable SPI state</td>
</tr>
<tr>
<td>Enable</td>
<td>Enable SPI state</td>
</tr>
<tr>
<td>verify</td>
<td>Verify current SPI state</td>
</tr>
</tbody>
</table>

SPI distribution mode is disabled by default.

**Example**

```
ch01-01 > asg dxl spi enable
```

Output:

```
Initiating configuration sequence:
Enabling SPI on chassis1 SSMs.
Enabling SPI on chassis2 SSMs.
Updating blades distribution matrix, Matrix size 1024.
```
From Smart Dashboard, you can choose one of these settings to control the number of VPN tunnels between peer gateways:

- One VPN tunnel per each pair of hosts. A VPN tunnel is created for every session initiated between every pair of hosts.

- One VPN tunnel per subnet pair. Once a VPN tunnel has been opened between two subnets, subsequent sessions between the same subnets share the same VPN tunnel. This is the default setting and is compliant with the IPSec industry standard.

- One VPN tunnel per Gateway pair. One VPN tunnel is created between peer gateways and is shared by all hosts behind each peer gateway.

Configuration can be done from:

- Gateway Properties > IPSec VPN > VPN Advanced
- Star or Meshed Community Properties > Tunnel Management
SmartCenter Configuration

From SmartCenter Server you can divide 'VPN per subnet' tunnel to smaller subnets, by editing the max_subnet_for_range parameter in the relevant "user.def" file. (There is one "user.def" per supported CP version on the management machine)
For example the following configuration will divide the subnet 11.11.11.0/24 to 16 small subnets (11.11.11.1-11.11.11.16, 11.11.11.17-11.11.11.32 and so on) and SPI will be shared according to the smaller subnets instead of the big one.

```
#ifndef _user_def_
#define _user_def_

// User defined INSPECT code

all@my_vpn = { <11.11.11.0, 11.11.11.255 ; 255.255.255.240> }
#endif /* _user_def */
```

Note: These subnets are only being applied to the specified gateway, in this case, "my_vpn" gateway.

**SPI Affinity**

**Description**

Multi Queue technology improves network performance by distributing packet processing across cores. Since the same IP addresses are used for most traffic on VPN packets, to improve the distribution to multiple cores, the distribution is done according to the SPI field in the packet header instead of the IP address field.

Note: This mode should be configured when “SPI based Traffic Distribution in SSM160” is configured.

**Configuration**

In order to enable, disable or view the SPI affinity configuration, run "asg_spi_affinity" (from Expert shell):
Syntax: `asg_spi_affinity [enable <1|2|all> [vlan <1|2|all>]] | disable | verify`

<table>
<thead>
<tr>
<th>option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disable</td>
<td>Disable SPI affinity</td>
</tr>
</tbody>
</table>
| enable <1|2|all> [vlan <1|2|all>] | Enable SPI affinity for given SSMs (1/2/all).
|                         | **vlan**: use this option with the relevant SSM ID (1/2/all). If a vlan is configured on the given SSM interfaces. |
|                         | Note: If not all interfaces on the SSM are configured as VLANs, set this option if a major part of the traffic passes through the VLAN interfaces. |
| Verify                  | Shows the current status of SPI affinity                         |

**Note**: SPI affinity might affect the distribution of clear packets, therefore we recommend that you use SPI affinity if most of the inbound traffic is VPN traffic.

**VPN Templates**

VPN templates are part of the SecureXL templates, which are used to accelerate session rate, particularly for short connections (HTTP, DNS). These templates allow the creation of new connections in the acceleration layer and only notify the Firewall layer if the connection is long or F2Fed.

Until now, connections that required encryption/decryption suffered from a double performance penalty:

1. Connections were not able to use SecureXL templates, leading to a low session rate result for VPN traffic.
2. Since VPN is not integrated with CoreXL, all the connections were handled on a single core (instance 0).

After enabling offloading the VPN templates, both penalties are irrelevant for session rate tests since connections now use SecureXL templates. The connections are handled in SecureXL instead of the firewall, and all the cores that are assigned to SecureXL process them.

**Configuration**

VPN templates are enabled by default. In order to disable VPN templates the change the "cphwd_offload_vpn_templates" value to 0:

5. Run from gclish: `update_conf_file fwkern.conf cphwd_offload_vpn_templates=0`
6. Reboot all blades.

In order to re-enable VPN templates, change the value of "cphwd_offload_vpn_templates" back to 1. You can also use the "lwctl set int" option to change the values, but the change does not survive reboot and does not sync to other blades.
Chapter 3

Configuring the 6in4 Internet Transition Mechanism

Description
Use this command to move IPv6 traffic over a network that does not support IPv6. The command uses the 6in4 Internet transition protocol to encapsulate IPv6 traffic for IPv4 links.

To create 6in4 virtual interfaces, run these commands in this order:

- add interface <physical-if> 6in4 <6in4-id> remote <remote-ipv4-address> [ttl "ttl"]
- set interface <sit if name> ipv6-address <address> mask-length 64

Adding the Interface
Use this command to add the interface.

Syntax
add interface <physical-if> 6in4 <6in4-id> remote <remote-ipv4-address> [ttl "ttl"]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>physical-if</td>
<td>The physical interface encapsulated traffic will leave the system from, for example eth1-01.</td>
</tr>
<tr>
<td>6in4-id</td>
<td>A numerical identifier for the 6in4 Virtual Interface.</td>
</tr>
<tr>
<td>remote-ipv4-address</td>
<td>IPv4 address of the remote peer.</td>
</tr>
<tr>
<td>ttl</td>
<td>Time-to-live: the number of router hops before packets are discarded.</td>
</tr>
</tbody>
</table>

Example
> add interface eth1-01 6in4 999 remote 50.50.50.10
1_01:
Success

Comments
- Despite having specified a single physical interface (eth1-01) on the command line, the virtual (sit_6in4) interface is created for eth1-01 on all SGMs.
- To see the virtual interfaces for each SGM, run: show interface eth1-01 6in4s.

Setting the Interface
Use this command to set the interface.

Syntax
set interface <sit if name> ipv6-address <address> mask-length 64
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sit if name</td>
<td>The name of the virtual interface, which begins: sit_6in4_&lt;ID_number given in previous command&gt;.</td>
</tr>
<tr>
<td>address</td>
<td>IPv6 address.</td>
</tr>
</tbody>
</table>

**Example**

> set interface sit_6in4_999 ipv6-address 30:30:30::1 mask-length 64

Success

### Deleting the 6in4 Virtual Interface

**Run:** delete interface <physical-if> 6in4 <6in4-id>. For example:

> delete interface eth1-01 6in4 999

Success

### Asg Search and 6in4

- When using the asg search command to discover which SGM handles a specific connection (actively or as backup) and which chassis, IPv4 addresses of a remote peer may show as being handled by more than 1 SGM.
- asg search run on IPv6 addresses show:
  - 1 SGM on the active chassis
  - 1 SGM on the standby chassis
Generic Routing Encapsulation – GRE (asg_gre)

**Description:**
Generic Routing Encapsulation (GRE) is a tunneling protocol that can encapsulate a wide variety of network layer protocols inside virtual point-to-point links over an Internet Protocol internetwork.

**Syntax:**
# asg_gre load | stat | verify

**Example:**

**Configuration:**
To configure GRE, you will need to edit this configuration file:
$FWDIR/conf/gre_loader.conf

**Tunnel configuration:**
tunnel=<tunnel interface name> local_tun_addr=<local tunnel ip address> remote_tun_addr=<remote tunnel ip address> phy_ifname=<physical interface name> local_addr=<local physical address> remote_addr=<remote physical address> ttl=<ttl>

**Route configuration:**
tunnel_route=<tunnel interface name> remote_tun_addr=<remote tunnel ip address> network=<network>

**Configuration Example:**
To configure tunnel interface with these parameters:
Tunnel interface name: "GREtun"
Local tunnel address 10.0.0.3
Remote tunnel address 10.0.0.4
Physical interface eth2-01
Local address 40.40.40.1
Remote address 40.40.40.2
ttl 64

Use the following line:
tunnel=GREtun local_tun_addr=10.0.0.3 remote_tun_addr=10.0.0.4 phy_ifname=eth2-01 local_addr=40.40.40.1 remote_addr=40.40.40.2 ttl=64

To add route for 50.50.50.0/24 to go through the tunnel use the following line:
tunnel_route=GREtun remote_tun_addr=10.0.0.4 network=50.50.50.0/24

**Note:** All parameters are required

After editing the configuration file, use asg_gre to load it:

**Output:**

```bash
# asg_gre load
# asg_gre load
Copying configuration file to all blades... done
1_01:
Clearing existing GRE tunnels...
Loading GRE module... Done
Loading tunnel interface: GREtun
Loading route: 50.50.50.11/32 via 10.0.0.4 (GREtun)
Loading tunnel interface: GREtuA
Loading tunnel interface: GREtuB
Loading tunnel interface: GREtuC
```
Role Based Administration (RBA)

Description:

The access to gclish features is controlled by Role Based Administration (RBA): each user is assigned with a role. Each role has a set of read-only features and read-write features. The user is not exposed to any features, other than the ones assigned to his role.

RBA configuration and properties in 61000 is identical to Gaia. Please refer to Gaia Admin Guide for more details.

Few notes:

Extended commands have no read/write notion. When an extended command is added to a role (either as read or write), it can be executed by the users assigned to this role, regardless of its implications.

Each extended command should be separately added to role. Since asg command is the “entrance” to the 61000 Security System, it usually needs to be added to all roles.

In order to allow user to run extended commands, its uid must be zero. This property is enforced when adding new users.

The user account information file located at /etc/passwd should not be edited by the user. RBA configuration should be performed only via gclish.

Example:
g61000-ch01-01 > add rba role myRole domain-type System readonly-features chassis,interface readwrite-features route
g61000-ch01-01 > add user myUser uid 0 homedir /home/myUser
g61000-ch01-01 > set user myUser password
g61000-ch01-01 > add rba user myUser roles myRole
g61000-ch01-01 > show rba role myRole
RADIUS Authentication

Description
RADIUS (Remote Authentication Dial-In User Service) is a client/server authentication system that supports remote-access applications. User profiles are kept in a central database on a RADIUS authentication server. Client computers or applications connect to the RADIUS server to authenticate users. You can configure the 61000 Security System to work as a RADIUS client. The 61000 Security System does not include RADIUS server functionality. You can configure the 61000 Security System to authenticate users even when they are not defined locally. See Configuring Non-local RADIUS Users. You can configure your 61000 Security System computer to connect to than one RADIUS server. If the first server in the list is unavailable, the next RADIUS server in the priority list connects. You can delete a server at all times. Setting 61000 as a Radius client

Use the `aaa radius-servers` commands to add, configure, and delete Radius authentication servers

Syntax:
To configure RADIUS for use in a single authentication profile:

```
add aaa radius-servers priority VALUE host VALUE [ port VALUE ] prompt-secret timeout VALUE
add aaa radius-servers priority VALUE host VALUE [ port VALUE ] secret VALUE timeout VALUE
```

Example: Adding a new radius server 1.1.1.1 which listens on port 1812

```
add aaa radius-servers priority 1 host 1.1.1.1 port 1812 prompt-secret timeout 3
```

To delete a RADIUS configuration:

```
delete aaa radius-servers priority VALUE
```

To change the configuration of a RADIUS entry:

```
set aaa radius-servers priority VALUE host VALUE
set aaa radius-servers priority VALUE new-priority VALUE
set aaa radius-servers priority VALUE port VALUE
set aaa radius-servers priority VALUE prompt-secret
set aaa radius-servers priority VALUE secret VALUE
set aaa radius-servers priority VALUE timeout VALUE
```

Note: the configuration is done according to the priority and not the sever ID or name.

To view a list of all servers associated with an authentication profile:

```
show aaa radius-servers list
```

To view the RADIUS server configuration:

```
show aaa radius-servers priority VALUE host
show aaa radius-servers priority VALUE port
show aaa radius-servers priority VALUE timeout
```

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Note: After the 61000 is configured as a RADIUS client, any authentication request will be forwarded to the RADIUS server. As a result, every account which is configured locally should be configured on the RADIUS server as well.

**Configuring Non-local RADIUS Users**

In order to allow login with non-local user to the 61000 Security System, you need to define a default role for all non-local users that are configured in the Radius server.

The default role can include a combination of administrative (read/write) access to some features, monitoring (read-only) access to other features, and no access to other features.

**Syntax:** to define default role for non-local users

```
add rba role radius-group-any domain-type System readonly-features <List>
readwrite-features <List>
```

- **readonly-features <List>** - Comma separated list of Gaia features that have read only permissions in the specified role.
- **readwrite-features <List>** - Comma separated list of Gaia features that have read/write permissions in the specified role.
Example:
add rba role radius-group-any domain-type System readonly-features arp

Verification:

Authenticate to the 61000 Security System with a non-local user:
   MyLaptop > ssh my_radius_user@my_61k_server

Upon successful authentication, the user ‘my_radius_user’ will be assigned the role ‘radius-group-any’ granted all the privileges defined in the radius-group-any role

Configuring Local Radius users (with specific role)

You can configure users to have different role then the default role by creating new users on the 61000 Security System systems and assigning them the required role.

Creating a new user

Syntax: To create a new local user
   add user <Name> uid 0 homedir <Path>

Example: add a new user named “local”
   add user local uid 0 homedir /home/local

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>Login name of the user.</td>
</tr>
<tr>
<td>homedir</td>
<td>Full path for the user home directory</td>
</tr>
</tbody>
</table>

Setting user password

It is recommended to leave the local user's password blank.

Setting user role

It is possible to choose a role from any preexisting roles, or to create a new role and to provide it with custom permissions. The “Adding a new role” section which is present inside this document outlines the procedure required for creating a new role.

Syntax: To assign a user to a role
   add rba user <User> roles <Role>

Example: to assign user “local” to role “radius”
   add rba user local roles radius

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>The user name to assign a role to.</td>
</tr>
</tbody>
</table>
**Adding a new role**

**Syntax:** To add a new role

```
add rba role <Name> domain-type System
readonly-features <List>
readwrite-features <List>
```

**Example:** Adding a new radius role

```
add rba role radius domain-type System
readonly-features chassis,configuration
readwrite-features aaa-servers
```

**Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>Determines the role's name.</td>
</tr>
<tr>
<td>readonly-features</td>
<td>Comma separated list of features to grant read only permissions for.</td>
</tr>
<tr>
<td>readwrite-features</td>
<td>Comma separated list of features to grant read/write permissions for.</td>
</tr>
</tbody>
</table>
Port Mirroring (Monitor Port) Configuration

Monitor Mode lets a Security Gateway listen to traffic from a Mirror port or Span port on a switch. Mirror port on a Check Point gateway is usually configured to monitor and analyze network traffic with no effect on production environment. The mirror port duplicates the network traffic and records the activity in logs.

You can use mirror ports:

- As a permanent part of your deployment, to monitor the use of applications in your organization.
- As an evaluation tool to see the capabilities of the Application Control and IPS blades before you decide to purchase them.

The mirror port does not enforce a policy and therefore you can only use it to see the monitoring and detecting capabilities of the blades.

Benefits of a mirror port include:

- There is no risk to your production environment.
- It requires minimal set-up configuration.
- It does not require TAP equipment, which is much more expensive.

Configuration:

1. Create a bridge interface via Clish shell and add the interface to the object topology via Smart Dashboard
   
   See configuration procedure in L2 Bridge Mode configuration section.

2. Attach the interface to the bridge you created via clish shell.
   
   Syntax:   clish> add bridging group <bridge_id> interface <IF_NAME>

   Example:  clish> add bridging group 1 interface eth2_1

   Verify:  run ’asg if -a’ and see that a bridge has been created with the name br<group_id> and the interface is defined as ‘Bridge Port’

3. Create the file /etc/netconf.C on SGM 1_1 expert shell and edit it with the interface name, its MAC address and bridge interface name.
example:

```
conf
  : {conns
     : {conn
        : ifname (eth2_1)
        : type (1)
        : mtu (1500)
        : onboot (1)
        : iff-up (1)
        : hwaddr ("00:1c:7f:a1:10:fe")
        : s-code (0)
        : master-bridge (br1)
        : mirror-port (1)
     )
  }

Note: For more than 1 monitor port add new 'conn' set to the 'conns' array with the relevant interface, mac address and bridge interface.

4. Copy the netconf.C file to all SGMs  
   [Expert@Chassis-ch01-01:0]# g_cp2blades /etc/netconf.C

5. Connect to SDB and turn off anti spoofing on the bridge interface.

6. Install Security policy

Notes:

1. APP Control policy- change destination default settings be from 'internet' to 'any'
2. Turn off Sequence Verifier in IPS (Reducing CPU Utilization)
3. Disable Out of State Protections (Reducing CPU Utilization)
4. Distribution mode should be set to general: run from gclish 'asg dxl dist_mode set' and choose General (option 2).
Chapter 4

Networking

Configuring Link Aggregation (Bonding)

Link aggregation combines multiple physical interfaces into a virtual interface called a bond. Bonded interfaces (known as slaves) add redundancy to a connection as well as increasing the connections throughput to a level beyond what is possible using a single physical interface.

To create an interface bond you need to run these commands in this order from the `gclish` shell:

<table>
<thead>
<tr>
<th>Commands in Running Order:</th>
<th>Purpose:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add bonding group &lt;BOND_id&gt;</td>
<td>Creates a bonding group</td>
</tr>
<tr>
<td>set bonding group &lt;BOND_id&gt; mode &lt;BOND_MODE&gt;</td>
<td>Sets a bonding mode: 802.3ad (LAPC) or XOR</td>
</tr>
<tr>
<td>set interface &lt;IF_NAME&gt; state on</td>
<td>Sets the slave interface to on</td>
</tr>
<tr>
<td>add bonding group &lt;BOND_ID&gt; interface &lt;IF_NAME&gt;</td>
<td>Enslaves interfaces to the bond</td>
</tr>
</tbody>
</table>

**Note** - Before running the link aggregation commands, make sure that the slave interfaces do not have an IP Address already assigned.

**Creating a Bonding Group.**

- **Description**: Use this command to create a bonding group. A bonding group is a single virtual interface or bond. A bond can contain multiple Slaves.

  **Note**: the `<BOND_id>` must be a number. The bond name is created automatically with the bond id. For example, entering 4 for the bond id creates a virtual interface named bond4.

- **Syntax**: Add bonding group `<BOND_id>`

- **Example**: `> add bonding group 4`
Output
1_01:
success
1_02:
success
1_03:
success
2_01:
success
2_03:
success
>

Explanation
Running the command creates one virtual interface, bond4, consisting of all the SGM interfaces on each chassis.

**Setting a Bonding Mode**

Description
Use this command to set a bonding mode.

The following Bond modes are supported in the 61000 Security System:

- **8023AD (LACP):** Do dynamic bonding according to the IEEE 802.3ad protocol
- **Active/Backup:** Bond build up from one interface Active while other interface is in standby. When the active interface encounters a problem failover occurs to other Bond interface.
- **XOR:** Do load sharing based on layer2, or 3 and 4.

Note: round-robin mode is not supported on the 61000 Security System.

Syntax  
set bonding group <BOND_id> mode <BOND_MODE>

Example
set bonding group 4 mode 8023A

Output
1_01:
success
1_02:
success
1_03:
success
2_01:
success
2_03:
success
>

Explanation
Physical interfaces enslaved to bond4 do load sharing according to the 802.3ad protocol
Setting a Polling interval
Use this command to set the polling interval.

Syntax set bonding group <BOND_ID> mii-interval 100

Explanation The polling interval is how often (in milliseconds) the OS checks to see if the bond is up.

Setting the Slave Interface to On
Description Use this command to switch the interface on or off.

Note: Run this command from the Bash shell.

Syntax set interface <Interface_name> state on

Example set interface eth1-02 state on

Enslaving Interfaces
Use this command to enslave a physical interface to a named bond.

Syntax add bonding group <BOND_ID> interface <Interface_name>

Example add bonding group 4 interface eth1-02

Explanation Adds interface eth1-02 to bond4

Removing Slaves from a Bond
To remove a slave interface from a bond run:

Syntax delete bonding group <bond_id> interface <interface_name>

Example delete bonding group 1 interface eth1-02

Note - There is no command to delete all slave interfaces at the same time.

Deleting a Bonding Group
To delete a bonding group you must first delete all slaves one by one. Then run:

Syntax delete bonding group <bond_id>

Example delete bonding group 4

Explanation This command deletes bond4

Configuring VLANs
Description Use this command to configure VLANs.

Syntax add interface <interface> vlan <vlan-id>
set interface <interface>.<vlan-id> ip-address <ip-address> mask-length <mask-len>
delete interface <interface> vlan <vlan-id>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>The name of the interface</td>
</tr>
<tr>
<td>vlan</td>
<td>Vlan ID number</td>
</tr>
<tr>
<td>mask-length</td>
<td>Network mask length</td>
</tr>
</tbody>
</table>

**Example 1**

```
add interface eth2-03 vlan 444
```

**Output**

```
> add interface eth2-03 vlan 444
  1_01:
success
```

**Example 2**

```
set interface eth2-03.444 ipv4-address 30.30.30.1 mask-length 24
```

**Output**

```
> set interface eth2-03.444 ipv4-address 30.30.30.1 mask-length 24
  1_01:
success
```

**Example 3**

```
show interface eth2-03 vlans
```

**Output**

```
> show interface eth2-03 vlans
  1_01:
  eth2-03.444
```

**Comments**
The output shows VLAN interfaces on physical interface eth2-03.

**Example 4**

```
delete interface eth2-03 vlan 444
```

**Output**

```
> delete interface eth2-03 vlan 444
  1_01:
success
```

### Configuring Dynamic Routing - Unicast

To ease the administrative and operational overhead of using only static routes, the 61000 Security Systems supports dynamic routing protocols OSPF and BGP to:

- Collect routing data regarding remote networks
- Automatically add this data to the system's routing table
To set OSPF on an interface:

Description: Use this command to enable the OSPF protocol on an interface. The 61000 Security Systems implements the ROUTED daemon to listen and send OSPF messages on this interface only.

Syntax:

```
set ospf interface <interface> <area> [on|off]
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>interface</td>
<td>The interface the ROUTED daemon will use to listen for and send OSPF messages.</td>
</tr>
</tbody>
</table>
| Area      | - Specifies the area ID. The area ID must be one of these:  
           |   - An IPv4 address  
           |   - A value between 1 and 4294967295  
           |   - backbone  
           |   - By default, the backbone area is enabled. |
| on/off    | Whether the daemon is listening to messages from the set area? |

Example: 

```
> set ospf interface eth1-01 area backbone on
```

Comments:

- Before running this command, you must run the `set router-id <IP address>` command. If you want to set ospf on interface eth1-01, and the IP address of eth1-01 is 40.40.40.1, then you must run: `set route-id 40.40.40.1 first.`
- To verify that the interface has OSPF enabled, run: `show ospf interfaces`
- To show OSPF state in relation to its neighbors, run: `show ospf neighbors`
- To show OSPF statistics, run: `show ospf summary`

To set BGP:

To configure BGP you need to:

- Set the ID of the Autonomous System
- Set at least one BGP neighbor

To set the AS:

Description: Use this command to set the AS number

Syntax:

```
set as <ID number>
```

Example: 

```
set as 2
```

To set a BGP neighbor:

Description: Use this command to set a BGP neighbor

Syntax:

```
set bgp <internal|external> remote-as <AS number> peer <peer IP address> [on|off]
```

Example: 

```
set bgp internal remote-as 200 peer 192.168.1.1 on
```
### Networking

#### System Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>internal</td>
<td>external</td>
</tr>
<tr>
<td>AS number</td>
<td></td>
</tr>
<tr>
<td>peer address</td>
<td>IP address of remote peer</td>
</tr>
</tbody>
</table>

**Examples**

- set bgp external remote-as 24 on
  
  Adds AS 24 to the system’s configuration

- set bgp external remote-as 24 peer 40.40.40.24 on
  
  Sets the local system interface 40.40.40.24 as a BGP peer for AS 24.

**Comments**

To verify BGP is running:

- To show BGP peers, run: `show bgp peers`
- To show BGP state, run: `show bgp summary`

To deactivate BGP:

- set bgp external remote-as 24 off
- set bgp external remote-as 24 peer 40.40.40.24 off

---

**Changing the Default VMAC (asg_unique_mac_utility)**

**Description**

By default, all 61000 Security Systems have the same VMAC address. This prevents locating more than one setup (Dual Chassis or Single Chassis) on the same network segment. The `asg_unique_mac_utility` command changes the:

- Interface’s default VMAC to a unique value
- Hostname

**Note** - Changing the unique VMAC address results in loss of traffic and connections

**Syntax**

```
asg_unique_mac_utility
```

**Output**

```
+-----------------------------+                          +-----------------------------+                          +-----------------------------+
|                             | Unique MAC Utility       |                             | Unique MAC                   |                             |
|                             | ------------------------- |                             | [25d]                         |                             |
| HOSTNAME                    | [copmodule]              |                             |                               |                             |
| Unique MAC                  | [25d]                    |                             |                               |                             |
```

Choose one of the following options:

1) Set Hostname with Unique MAC wizard
2) Apply Unique MAC from current HOSTNAME
3) Manual set Unique MAC
4) Back to Unique MAC factory default (254)
5) Exit

**Explanation**

Use this command if you intend to deploy a number of 61000 Security Systems on the same network segment.

The menu has four options:
1) Set Hostname with Unique MAC wizard

Using this option you enter:

- A setup name
- A unique MAC setup number between 1-254.

The option adds the _asg suffix and setup number to the setup name. For example:

<table>
<thead>
<tr>
<th>Setup Name</th>
<th>Suffix</th>
<th>Setup number</th>
</tr>
</thead>
<tbody>
<tr>
<td>armgdn</td>
<td>_asg</td>
<td>22</td>
</tr>
</tbody>
</table>

This results in a new Hostname with a unique MAC value of 22 (16 in HEX):

<table>
<thead>
<tr>
<th>New HOSTNAME</th>
<th>Unique MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>armgdn_asg22</td>
<td>22</td>
</tr>
</tbody>
</table>

The setup number replaces the default Magic MAC value of 254. After running this option, all interfaces of type ethX-YZ have the a unique MAC value of 22 (16 in HEX)

2) Apply Unique MAC from current HOSTNAME

Use this option to change the system’s VMAC. The option automatically sets a new VMAC on the relevant interfaces. The new VMAC is derived from the setup number within the hostname. For this reason, the existing hostname must first comply with the setup name/ asg suffix/ setup number convention.

3) Manual Set Unique MAC

Use this option to change the unique MAC according to your own input without changing the HOSTNAME value. The existing HOSTNAME does not have to comply with the setup name/ asg suffix/ setup number convention.

- **Note** - Manually setting the unique MAC without changing the HOSTNAME can lead to confusion when number of 61000 Security Systems exist on the same network segment.

4) Revert to Unique MAC Factory Default

Use this option to set the unique MAC value to its default value (254)

### Verifying the New MAC Address

Use these commands to make sure that the unique MAC value has changed:

- For the unique MAC DB value, run (from the bash shell): `g_allc dbget chassis:private:magic_mac`

```
# # g_allc dbget chassis:private:magic_mac
-** 4 sgms: 1_01 1_02 2_02 2_03 --*
22
```

- For the unique MAC Kernel value, run (from `gclish`): `fw ctl get int fwha_mac_magic`

```
> fw ctl get int fwha_mac_magic
-** 4 sgms: 1_01 1_02 2_02 2_03 --*
fwha_mac_magic = 22
```

You can also display the magic attribute within type ethX-YZ interfaces by using the `ifconfig` command:
# ifconfig eth1-01

eth1-01    Link encap:Ethernet  HWaddr 00:1C:7F:81:01:16
inet6 addr: fe80::21c:7fff:fe81:116/64 Scope:Link
UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500 Metric:1
RX packets:154820 errors:0 dropped:0 overruns:0 frame:0
TX packets:23134 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0 RX bytes:15965660 (15.2 MiB)
TX bytes:2003398 (1.9 MiB)
Configuring Source Based Routing

Source-based routing lets you forward traffic to a destination other than that specified by the destination address in the packet. Source based routing works by maintaining multiple routing tables. Each routing table has a unique set of rules. Based on the source IP address or a system interface, incoming traffic is associated with a specified routing table. Traffic is then routed according to the rules of the table.

To configure source based routing you must:
- Define multiple routing tables
- Associate traffic (based on source IP or incoming interface) with a specified routing table

To create multiple routing tables:
You create a routing table by defining a route. For example:

- `ip ro add default via 151.1.2.2 table 3`
  Running this command creates table 3 with a default route via 151.1.2.2
- `ip ro add default via 251.1.2.2 table 4`
  Running this command creates table 4 with a default route via 251.1.2.2

To associate traffic from a specified interface with a specified routing table:
1. On the gateway, open `$FWDIR/bin/iproute.load` for editing.
2. Associate the traffic using this syntax:
   `ip rule add dev <incoming interface> table <table number>`
   For example, to add a rule that routes traffic from eth3 according to table 3, run:
   `ip rule add dev eth3 table 3`

   **Note** -
   - For IPv6, replace `ip` with `ip -6`. For example: `ip -6 ip -6 rule add dev eth3 table 3`
   - To see rules already listed in table 3, run: `iproute showtable 3`

1. Save and close the file.
2. Copy the file to all SGMs by running: `g_cp2blades $FWDIR/bin/iproute.load`.

To associate traffic from a specified source with a specified routing table:
1. On the gateway, open `$FWDIR/bin/iproute.load` for editing.
2. Associate the traffic using this syntax:
   `ip rule add from <ip address> table <table number>`
   For example, to add a rule that routes traffic from 1.1.1.1 according to table 4, run:
   `ip rule add from 1.1.1.1 table 4`

   **Note** -
   - For IPv6, replace `ip` with `ip -6`
   - To see rules already listed in table 4, run: `iproute showtable 4`

1. Save and close the file.
2. Copy the file to all SGMs by running: `g_cp2blades $FWDIR/bin/iproute.load`.

To create a default route for a specified routing table:
1. On the gateway, open `$FWDIR/bin/iproute.load` for editing.
2. Create a default route using this syntax:
   `ip ro add default via <IP address> table <table number>`
   For example, to add a default route to table 3, run:
   `ip ro add default via 151.1.2.2 table 3`
   (If necessary, replace `ip` with `ip -6`.)
3. Save and close the file.
4. Copy the file to all SGMs by running: `g_cp2blades $FWDIR/bin/iproute.load`

**To delete an interface from a routing table:**

1. On the gateway, open `$FWDIR/bin/iproute.load` for editing.

2. Delete interfaces using this syntax:
   ```
   ip rule del dev <incoming interface> table <table number>
   ```
   For example, to delete eth3 from table 3, run:
   ```
   ip rule del dev eth3 table 3
   ```
   (If necessary, replace `ip` with `ip -6`.)

3. Save and close the file.

4. Copy the file to all SGMs by running: `g_cp2blades $FWDIR/bin/iproute.load`

**To delete a default route:**

1. On the gateway, open `$FWDIR/bin/iproute.load` for editing.

2. Delete default routes using this syntax:
   ```
   ip ro del default via <IP address> table <table number>
   ```
   For example, to delete default route 151.1.2.2 from table 3, run:
   ```
   ip ro del default via 151.1.2.2 table 3
   ```
   (If necessary, replace `ip` with `ip -6`.)

3. Save and close the file.

4. Copy the file to all SGMs by running: `g_cp2blades $FWDIR/bin/iproute.load`

**Verification**

To make sure source based routing is taking place, examine the local routing table.

**To see the local routing table:**

1. Enter `shell` to exit gclish.

2. Enter `iproute.list`.
   - Shows IPv4 routes in all the routing tables.

3. Enter `iproute.list -6`.
   - Shows IPv6 routes in all the routing tables.

**To compare routing tables on all SGMs:**

1. On the command line, enter `shell` to exit the gclish.

2. Enter `g_allc iproute.list`.
   - Shows IPv4 addresses.

3. On the command line, enter `g_allc iproute.list -6`
   - Shows IPv6 routes.

*Note* - After editing `iproute.load`, you must copy the edited file to all SGMs to implement the changes.

**ECMP Configuration**

**Description**

Equal-cost multi-path routing (ECMP) is a routing strategy where you manually define a static route to a number of next-hop gateways. It potentially offers substantial increases in bandwidth by load-balancing traffic over multiple paths to reach the destination network defined in the static route.
Syntax

set static-route <network> nexthop gateway address <gw ip address> on

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network&gt;</td>
<td>The IP address of the destination network</td>
</tr>
<tr>
<td>&lt;gw ip address&gt;</td>
<td>The IP address of the next-hop gateway</td>
</tr>
</tbody>
</table>

Example

set static-route 50.50.50.0/24 nexthop gateway address 20.20.20.101 on
set static-route 50.50.50.0/24 nexthop gateway address 20.20.20.102 on
set static-route 50.50.50.0/24 nexthop gateway address 20.20.20.103 on

Comments

To reach addresses on the 50.50.50.0/24 network, packets must first be forwarded to one of these gateways:

- 20.20.20.101
- 20.20.20.102
- 20.20.20.103

Setting the static route enforces the first hop to one of these gateways.

Verification

To make sure static routes to the next-hop gateways are being enforced, run: show route static

> show route static
1_01:
Codes: C - Connected, S - Static, R - RIP, B - BGP, O - OSPF IntraArea (IA - InterArea, E - External, N - NSSA) A - Aggregate, K - Kernel Remnant, H - Hidden, P - Suppressed

S 0.0.0.0/0 via 192.168.33.1, eth2-01, cost 0, age 2092
S 5.5.5.0/24 via 20.20.20.101, eth1-01, cost 0, age 322
via 20.20.20.102, eth1-01
via 20.20.20.103, eth1-01

The output shows that the static route to 50.50.50.0/24 is via three next-hop gateways.

Disabling ECMP

ECMP is enabled by default. To disable it:

4. Open this file for editing: $PPKDIR/boot/modules/simkern.conf . If simkern.conf does not exist, create it.
5. Add this line: sim_routing_by_source=0
6. Save the file and reboot.
**Enhanced Failover of ECMP Static Routes**

**Description**

The enhanced routing features automatically start failover on detection of unreachable next hop gateways for ECMP static routes. It ensures that the required destination will be routed only from reachable next-hops by deleting unreachable next-hops from the routing table, and add it again when they are reachable.

The new functionality probes each next hop gateway of a static route to detect its reachability status. Probing is done on each SGM, with "ping", the standard ICMP echo protocol. If the next hop is unreachable it is being removed from the routing table and re-entered when it is detected as reachable.

**Syntax**

In order to activate enhanced failover on a static route run from gclish:

```bash
> set static-route <network>/<subnet length> ping on
```

Note: enhanced ECMP failover can be configured after you configured ECMP static route. (see Configuring)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network&gt;</td>
<td>The IP address of the destination network</td>
</tr>
<tr>
<td>&lt;subnet length&gt;</td>
<td>The subnet length of the destination network</td>
</tr>
</tbody>
</table>

In order to adjust ping behavior, use:

```bash
> set ping count <VALUE>
> set ping interval <VALUE>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>count &lt;VALUE&gt;</td>
<td>Number of packets to be sent before next hop is declared dead</td>
</tr>
<tr>
<td>Interval &lt;VALUE&gt;</td>
<td>Time in seconds to wait between two consecutive pings</td>
</tr>
</tbody>
</table>

**Example**
Step 1: set ECMP for destination 5.5.5.0/24

> set static-route 5.5.5.0/24 nexthop gateway address 10.33.85.2 on
> set static-route 5.5.5.0/24 nexthop gateway address 10.33.85.4 on
> set static-route 5.5.5.0/24 nexthop gateway address 10.33.85.100 on
> show route

1_01:
Codes: C - Connected, S - Static, R - RIP, B - BGP, O - OSPF IntraArea (IA - InterArea, E - External, N - NSSA) A - Aggregate, K - Kernel Remnant, H - Hidden, P - Suppressed

S 0.0.0.0/0 via 192.168.33.1, eth2-01, cost 0, age 2092
S 5.5.5.0/24 via 10.33.85.2, eth1-01, cost 0, age 322
via 10.33.85.4, eth1-01
via 10.33.85.100, eth1-01

Step2: enable failover ECMP on all static route configured for destination 5.5.5.0/24

> set static-route 5.5.5.0/24 ping on

Step3: validation

When next-hop 10.33.85.2 is unreachable: (no ICMP replies), after 3 pings (by default) it will be removed from the routing table:

[Expert@CH_Lena-ch02-01]# tcpdump -nepi eth1-01 host 10.33.85.2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1-01, link-type EN10MB (Ethernet), capture size 96 bytes
14:40:48.388032 00:1c:7f:a1:01:55 > 00:50:56:a7:7f:f5, ethertype IPv4 (0x0800), length 62: 10.33.85.1 > 10.33.85.2: ICMP echo request, id 53007, seq 43981, length 28
14:40:58.388425 00:1c:7f:a1:01:55 > 00:50:56:a7:7f:f5, ethertype IPv4 (0x0800), length 62: 10.33.85.1 > 10.33.85.2: ICMP echo request, id 53007, seq 43981, length 28
14:41:08.387895 00:1c:7f:a1:01:55 > 00:50:56:a7:7f:f5, ethertype IPv4 (0x0800), length 62: 10.33.85.1 > 10.33.85.2: ICMP echo request, id 53007, seq 43981, length 28

The route has been deleted from the routing table

01 > show route
1_01:
Codes: C - Connected, S - Static, R - RIP, B - BGP, O - OSPF IntraArea (IA - InterArea, E - External, N - NSSA) A - Aggregate, K - Kernel Remnant, H - Hidden, P - Suppressed

S 0.0.0.0/0 via 192.168.33.1, eth2-01, cost 0, age 2511
S 5.5.5.0/24 via 10.33.85.4, eth1-01, cost 0, age 52
via 10.33.85.100, eth1-01

When 10.33.85.2 is reachable again we can see in the tcpdump that it replies to ping requests and it is added to the routing table:

[Expert@CH_Lena-ch02-01]# tcpdump -nepi eth1-01 host 10.33.85.2
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1-01, link-type EN10MB (Ethernet), capture size 96 bytes
14:38:08.388224 00:1c:7f:a1:01:55 > 00:50:56:a7:7f:f5, ethertype IPv4 (0x0800), length 62: 10.33.85.1 > 10.33.85.2: ICMP echo request, id 53007, seq 43981, length 28
14:38:08.388462 00:50:fc:58:80:0a > 00:1c:7f:0f:00:fe, ethertype IPv4 (0x0800), length 62: 10.33.85.2 > 10.33.85.1: ICMP echo reply, id 53007, seq 43981, length 28
14:38:18.387762 00:1c:7f:a1:01:55 > 00:50:56:a7:7f:f5, ethertype IPv4 (0x0800), length 62: 10.33.85.1 > 10.33.85.2: ICMP echo request, id 53007, seq 43981, length 28
Validation

7. Run from gclish show route and verify that only ECMP static routes with reachable next-hops are shown.
8. Run tcpdump to verify that each few seconds there is a ping request on the interface with static route and ping on.

Proxy ARP for Manual NAT – (local.arp file)

Description:
Proxy ARP is a mechanism that allows the configuration of a Gateway to respond to ARP requests on behalf of other hosts. For a complete documentation regarding Proxy ARP configuration please refer to sk30197.

Configuration:
In order to configure the proxy ARP mechanism on the 61000 Security System Gateway:

1. Add any IPs for which the 61000 Security System should answer to ARP requests and the respective MAC addresses to be advertised to the $FWDIR/conf/local.arp file on the local SGM.

   Note: Interface VMAC value is different between Chassis when working on a Dual Chassis setup. When editing the local.arp file, MAC values should be taken from the local SGM.

   For example, in order to reply to ARP requests for IP 192.168.10.100 on interface eth2-01 with MAC address 00:1C:7F:82:01:FE, add the following entry to the local.arp file:

   192.168.10.100 00:1C:7F:82:01:FE

2. Execute the command local_arp_update on the SGM with the updated file in order to distribute it among all the SGMs in the system. That command distributes the local.arp file to any SGM in the system, automatically changes the MAC values for SGMs on another chassis.

3. Make sure "Automatic ARP Configuration" is disabled in SmartDashboard:
   Smart Dashboard -> Policy -> Global Properties -> NAT -> and disable "Automatic ARP configuration".

4. Install policy (in order for the updated proxy ARP entries to be applied)
Notes:
1. When adding additional SGMs to a system that has the proxy ARP configured, the local.arp file will be copied and applied during the configuration cloning.
2. Proxy ARP is also required when configuring Connect Control on the 61000 Security System.

Verification:
In order to verify that all the entries in local.arp file are applied correctly on the system run asg_local_arp_verifier. Manual comparison can be done by running `g_fw ctl arp`.

Port speed configuration

*asg_port_speed*

The "asg_port_speed" tool is a useful tool for setting SSMs interfaces speed on both chassis and verifying there values. While setting new interface speed the asg_port_speed run on both chassis and update the relevant SSM and the configuration file (/etc/ssm_port_speed.conf).

The verifier checks that SSMs interfaces speed have the same configuration on both chassis and that the configuration file is the same on all active SGMs.

Any difference between configurations will be prompt as Failed. The verifier also set to run in 'asg diag verify'.

Syntax:

<table>
<thead>
<tr>
<th>use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>asg_port_speed set &lt;interface&gt; &lt;speed&gt;</code></td>
<td>Set interface speed on both chassis. Speed value is in Mbps (could be 10000/1000/auto). Note : Setting Auto-speed is not supported</td>
</tr>
<tr>
<td><code>asg_port_speed verify</code></td>
<td>verify that all SSMs interfaces on both chassis are configured to the same speed configured in the configuration file and that configuration file is the same on all active SGMs.</td>
</tr>
<tr>
<td><code>asg_port_speed --help</code></td>
<td>Prints the usage of the tool.</td>
</tr>
</tbody>
</table>
**Examples:**

**asg_port_speed set**

```bash
# asg_port_speed set eth1-06 1000

Chassis1

Interface eth1-06 speed was set to 1000

Updating /etc/ssm_port_speed.conf
Copy /etc/ssm_port_speed.conf to all SGMs
Operation completed successfully
```

**asg_port_speed verify**

```bash
# asg_port_speed verify

Comparing SSMs configuration with /etc/ssm_port_speed.conf configuration

<table>
<thead>
<tr>
<th>Interface</th>
<th>Conf</th>
<th>Chassis1</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth1-01</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-02</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-03</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-04</td>
<td>1G</td>
<td>1G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-05</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-06</td>
<td>1G</td>
<td>1G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-07</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-09</td>
<td>40G</td>
<td>40G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-10</td>
<td>auto</td>
<td>auto</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-11</td>
<td>auto</td>
<td>auto</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-12</td>
<td>auto</td>
<td>auto</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-13</td>
<td>40G</td>
<td>40G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-14</td>
<td>auto</td>
<td>auto</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-15</td>
<td>auto</td>
<td>auto</td>
<td>Passed</td>
</tr>
<tr>
<td>eth1-16</td>
<td>auto</td>
<td>auto</td>
<td>Passed</td>
</tr>
<tr>
<td>eth2-01</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth2-02</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth2-03</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth2-04</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth2-05</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
<tr>
<td>eth2-06</td>
<td>10G</td>
<td>10G</td>
<td>Passed</td>
</tr>
</tbody>
</table>
```
Comparing /etc/ssm_port_speed.conf on all SGMS
/etc/ssm_port_speed.conf is the same on all active SGMs

SSMs port speed verification SUCCEEDED

Limitations

- asg_port_speed tool doesn’t support setting of 40GbE interface
- SSM160 data port can be configured to 10GbE, 1GbE or auto mode.
- SSM60 data port speed is 10GbE and it is not configurable.

**QSFP Data port speed configuration (40GbE / 4x10GbE)**

Setting port speed to 40GbE

Run the following procedure in order work in 40G mode. On dual chassis configuration, run this procedure on the SSM of both chassis.

1. Connect to the SSM shell (see SSM160 CLI section)
2. Run the following on the SSM:
Setting port to 4x10GbE (this is the default configuration)

Run the following procedure in order work in 4x10G mode. On dual chassis configuration, run this procedure on the SSM of both chassis.

1. Connect to the SSM shell (see SSM160 CLI section)
2. Run the following on the SSM:

T-HUB4#unhide private
Password: private (not shown)
T-HUB4#show private shell
/batm/var/scriptfs # /batm/binux/bin/ub_util -s ahub4_40G yes
Writing field <ahub4_40G> with value <yes>
Success
/batm/var/scriptfs # exit
T-HUB4#config terminal
Entering configuration mode terminal
T-HUB4(config)#system reload manufacturing-defaults
Are you sure that you want to delete existing configuration and
reload manufacturing default configuration (yes/no)? yes
Validation:
The 40G ports are 1/1/1 and 1/2/1. In order to verify the speed, do as follows:

T-HUB4#show port 1/1/1 detailed

```
=========================================================================
Ethernet Interface
=========================================================================
Interface          : 1/1/1
Description        :
Admin State        : up              Port State : down
Config Duplex      : full            Operational Duplex : unknown
Config Speed       : 40000           Operational Speed(Mbps) : unknown
-------------------------------------------------------------------------
Flow Control       : disabled
Dual Port          : No              Active Link : No-Link
-------------------------------------------------------------------------
Default VLAN       :                 MTU[Bytes] : 1544
MAC Learning       :
LAG ID             : N/A
=========================================================================
=========================================================================
Transceiver Data
=========================================================================
Transceiver Type   : Unknown
Cable Connector    : MPO-Parallel-Optic
Vendor Name        : AVAGO              Encoding : SONET-Scrambled
Manufacture Date   : 2010/11/18 - 0 Media : n/a
Serial Number      : QA460230           TX Laser Wavelength: n/a
Part Number        : AFBR-79E4Z-D
Revision Level     : 01Bh
Link Length Support: 5000 for SMF;
=========================================================================---
Transceiver Compliance                     Fibre Channel:
Ethernet   : Unknown                       Media : Unknown
InfiniBAND : 1X-LX                         Tech : Unknown
10G        : Unknown                       Speed : unknown
ESCON      : Unknown                       Length: unknown
SONET      : Unknown
```

Diagnostic:  
- Digital Diagnostic Monitoring: yes  Nominal: 10300
- Internal Calibration: no  Maximum: 1545% above nominal
- External Calibration: no  Minimum: 22866% below nominal
- Average Power Measurement: yes
- Address Change Required: no

Management Port Speed Configuration

Run the following procedure in order to set the speed of a management port. On dual chassis configuration, run this procedure on the SSM of both chassis.

1. Connect to the SSM shell (see SSM160 CLI section)
2. Run the following on the SSM:
   ```
   #config
   #port<port>
   #speed <speed value>
   #commit
   #end
   ```
3. Run `show port <port>` to validate port speed.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **port** | In SSM160 use:  
- 1/5/3 for ethx-mgmt03  
- 1/5/4 for ethX-mgmt04  
In SSM60 use:  
- 1/5/1 for ethx-mgmt01  
- 1/5/2 for ethX-mgmt02 |
| **speed** | Speed value is in Mbps could be 1000/100. |

**Example:**

```
T-HUB4#config
Entering configuration mode terminal
--- WARNING -----------------------------------------------
Running db may be inconsistent. Enter private configuration mode and install a saved configuration.
---
T-HUB4(config)#port 1/5/4
--- WARNING -----------------------------------------------
Running db may be inconsistent. Enter private configuration mode and install a saved configuration.
---
T-HUB4(config-port-1/5/4)#speed 100
--- WARNING -----------------------------------------------
Running db may be inconsistent. Enter private configuration mode and install a saved configuration.
---
T-HUB4(config-port-1/5/4)#commit
% No modifications to commit.
---
T-HUB4(config-port-1/5/4)#end
T-HUB4#show port 1/5/4
```

```
<table>
<thead>
<tr>
<th>Ethernet Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface          : 1/5/4</td>
</tr>
<tr>
<td>Description        :</td>
</tr>
<tr>
<td>Admin State        : up   Port State     : up</td>
</tr>
<tr>
<td>Config Duplex      : auto Operational Duplex : full</td>
</tr>
<tr>
<td>Config Speed       : 100 Operational Speed(Mbps) : 100</td>
</tr>
<tr>
<td>Flow Control       : disabled</td>
</tr>
<tr>
<td>Dual Port          : No   Active Link     : RJ45</td>
</tr>
<tr>
<td>Default VLAN       : 1     MTU[Bytes]     : 1544</td>
</tr>
<tr>
<td>MAC Learning       :</td>
</tr>
<tr>
<td>LAG ID             : N/A</td>
</tr>
</tbody>
</table>
```
Multicast Configuration

Description
Multicast is a method of sending IP datagrams to a group of interested receivers in a single transmission. Multicast group address is used by sources and the receivers to send and receive multicast messages. Sources use the group address as the IP destination address in their data packets. Receivers use this group address to inform the network that they are interested in receiving packets sent to that group. For example, if some content is associated with group 239.1.1.1, the source will send data packets destined to 239.1.1.1. Receivers for that content will inform the network that they are interested in receiving data packets sent to the group 239.1.1.1. The receiver joins 239.1.1.1.

Dynamic Multicast Routing (PIM Dense Mode) Configuration

1. For each interface that uses PIM Dense mode run via gclish:
   
   set pim interface <interface name> on

2. Set PIM mode to Dense. Run via gclish:
set pim mode dense

Validation

Run from gclish: show pim interfaces

Example

> set pim interface eth1-01 on 1_01:
success
> set pim interface eth1-02 on 1_01:
success
> set pim interface eth2-01 on 1_01:
success
> set pim mode dense
1_01:
success
> show pim interfaces
1_01:
Status flag: V - virtual address option enabled
Mode flag: SR - state refresh enabled

<table>
<thead>
<tr>
<th>Interface</th>
<th>Status</th>
<th>State</th>
<th>Mode</th>
<th>DR Address</th>
<th>DR Pri</th>
<th>NumNbrs</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth2-01</td>
<td>Up</td>
<td>DR</td>
<td>dense</td>
<td>2.2.2.10</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>eth1-01</td>
<td>Up</td>
<td>DR</td>
<td>dense</td>
<td>12.12.12.10</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>eth1-02</td>
<td>Up</td>
<td>DR</td>
<td>dense</td>
<td>22.22.22.10</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Static Multicast Routing (SMCRoute) configuration

When working with SMCRoute, dynamic multicast routing should be disabled. The SMCRoute is not included in the OS and should be added manually. Please contact Check Point support.

SMCRoute daemon configuration:

g_all dbset process:smcroute:runlevel 4
g_all dbset process:smcroute:path /bin
g_all dbset process:smcroute:arg:1 -d
g_all dbset :save

Start the SMCRoute daemon

g_all tellpm process:smcroute t

Stopping the SMCRoute daemon

g_all tellpm process:smcroute

g_all /bin/smcroute -k

SMCRoute Routing configuration

To add route:

g_all /bin/smcroute -a <InputIntf> <OriginIpAdr> <McGroupAdr> <OutputIntf> [<OutputIntf>] ...

To remove route:

g_all /bin/smcroute -r <InputIntf> <OriginIpAdr> <McGroupAdr> - remove route

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InputIntf</td>
<td>&lt;InputIntf&gt; can be any network interface as listed by ‘ifconfig’ but not the loopback interface.</td>
</tr>
</tbody>
</table>
### OriginIpAdr
The source IP address of the multicast packets that will be routed by this entry. It is a unicast IP address not a multicast IP address.

### McGroupAdr
The IP address of the multicast group that will be forwarded.

### <OutputIntf> [<OutputIntf>]...
A list of one or more network interfaces to which the multicast packets will be forwarded.

**Example:**
```
g_all /bin/smcroute -a eth2-01 2.2.2.1 225.0.90.90 eth1-01 eth1-02
```

### Multicast restrictions
Multicast access restrictions can be defined on each interface. These restrictions specify multicast groups (that is, addresses or address ranges) to allow or block.

**Configuration**
Open the SmartDashboard and edit the Multicast Restrictions tab:
Go to Gateway Properties > Topology > Add or Edit interface > Multicast Restrictions tab.

![Multicast Restrictions Interface Properties](image)
**Parameter** | **Description**
--- | ---
Drop multicast packets whose destination is in the list | Specifies that outgoing packets from this interface to the listed multicast destinations will be dropped.
Drop all multicast packets except those whose destination is in the list | Specifies that outgoing packets from this interface to all multicast destinations except those listed will be dropped.
Add | Add a Multicast address or address range to the list.
Remove | Remove a selected Multicast address or address range from the list
Tracking | Allows you to choose whether and how to track when multicast packets are dropped.

**Limitations:**
Multicast restriction is not supported on bridge interfaces.

**Multicast acceleration**
Multicast acceleration allows SecureXL to accelerate multicast flow, also in Fan-out scenarios.

**Configuration**
Multicast acceleration is enabled by default.
In order to enable/disable it run from gclish the flowing set of commands:

```
  sim feature mcast_route_v2 {on | off}
fwaccel off
fwaccel on
```

**Limitations**
Multicast acceleration supports IPv4 only.

**Validation and Debugging**
1. Run `fwaccel stat`
Networking

> fwaccel stat
-*= 4 blades: 1_01 1_02 2_01 2_02 --*
Accelerator Status : on
Accept Templates   : enabled
Drop Templates     : disabled
NAT Templates      : enabled
Accelerator Features : Accounting, NAT, Cryptography, Routing,
hasClock, Templates, Synchronous, IdleDetection, Sequencing, TcpStateDetect, AutoExpire,
DelayedNotif, TcpStateDetectV2, CPLS, McastRouting
WireMode, DropTemplates, NatTemplates,
Streaming, MultiFW, AntiSpoofing, DoS Defender,
ViolationStats, Nac, AsychronicNotif,
McastRoutingV2, ConnectionsLimit
Cryptography Features : Tunnel, UDPEndapsulation, MD5, SHA1, NULL,
3DES, DES, CAST, CAST-40, AES-128, AES-256,
ESP, LinkSelection, DynamicVPN, NatTraversal,
EncRouting, AES-XCBC, SHA256

2. Display the accelerator's connections table by running: fwaccel conns
3. Display multicast statistics by running: fwaccel stats -m
4. Enable SIM debug using the command: sim dbg -m drv + routing
example:

the following example disables the feature.

> sim feature mcast_route_v2 off

  --*-- 4 blades: 1_01 1_02 1_03 1_04 --*--

Feature will be disabled the next time acceleration is started/restarted

> fwaccel off

  --*-- 4 blades: 1_01 1_02 1_03 1_04 --*--

SecureXL device disabled.

> fwaccel on

  --*-- 4 blades: 1_01 1_02 1_03 1_04 --*--

SecureXL device is enabled.

> fwaccel stat

  --*-- 4 blades: 1_01 1_02 1_03 1_04 --*--

Accelerator Status : on
Accept Templates : enabled
Drop Templates : disabled
NAT Templates : enabled
Accelerator Features : Accounting, NAT, Cryptography, Routing,
                     HasClock, Templates, Synchronous, IdleDetection,
                     Sequencing, TcpStateDetect, AutoExpire,
                     DelayedNotif, TcpStateDetectV2, CPLS,
                     McastRouting,
                     WireMode, DropTemplates, NatTemplates,
                     Streaming, MultiFW, AntiSpoofing, DoS Defender,
                     ViolationStats, Nac, AsychronicNotif
Cryptography Features : Tunnel, UDPEncapsulation, MD5, SHA1, NULL,
                       3DES, DES, CAST, CAST-40, AES-128, AES-256,
                       ESP, LinkSelection, DynamicVPN, NatTraversal,
                       EncRouting, AES-XCBC, SHA256

DHCP relay
Description

BOOTP/DHCP Relay extends Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) operation across multiple hops in a routed network. In standard BOOTP, all interfaces on a LAN are loaded from a single configuration server on the LAN. BOOTP Relay allows configuration requests to be forwarded to and serviced from configuration servers located outside the single LAN.

BOOTP/DHCP Relay offers the following advantages over standard BOOTP/DHCP:

- You can provide redundancy by configuring an interface on the Check Point system to relay client configuration requests to multiple servers. With this setup, configuration requests are relayed to all the listed servers simultaneously.
- You can provide load balancing by configuring multiple interfaces on the Check Point system to relay client configuration requests to different servers.
- It allows you to centrally manage client configuration across multiple LANs. This is particularly useful in large enterprise environments.

The Gaia implementation of BOOTP Relay is compliant with RFC 951, RFC 1542, and RFC 2131. BOOTP Relay supports Ethernet and IEEE 802 LANs by using canonical MAC byte ordering, that is, clients that specify Bootp htype=1: 802.3 and FDDI.

When an interface configured for BOOTP Relay receives a boot request, it forwards the request to all the servers in its server list. It does this after waiting a specified length of time to see if a local server answers the boot request. If a primary IP is specified, it stamps the request with that address, otherwise it stamps the request with the lowest numeric IP address specified for the interface.

Configuration

Use these commands to configure BOOTP properties for specific interfaces:

```
set bootp interface VALUE off
set bootp interface VALUE primary VALUE wait-time VALUE on
set bootp interface VALUE relay-to VALUE off
set bootp interface VALUE relay-to VALUE on
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface VALUE</td>
<td>Interface name</td>
</tr>
<tr>
<td>primary VALUE</td>
<td>The ip_address to stamp as the gateway address on all BOOTP requests.</td>
</tr>
<tr>
<td>wait-time VALUE</td>
<td>The wait-time value Specifies the minimum amount of time, in seconds, to wait before forwarding a bootp request. Each client-generated bootp request includes the elapsed time since the client began the booting process. The bootp relay does not forward the request until the indicated elapsed time at least equals the specified wait time. This delay provides an opportunity for a local configuration server to reply before attempting to relay to a remote server. Valid values : 0-65535</td>
</tr>
<tr>
<td>relay-to VALUE</td>
<td>The server to which BOOTP requests are forwarded. You can specify more than one server</td>
</tr>
<tr>
<td>off</td>
<td>Disables BOOTP on the specified interface.</td>
</tr>
</tbody>
</table>
Example

> set bootp interface eth1-04 relay-to 20.20.20.200 on

Verification

Use this group of commands to monitor and troubleshoot the BOOTP implementation:

```
show bootp
    interface  - BOOTP/DHCP Relay Interface
    interfaces - All BOOTP/DHCP Relay Interfaces
    network    - BOOTP/DHCP Relay Network
    networks   - All BOOTP/DHCP Relay Networks
    stats      - BOOTP/DHCP Relay Statistics
```

Configuring IPv6 Support (asg dxl ipv6)

Description
Use this command to configure IPv6 support.

Syntax
```
asg dxl ipv6 < enable | disable | verify [-v] >
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>enable</td>
<td>Enables support for IPv6 and updates the distribution matrix size. The distribution matrix is a table containing SGM IDs and used to determine to which other SGMs a packet should be forwarded.</td>
</tr>
<tr>
<td>disable</td>
<td>Disables support for IPv6</td>
</tr>
<tr>
<td>verify</td>
<td>Verifies the current IPv6 status</td>
</tr>
<tr>
<td>[-v]</td>
<td>-v verbose mode shows the current status of IPv6 on all SGMs and notes when the SGMs varies</td>
</tr>
</tbody>
</table>

Example 1
```
asg dxl ipv6 enable
```

Output
```
> asg_dx1 ipv6 enable
You are about to perform enable of IPv6 on blades: all
Reboot may be required.
This may have a negative effect on your performance.
Are you sure? (Y - yes, any other key - no) y
Enable of IPv6 requires auditing
Enter your full name: user
Enter reason for enable of IPv6 [enable IPv6 traffic]:
WARNING: Enable of IPv6 on blades: all, user: user, reason: enable IPv6 traffic
Initiating configuration sequence:
Enabling IPv6 on blades.
Enabling IPv6 on chassis1 SSMs.
Enabling IPv6 on chassis2 SSMs.
Updating blades distribution matrix, Matrix size 2024.
operation finished successfully.
IPv6 enabled
Reboot is required for changes to take effect.
Would you like to save current configuration? (Y - yes, any other key - no) y
You are about to perform reboot on blades: all
It will cause connectivity issue for a period of time
Are you sure? (Y - yes, any other key - no) y
```
Example 2

`asg dxl ipv6 verify`

Output

IPv6 verification:
Collecting information, this may take several seconds.
Collecting process finished, Information gathered from 2 blades.

starting verification.
Verification finished successfully.

System is configured to:
IPv6: enabled

Comments

- The system in configured to Enable IPv6
- 2 SGMs verified as IPv6 enabled

ND Advertisement DoS Attack Defense Mechanism

Neighbor Discovery (ND) is a mechanism used by nodes in an IPv6 network to learn the local topology. Neighbor Discovery is subject to attacks that can:

- Redirect IP packets to unauthorized nodes
- Cause denial of service (DoS)
- Intercept and optionally change packets destined for other nodes.

To minimize threats against the IPv6 Neighbor discovery mechanism, 61000 Security Systems limits the number of the ND Adv packets that can be forwarded to other SGMs by the SGM that receives it. The number is controlled by a threshold value. By default, in a 10 second period no more than 5000 ND advertisements can be forwarded to other SGMs.

To enable the ND Advertisement DoS Attack Defense Mechanism:

Run: `fw ctl set int <parameter> <value>`

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fwha_ch_nda_dos_attack_enabled</code></td>
<td>Enables or disables the ND Adv DoS defense mechanisms</td>
</tr>
<tr>
<td></td>
<td>Possible values:</td>
</tr>
<tr>
<td></td>
<td>0 - Disable</td>
</tr>
<tr>
<td></td>
<td>1 - Enable (default)</td>
</tr>
<tr>
<td></td>
<td>2 - Monitor</td>
</tr>
<tr>
<td></td>
<td>In monitor mode, the attack event is logged but no action taken.</td>
</tr>
<tr>
<td><code>fwha_ch_nda_forwarding_threshold</code></td>
<td>Configures the forwarding threshold, the number of ND Adv packets forwarded to other SGMs during the forwarding interval.</td>
</tr>
<tr>
<td></td>
<td>Default value: 5000</td>
</tr>
<tr>
<td><code>fwha_ch_nda_forwarding_interval</code></td>
<td>Configures the forwarding interval (in milliseconds).</td>
</tr>
<tr>
<td></td>
<td>Default value: 10000</td>
</tr>
</tbody>
</table>
Policy Installation and the Single Management Object

The Single Management Object (SMO), a software technology used to manage 61000 Security Systems gateways, can handle up to 24 SGMs (24 in a dual chassis deployment).

- Under SMO, multiple SGMs have the same management IP address.
- Management tasks such as policy installation and logging are handled by one SGM, called the SMO Master.
- The SMO Master is active SGM with the lowest ID.

During policy installation:
1. The Security Management server installs the policy on the SMO Master.
2. The SMO distributes the policy to all SGMs.
3. Each SGM begins installing the policy locally, and sends and receives policy stage updates to and from the other SGMs. SGMs need to install the policy in a synchronized manner. Policy installation has four stages:
   a) **Policy Started**
      Indicates that Policy installation has started on the local SGM.
   b) **Policy Ready2Finish**
      Local policy installation has completed, but the SGM is waiting for other SGMs to reach the same stage.
   c) **Policy Completed**
      The policy is applied in a way that synchronizes with the other SGMs.
   d) **Enforcing Security**
      The SGM enforces the new policy.

   **Note**: When installing the 61000 Security System, SGMs enforce an initial policy where only the implied rules necessary for management are enforced.

Uninstalling a Policy:

A policy can be uninstalled from the gateway in the following way:

Over a serial connection, run: `asg policy unload`.

```
# asg policy unload
You are about to perform unload policy on blades: all
All SGMs will be in DOWN state, beside local SGM. It is recommended to run the procedure via serial connection
Are you sure? (Y - yes, any other key - no) y
```

Note:
Uninstalling policy from SmartDashboard is not supported.
Installing or Fetching a Policy

A policy can be:

1. **Installed using SmartDashboard (Policy > Install)**
   
2. **Fetched from the Security Management server by running**: `asg policy fetch`:

   ```bash
   # asg policy fetch
   Fetch local operation close all active connections therefore must be performed via serial connection or by policy installation from SmartDashboard.
   Press y to continue, or any other key to exit [n]y
   Installing policy on local SGM..
   Installing policy on local SGM succeed
   Installing policy on remote SGMs, it may take few seconds, run "asg monitor" to monitor the system
   
   Note - This command must be run over a serial connection.
   
Useful Commands

- `asg stat -i tasks`

Use this command to identify the SMO and view how tasks are distributed on the SGMs.

<table>
<thead>
<tr>
<th>Chassis ID: 1</th>
<th>Blade ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task (Task ID)</td>
<td></td>
</tr>
<tr>
<td>SMO</td>
<td>(0) 1(local)</td>
</tr>
<tr>
<td>CH Monitor</td>
<td>(3) 1(local)</td>
</tr>
<tr>
<td>UIPC</td>
<td>(5) 1(local)</td>
</tr>
<tr>
<td>General</td>
<td>(1) 2</td>
</tr>
<tr>
<td>DR Manager</td>
<td>(4) 2</td>
</tr>
<tr>
<td>LACP</td>
<td>(2) 3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis ID: 2</th>
<th>Blade ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task (Task ID)</td>
<td></td>
</tr>
<tr>
<td>CH Monitor</td>
<td>(3) 1</td>
</tr>
<tr>
<td>UIPC</td>
<td>(5) 1</td>
</tr>
<tr>
<td>General</td>
<td>(1) 2</td>
</tr>
<tr>
<td>LACP</td>
<td>(2) 3</td>
</tr>
</tbody>
</table>

- `asg monitor`

Use this command to monitor policy installation.


<table>
<thead>
<tr>
<th>Chassis 1</th>
<th>ACTIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blade ID</td>
<td>Security GW State</td>
</tr>
<tr>
<td>1 (local)</td>
<td>UP</td>
</tr>
<tr>
<td>2</td>
<td>UP</td>
</tr>
<tr>
<td>3</td>
<td>UP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis 2</th>
<th>STANDBY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blade ID</td>
<td>Security GW State</td>
</tr>
<tr>
<td>1</td>
<td>UP</td>
</tr>
<tr>
<td>2</td>
<td>UP</td>
</tr>
<tr>
<td>3</td>
<td>UP</td>
</tr>
</tbody>
</table>

- `asg policy verify`
Use this command to make sure the SGMs have the same policy installed.

<table>
<thead>
<tr>
<th>Blade</th>
<th>Policy Name</th>
<th>Policy Date</th>
<th>Policy Signature</th>
<th>Verification</th>
</tr>
</thead>
</table>

- asg_blade_config pull_policy policy <SGM_sync_ip>

If there is a problem with the policy on one of the SGMs, for example one of the SGMs has the wrong policy, run this command to manually pull a valid policy from a specified SGM.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-if</td>
<td>Enter the name of the interface, such as eth1</td>
</tr>
</tbody>
</table>

### Policy and Configuration Cloning

In the 61000 Security System Gateway, configuration is identical on all SGMs. When SGM goes up, it pulls all configurations from SMO (if there is SMO). If there is no SMO (meaning the SGM goes up while there is no other SGM up), the SGM will run its local configuration.

Configuration includes two parts:
1. FW policy.
2. Set of files as defined under "xfer files list".

"xfer files list" file can be found under "/etc/xfer_file_list" and containing the files that would be pulled during configuration cloning. Each pulled file is matched against the already file available on the machine. The structure of each entry in that file is composed of two parts. First part is a path to the file. Second part is action. The format of the file is explained in "Cloning the Configuration" section.

The configuration cloning is done automatically every time SGM goes up, and can also be done manually by the user (only for troubleshooting).

### Cloning the Firewall Policy

When installing a policy from the Security Management server, the Single Management Object (SMO) distributes the firewall policy to all Security Gateway Modules. Also, each time SGM goes up, it clones the Firewall Policy from the SMO (if there is SMO). If there is a policy problem on a Security SGM, use this command to manually pull and apply (clone) the firewall policy from another SGM:

**Syntax**

```
asg_blade_config pull_config policy <SGM_sync_ip>
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGM_sync_ip</td>
<td>IP address of sync port</td>
</tr>
</tbody>
</table>

**Note** - If necessary, use `asg stat -i all_sync_ips` to obtain a list of all SGM sync ip addresses.

**Example**

```
# asg_blade_config pull_config policy 192.0.2.1
```

### Cloning the Configuration

This clones the firewall policy plus the set of configuration files listed in `/etc/xfer_file_list`. Configuration cloning automatically occurs during a reboot or when these commands are run:
If there is a configuration problem on a Security Gateway Module, use this command to manually pull and apply (clone) the configuration from another Security Gateway Module:

**Syntax**
```
asg_blade_config pull_config all <SGM_sync_ip>
```

**Parameter** | **Description**
--- | ---
SGM_sync_ip | IP address of sync port

**Example**
```
# asg_blade_config pull_config all 192.0.2.1
```

**Note** - If necessary, use `asg stat -i all_sync_ips` to obtain a list of all Security Gateway Module sync ip addresses.

**Explanation**
All Security Gateway Modules maintain a local configuration. The local configuration consists of the firewall policy and the set of files listed in `/etc/xfer_file_list`. The `xfer_file_list` file has this format:

<table>
<thead>
<tr>
<th>Path to file</th>
<th>File name</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>$FWDIR/modules/</td>
<td>fwkern.conf</td>
<td>/bin/false</td>
</tr>
</tbody>
</table>

Each line describes a path to a configuration file, in this case `fwkern.conf`, followed by an action to take if the "pulled" file is different from the local file. (When you clone the configuration, the firewall policy and configuration files are "pulled" from the specified SGM and matched against the local versions). The action attributed to each file in the list can be:

<table>
<thead>
<tr>
<th>Action</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>/bin/false</td>
<td>Reboot immediately after cloning completes</td>
</tr>
<tr>
<td>/bin/true</td>
<td>No reboot required.</td>
</tr>
</tbody>
</table>

If the entry does not specify `/bin/true` or `/bin/false` for a given file, a "callback script" decides on the necessary action.

---

**MAC Addresses and Bit Conventions**

**MAC Addresses**
MAC addresses divide into three types:

- **BMAC.** A MAC address assigned to all interfaces with the "BPEthX" naming convention. Unique per member. It does not rely on the interface index number.
- **VMAC.** A MAC address assigned to all interfaces with "ethX-YZ" naming convention. Unique per chassis, it does not rely on the interface index number.
- **SMAC.** A MAC address assigned to Sync interfaces. Unique per member, it does not rely on the interface index number.

**Bit Conventions**

**BMAC**

- 1 - 1 bit stating if this address is BMAC/SMAC(0) or VMAC(1) to avoid possible collision with VMAC space.
- 2,...,8 - 7 bits that state the member ID (starting from 1) - limited to 127 members
MAC Addresses and Bit Conventions

- 9,...,13 - zero bits.
- 14 - 1 bit stating if this address is BMAC(0) or SMAC(1) to avoid possible collision with SMAC space.
- 15,16 - 2 bits that state the absolute interface number (taken from interface name: i.e. in BPEthX, X is the interface number - limited to 4 interfaces.)

**SMAC**
- 1 - 1 bit stating if this address is BMAC/SMAC(0) or VMAC(1) to avoid possible collision with VMAC space.
- 2,...,8 - 7 bits that state the member ID (starting from 1) - limited to 127 members.
- 9 - 1 bit stating whether it is Sync1(0) or Sync2(1).
- 9,...,13 - zero bits.
- 14 - 1 bit stating if this address is BMAC(0) or SMAC(1) to avoid possible collision with BMAC space.
- 16 - 1 bit stating whether it is Sync1(0) or Sync2(1).

**VMAC**
- 1 - 1 bit stating if this address is BMAC/SMAC(0) or VMAC(1) to avoid possible collision with BMAC/SMAC space.
- 2,...,3 - 2 bits to indicate chassis id (starting from 0) - limited to 4 boxes.
- 4,...,8 - 5 bits to indicate switch number - limited to 32 switches.
- 9,...,16 - 8 bits to indicate port number - limited to 256 ports per switch.

### Verifying the MAC Address (asg_mac_resolver)

**Description**

All three types of MAC address (BMAC, VMAC, SMAC) can be verified using the `asg_mac_resolver` utility. From the given MAC address, `asg_mac_resolver` determines the:

- MAC type
- Chassis ID
- SGM ID
- Assigned interface

**Syntax**

```
asg_mac_resolver <MAC Address>
```

**Example**

```
asg_mac_resolver 00:1C:7F:01:00:FE
```

**Output**

```
[00:1C:7F:01:00:FE, BMAC] [Chassis ID: 1] [SGM ID: 1]
[Interface: BPEth0]
```

**Explanation**

- The given MAC Address was taken from the interface BPEth0, within SGM 1 on Chassis 1.
- Assuming 00:1C:7F:XY:ZW:FE is the structure of the MAC address, MAC magic attribute is denoted by FE.
- INDEX are 16 bits (2 Bytes) denoted by XY:ZW 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16.
SyncXL

Description:

SyncXL mechanism is the Sync methodology that is used by the 61000 Security System. SyncXL assures that each connection is synched only to one backup SGM on the local Active chassis and to one backup SGM on the Standby chassis in Dual Chassis configuration. The backup on the local Active chassis is the SGM that handles the connection if the original SGM that recorded the connection fails (an SGM failover event). The backup on the Standby chassis is the SGM that handles the connection in a chassis HA failover event. The goal of SyncXL is to enable performance and connection capacity scalability when adding more SGMs to the system.

Whenever SGM or the Chassis changes state, all SGMs need to update their backup according to the updated State. The process of updating the active and backup SGMs is triggered automatically after the following events:

- **Failure of SGM** – connections that had a backup connection on that SGM needs to be synced to another backup SGM
- **Recovery of SGM** – the new recovery SGM can now act as a backup for connections that are active on other SGMs
- **Chassis HA failover** – when the Active chassis fails, the Standby chassis becomes Active and a backup entry is placed for each of the connections it handles.

The SyncXL mechanism can be configured via the `asg_sync_manager`. See the `asg_sync_manager` section.

Sync to Standby Chassis - Active SGMs ratio:

To cope with load and capacity, the Standby Chassis is required to have 50% of SGMs that are UP compared to that number on the Active Chasssis. For example: if on the Active chassis there are 10 SGMs that are UP, while on the Standby chassis there are less than 5 SGMs that are UP, sync between chassis is automatically disabled.

The ratio parameter can be configured via `asg_sync_manager`. See the `asg_sync_manager` section.

Verification:

- Use `asg search` utility to make sure each connection has backups on the local chassis and on the other chassis when working on dual chassis configuration
- The name of the process that chooses a backup SGM for each connection in the system is “iterator”. Statistic on the last operation of the iterator can be collected with the command “bcstats iterator”

```
[Expert@hpd_55-ch01-01]# bcstats iterator
Last Iterator Statistics:
-------------------------------------
Start time: Thu Sep 13 10:48:18 2012
Running time: 0 Seconds
Status: Finished
Reason: Chassis ID 2 state was changed to STANDBY
Total connections iterated 38
Connections w/ sync action 0
```

Exceptions:

- VOIP connections are synced to all SGMs
- Local connections (ssh from/to 61000 Security System pseudo IP) are not synced.
- SyncXL does not work on the Sync interface and on the Management Interfaces
Security Group (asg security_group)

Description
To be part of the Security Gateway, an SGM must belong to the Security Group. SGMs are added to the Security group using the `asg security_group` command. SGMs in the security group:

- Are selected during the initial installation procedure (after running: `#setup`)
- Are automatically installed once installation of the first SGM has completed
- Can be changed by using the `asg security_group` command

Syntax
`asg security_group`

Example
`asg security_group`

Output
```bash
> asg security_group
+--------------------------------------+
|       Security Group Utility         |
+--------------------------------------+
Current Security Group:
+-------------------------+------------------+
| Chassis | Security Gateway Modules |
+-------------------------+------------------+
| 1 | 1,2,3 |
| 2 | 1,2,3 |
+-------------------------+------------------+
Choose one of the following options:
1) Add SGMs to Security Group
2) Remove SGMs from Security Group
3) Exit
```

Comments
Select which SGMs should be added or removed from the security group. Note that:

- An SGM added to the security group automatically joins the single management object of the Security Gateway and then reboots
- Before you remove an SGM from the security gateway, make sure that its `state` is DOWN.
- To optimize connection distribution amongst the SGMs, keep the security group updated with the actual number of SGMs in the appliance.

⚠️ Important - Run: `asg security_group verify` to make sure that the security group is correctly configured.
Layer-2 Bridge Mode configuration

Description:
A security gateway in bridge mode operates as a regular firewall, inspecting traffic and dropping or blocking unauthorized or unsafe traffic, and is invisible to all Layer-3 traffic. When authorized traffic arrives at the gateway, it is passed from one interface to another through a procedure known as bridging. Bridging creates a Layer-2 relationship between two interfaces, whereby any traffic that enters one interface always exits the other. This way, the firewall can inspect and forward traffic without interfering with the original IP routing.

Chassis HA in L2 Bridge mode
61000 Dual chassis in L2 act in Active-Standby mode, where Active chassis pass STP and Standby chassis drops it. 61000 maintain MAC shadow table which caches the learned MAC passing via the system. When Chassis failover occurs, the new Active chassis generates advertisement packets with the cached MAC, letting remote switches learn the MAC, and start to pass STP.

**Pre Configuration:**

**Dual Chassis**

If working in **Dual Chassis**, update module global parameter: (in Single chassis - this is not needed)

```
g_update_conf_file fwkern.conf fwha_active_standby_bridge_mode=1
```

Reboot

**Bridge with SSM60**

In SSM60, in order to support bridge mode, you need to enable `bridge_mode_on_ssm60` sim kernel variable:

```
g_update_conf_file simkern.conf bridge_mode_on_ssm60=1
```

Reboot

**Bridge with VLAN Trunk**

If working with bridge interface that pass VLAN trunks, make sure vlan performance enhancement is enabled:

```
g_vlan_perf_enhancement -s
```

**Configuration:**

Create bridge interface:

```
add bridging group <bridge_id>
```

Add interfaces to bridge:

```
add bridging group <bridge_id> interface <IF_NAME>
```

**Verification:**

```
asg_br_verifier – please refer to asg_br_verifier section
```

**GUI Configuration**

Bridge IP address should have no IP, hence "Get Topology" from the GUI will not identify bridge interfaces.

In order to define bridge interface in the GUI:

Go to topology, click "Add", enter bridge interface manually:
Layer-2 Bridge Mode configuration

Check Point Gateway - nir_mrgdn

- General Properties
- Topology
- NAT
- SecurePlatform
- Identity Authentication
- Legacy Authentication
- Logs and Masters
- Capacity Optimization
- Cooperative Enforcement
- Advanced

Topology

<table>
<thead>
<tr>
<th>Name</th>
<th>Network Type</th>
<th>IP Address</th>
<th>Network Mask</th>
<th>Topo</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth1-Mgmt4</td>
<td>External</td>
<td>172.16.5.72</td>
<td>255.255.255.0</td>
<td>Ext2</td>
</tr>
</tbody>
</table>

VPN Domain

- All IP Addresses behind Gateway are based on Topology information
- Manually defined CP_default_Office_Mode_addresses

Set domain for Remote Access Community...
Layer-2 Bridge Mode configuration

Check Point Gateway - nir_armgdn

Topology

Interface Properties

General  Topology  Multicast Restrictions

Name:

IP Address:

Net Mask:

Note: the interface name must exactly match the name the operating system uses for this interface. See help for further information.
If configured more than 1 bridge interface, GUI will warn:

This message can be safely ignored.
**Distribution Modes**

Distribution mode refers to the way in which an SSM disperses incoming traffic to SGMs. An SSM supports four distribution modes:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
</table>
| User     | • In user mode, the SGM that receives the packet is determined by the connection destination.  
           • User mode applies to a specified SSM.                                           |
| Network  | • In network mode, the SGM that receives the packet is determined by the connection source.  
           • Network mode applies to a specified SSM.                                          |
| General  | • In general mode, the SGM that receives the packet is determined by the connection source and destination.  
           • General mode applies to all SSM in the 61000 Security Systems.                  |
| Per-port | In per-port mode, each port on the SSM is configured separately to user mode or network mode. |

**Note**

- Despite having four distribution modes, User/Network is considered one mode as the two modes work together.
- The configuration of the first SSM must match the configuration of the second. For example, if the first SSM is User/Network the second SSM must also be User/Network.

**User/Network Mode**

By default, the distribution mode is derived from the interfaces Topology as configured in SmartDashboard:

- Data interfaces configured as Internal are set to User Mode.
- Data interfaces configured as External are set to Network Mode.

For example:

<table>
<thead>
<tr>
<th>SSM</th>
<th>Interface</th>
<th>Configured in SmartDashboard</th>
<th>Topology as:</th>
<th>Distribution Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Eth1-01</td>
<td>Internal</td>
<td></td>
<td>User</td>
</tr>
<tr>
<td></td>
<td>Eth1-03</td>
<td>Internal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Eth2-02</td>
<td>External</td>
<td></td>
<td>Network</td>
</tr>
<tr>
<td></td>
<td>Eth2-03</td>
<td>External</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Responding to the topology defined in SmartDashboard, the system sets SSM1 to *User mode* and SSM2 to *Network mode*. The system as a whole is in *User/Network* mode. These two modes work together.

**Note** -
- If all the data interfaces are set to internal, the system is still considered as being in user/network distribution mode, even though SSM1 and SSM2 are both set to User mode.
- Management and sync interfaces, not shown in the SmartDashboard **Topology** table, are not set in this way. An automatic mechanism attempts to set the distribution mode for management interfaces per SSM as either user or network.

**General Mode**

General mode can only be configured manually using the `asg dxl dist_mode` command ("Manually Configuring Distribution Modes (asg dxl dist_mode)" on page 200).

To cancel general mode:

- Use the `asg dxl dist_mode set` to change to a different distribution mode, or:
- Use `asg dxl dist_mode policy_control` to cancel the current mode and then reconfigure the interfaces as either external or internal using the **Topology** page in SmartDashboard.

**Note** -
- If you use the `asg dxl dist_mode` command, you redefine SSM interfaces (and related SGM interfaces) to be internal or external. However, this change is not reflected in the SmartDashboard **Topology** for the gateway.
- To see the updated configuration, run: `asg dxl dist_mode get -v`.

**Per-port Mode**

If you configure the links this way in SmartDashboard:

<table>
<thead>
<tr>
<th>SSM</th>
<th>Interface</th>
<th>Configured as:</th>
<th>Distribution Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Eth1-01</td>
<td>Internal</td>
<td>Per port</td>
</tr>
<tr>
<td></td>
<td>Eth1-03</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Eth2-02</td>
<td>Internal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Eth2-03</td>
<td>External</td>
<td></td>
</tr>
</tbody>
</table>

The 61000 Security Systems is now in *per port* distribution mode. Each port on the SSM is configured separately to *internal* or *external*. The distribution mode is still *per port* if the interfaces are configured this way:

<table>
<thead>
<tr>
<th>SSM</th>
<th>Interface</th>
<th>Configured as:</th>
<th>Distribution Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Eth1-01</td>
<td>Internal</td>
<td>Per port</td>
</tr>
<tr>
<td></td>
<td>Eth1-03</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Eth2-02</td>
<td>External</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Eth2-03</td>
<td>External</td>
<td></td>
</tr>
</tbody>
</table>

Even though the interfaces on SSM2 are configured identically, as external, its distribution mode is not *user/network* but *per port* because of the configuration on SSM1.
Manually Configuring Distribution Modes (asg dxl dist_mode)

Description
Use this command to manually configure the distribution mode: the way in which an SSM disperses incoming traffic to SGMs.

Syntax
asg dxl dist_mode <get | set | verify | policy_control | help>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Shows current distribution mode.</td>
</tr>
<tr>
<td>get -v</td>
<td>Verbose output that details the configuration of each interface active on the SSM.</td>
</tr>
<tr>
<td>set</td>
<td>Sets a new distribution mode.</td>
</tr>
<tr>
<td>set -f</td>
<td>Forcefully sets the distribution mode.</td>
</tr>
<tr>
<td>verify</td>
<td>Verifies the current distribution mode.</td>
</tr>
<tr>
<td>verify -v</td>
<td>Shows all available system data on the current distribution mode.</td>
</tr>
<tr>
<td>policy_control</td>
<td>Cancels the current distribution mode. Use this parameter if you want to change the interface Topology in SmartDashboard.</td>
</tr>
</tbody>
</table>

Example 1  
asg dxl dist_mode get

Output
Distribution mode: user/network
SSM1: [mode: user, origin: policy]
SSM2: [mode: network, origin: policy]

Comments
- The system is configured to the user/network distribution mode
- SSM1 is set to User
- SSM2 is set to Network.
- Origin identifies the source of the configuration. If the origin is:
  - Policy
    The current distribution mode derives from the Topology as configured in SmartDashboard
  - Manual
    The current distribution mode is a result of the administrator running asg dxl dist_mode set.

Example 2  
asg dxl dist_mode get -v
Output

Distribution mode: user/network

SSM1: [mode: user, origin: policy]
  1) eth1-01: user [origin: policy]

SSM2: [mode: network, origin: policy]
  1) eth2-01: network [origin: policy]

Comments

Shows the configuration of each interface active on the SSM.

Setting the distribution mode:

1. From gclish, run `asg dxl dist_mode set`.

   The distribution mode configuration menu opens:

   ```
   distribution mode configuration:
   
   Current SSM1 distribution mode: user.
   Current SSM2 distribution mode: network.
   
   Please choose one of the following distribution modes:
   1) User/Network
   2) General
   3) Per port
   4) Exit
   >
   ```

   - If you decide on the **User/Network** distribution mode, you need to set the distribution mode for each SSM separately:

     ```
     configure distribution mode for SSM1.
     1) Network
     2) User
     >
     ```

     configure distribution mode for SSM2.

     ```
     1) Network
     2) User
     >
     ```

   - If you select **General**, the system:
     - Asks for confirmation
     - Sets the mode
     - Exits the configuration menu

   - If you select **Per port**, you need to configure the mode for each interface on an SSM:

     ```
     chosen system distribution mode: Per port.
     
     Setting SSM1 to distribution mode: port.
     
     Interfaces configuration:
     1) eth1-01:
        Distribution mode: User
        configure eth1-01 distribution mode: [currently: User]
        1) Network
        2) User
        >
     ```

2. When prompted, confirm the new configuration.

Reconfiguring Distribution Modes

Distribution modes can be reconfigured using:
- 
  asg dxl dist_mode set command ("Manually Configuring Distribution Modes (asg dxl dist_mode)" on page 200)

For manual configuration.

- **SmartDashboard Gateway Properties > Topology**

  Automatic configuration. Let the system derive the distribution mode from the gateway topology as defined in SmartDashboard.

If you want to reconfigure the distribution mode after configuring it manually using the asg dxl dist_mode set command, you first need to cancel the current configuration.

**To cancel the current distribution mode configuration:**

1. From gclish, run: asg dxl dist_mode policy_control.

   
   asg dxl dist_mode policy_control
   Operation completed successfully, use topology to set new distribution modes.
   If you choose not to, rerun "asg_dx1 dist_mode set -f"

2. To make sure that the distribution mode has been set to derive from the Topology as configured in SmartDashboard, run: asg dxl dist_mode verify. The origin text in the output should read policy.

   Distribution mode verification:
   Collecting information, this may take several seconds.
   Collecting process finished. Information gathered from 5 SGMs.
   starting verification.
   Verification finished successfully.
   System is configured to:
   single chassis
   Dual SSMs
   IPv6: disabled
   Matrix size: 1024
   Distribution mode: User/Network

   SSM1: [mode: user, origin: policy]
   SSM2: [mode: user, origin: policy]

   current distribution mode setting not optimal.
   suggest turning one of the SSMs to distribution mode: network.

   The origin should be policy.

**Two Alternative ways of making sure the correct distribution mode is set:**

- Run asg dxl dist_mode verify -v.

   Distribution modes per SSM shows in the output.

   | SSM1: [mode: user, origin: policy] | SSM2: [mode: user, origin: policy] |

   - When manually configuring the distribution mode, a file (dist_mode.conf) file is created (in /var/opt/CPsuite-R75/fw1/conf) and synchronized between the SGMs.

   The file has entries similar to these:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth1-01=1</td>
<td>=1 means eth1-01 is set to the User distribution mode.</td>
</tr>
<tr>
<td>eth1-02=0</td>
<td>=0 means eth1-02=0 is set to network distribution mode.</td>
</tr>
<tr>
<td>eth1-03=1</td>
<td>=1 means eth1-03 is set to the User distribution mode</td>
</tr>
</tbody>
</table>

  - SSM1 is set to per port because the interfaces are not set to the same distribution mode.
  - The SSM1 configuration sets SSM2 to per port as well, even though the interfaces on SSM2 are set to the identical distribution mode:
Chassis Control (asg_chassis_ctrl)

Description

Based on SNMP requests, chassis control is the mechanism by which SGMs communicate with SSMs and CMMs. This SNMP-based communication can be used to:

- Automatically monitor hardware components ("Showing Hardware Information for Monitored Components (asg hw_monitor)" on page 49).
- Manually configure and monitor SSM and CMMs using commands available in the chassis control utility.

Note: While you can configure SGMs using this utility, it is recommended to use the more comprehensive asg dxl command.

Syntax

asg_chassis_ctrl

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>active_sgms</td>
<td>Prints a list of active SGMs.</td>
</tr>
<tr>
<td>active_ssm</td>
<td>Prints the active SSM. An SSM not installed on the chassis or shutdown is considered inactive.</td>
</tr>
<tr>
<td>get_fans_status</td>
<td>Prints the current status of the chassis fans.</td>
</tr>
<tr>
<td>get_lb_dist</td>
<td>Prints the current distribution matrix from the given SSM. The matrix is a table containing SGM IDs, and used to determine to which other SGMs a packet should be forwarded.</td>
</tr>
<tr>
<td>get_ssm_firmware</td>
<td>Gets the firmware version of the given SSM.</td>
</tr>
<tr>
<td>get_ssm_config</td>
<td>Gets the configuration name of the given SSM.</td>
</tr>
<tr>
<td>get_ssm_type</td>
<td>Gets the type the given SSM</td>
</tr>
<tr>
<td>get_psu_status</td>
<td>Prints the current status of the power supply units.</td>
</tr>
<tr>
<td>get_pems_status</td>
<td>Print the current status of the chassis PEM units</td>
</tr>
<tr>
<td>get_cmm_status</td>
<td>Prints the current status of the CMM(s).</td>
</tr>
<tr>
<td>get_cpus_temp</td>
<td>Prints temperatures of the given SGM's CPUs.</td>
</tr>
<tr>
<td>get_dist_md5sum</td>
<td>Print the md5sum of the distribution matrix for the given SSM. Comparing this checksum against the checksum on other SSM verifies that they are in sync.</td>
</tr>
<tr>
<td>update_lb_from_db</td>
<td>Update SSMs according to the local database.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>enable_port</td>
<td>Administratively enables the given port on the SSM.</td>
</tr>
<tr>
<td>disable_port</td>
<td>Administratively disables the given port on the SSM.</td>
</tr>
<tr>
<td>get_ports_stat</td>
<td>Prints the port status of the given SSM.</td>
</tr>
<tr>
<td>set_port_speed</td>
<td>Sets the port speed</td>
</tr>
<tr>
<td>set_dist_mode</td>
<td>Sets the distribution mode to all ports of the given SSM. There are four</td>
</tr>
<tr>
<td></td>
<td>distribution modes: User, Network, General, and per port. The distribution</td>
</tr>
<tr>
<td></td>
<td>mode affects the way an SSM distributes traffic among the SGMs.</td>
</tr>
<tr>
<td>set_dist_mask</td>
<td>Sets the number of bits to be considered when calculating distribution. The</td>
</tr>
<tr>
<td></td>
<td>number of bits is derived from the distribution mode.</td>
</tr>
<tr>
<td>get_dist_mode</td>
<td>Print the ports distribution mode of the given SSM</td>
</tr>
<tr>
<td>get_dist_mask</td>
<td>Gets a summary of the distribution masks in the different modes.</td>
</tr>
<tr>
<td>get_matrix_size</td>
<td>Prints the SSM distribution matrix.</td>
</tr>
<tr>
<td>get_sel_info</td>
<td>Gets SEL data from a CMM. SEL is the event log of a CMM, used in</td>
</tr>
<tr>
<td></td>
<td>troubleshooting and system forensics.</td>
</tr>
<tr>
<td>restart_ssm</td>
<td>Restarts the given SSM.</td>
</tr>
<tr>
<td>restart_cmm</td>
<td>Restart the given CMM</td>
</tr>
<tr>
<td>start_ssm</td>
<td>Starts the given SSM.</td>
</tr>
<tr>
<td>shutdown_ssm</td>
<td>Shuts down the given SSM.</td>
</tr>
<tr>
<td>mib2_stats</td>
<td>Gets MIB2 statistics for the given SSM.</td>
</tr>
<tr>
<td>get_bmac</td>
<td>Gets blade MACs from SSM.</td>
</tr>
<tr>
<td>ipv6_enable</td>
<td>Enables IPv6 mode.</td>
</tr>
<tr>
<td>ipv6_disable</td>
<td>Disables IPv6 mode.</td>
</tr>
<tr>
<td>ipv6_status</td>
<td>Print IPv6 status.</td>
</tr>
<tr>
<td>help [-v]</td>
<td>Print help messages in [-v] verbose mode.</td>
</tr>
</tbody>
</table>
Comments

- To view the usage for any parameter, issue the parameter without any additional flags.

- For parameters which require an SSM ID, the `all` flag can be used to run the command on all SSMs.

- SNMP messaging between the SGMs, SSMs and CMMs can be configured using `gclish`. For example:
  
  ```
  > show chassis id 1 module SSM1 ip
  > show chassis id all module SSM2 status
  > set chassis id 2 general snmp_retries 3
  ```

- To make sure that the Chassis Control mechanism functions properly, run `asg_chassis_ctrl` for each of the chassis modules:

  ```
  > asg_chassis_ctrl get_cmm_status
  Getting CMM(s) status
  CMM #1 -> Health: 0, Active: 0
  CMM #2 -> Health: 1, Active: 1
  > asg_chassis_ctrl get_ssm_firmware all
  Firmware version of SSM1 is 7.5.18
  Firmware version of SSM2 is 7.5.18
  ```

  If both commands succeed, it means that the chassis control utility is able to communicate with the CMMs and SSMs.
Hybrid System

Description:
The number of physical cores on an SGM dictates the number of firewall and ppak instances that will run on the SGM (1 core per instance). For example, an SGM with 8 physical cores might have 4 instances of firewall and 4 instances of ppak, on the other hand an SGM with 12 physical cores might have 8 instances of firewall and 4 instances of ppak.
The number of firewall and ppak instances must be identical on all SGMs in order for the system to work properly. The hybrid systems mechanism allows the 61000 Security System to work with SGMs which have different numbers of physical cores.

When an SGM boots up, as part of the configuration cloning, it tries to adjust its instances number to the current instances number in the system (which is dictated by the SGM it clones the configuration from – usually the SMO). If the booting SGM has enough physical cores to match the other SGMs, then it will complete the boot process successfully and will go to “UP” state (note that some cores may remain unutilized). If on the other hand, the booting blade does not have enough physical cores to match the configuration, then it will remain in “DOWN” state and will have a “Cores” PNote (Problem Notification). To “fix” an SGM with the “Cores” PNote it must be rebooted, in order to try again to match the instance configuration in the system.

Configuration:
In order to manually configure the firewall instances number, run:
```
# cpconfig corexl instances [n]
```

n – the number of desired firewall instances

Note: The number of ppak instances will be automatically derived from the firewall instance configuration.

Verification:
In order to display the cores and instances information in the system, run:
```
# asg_cores_stat
```

GARP chunk mechanism

Description:
When Proxy ARP is configured, the Firewall responds to ARP requests for hosts other than itself. Whenever a chassis failover occurs, the new chassis sends GARP's with its own (new) MAC address in order to update the network with the new MAC.
In order to avoid GARPs flooding whenever chassis failover occurs, the GARPs are sent in chunks. The chunk mechanism is iterating on the proxy ARP IPs, and each time sends GARPs only for some of them until it completes the entire list.

In each HA Time Unit (HTU=0.1s) (=0.1 second) - a chunk of the GARP list is sent.
Whenever the iteration is finished send all the list, it waits N HTU and sends the list again.

Configuration:
In each HTU - a chunk of the GARP list is sent.
For example, if we want that 10 GARPs will be sent in each second
```
fwha_refresh_arps_chunk should be set to 1.
(command: # fw ctl set int fwha_refresh_arps_chunk 1)
```
For 50 GARPs/seconds,
```
fwha_refresh_arps_chunk should be set to 5.
(command: # fw ctl set int fwha_refresh_arps_chunk 5)
```
Whenever the iteration is finished sending GARP for the entire list, it waits N HTU and re-sends the GARP again. The time between the iterations can be configured with:

- `fwha_periodic_send_garps_interval1 = (1 HTU) /* should not be changed, send immediately after failover */`
- `fwha_periodic_send_garps_interval2 = (10 HTU) /* 01 seconds */`
- `fwha_periodic_send_garps_interval3 = (20 HTU) /* 02 seconds */`
- `fwha_periodic_send_garps_interval4 = (50 HTU) /* 05 seconds */`
- `fwha_periodic_send_garps_interval5 = (100 HTU) /* 10 seconds */`

In the above (default) configuration, after finishing iterate the list:
- wait 1 seconds and start send again
- wait 2 seconds and start send again.
- wait 5 seconds and start send again.
- wait 10 seconds and start send again.

To change interval:

```
fw ctl set int fwha_periodic_send_garps_interval<1-5> 1
```

To apply intervals:

```
fw ctl set int fwha_periodic_send_garps_apply_intervals 1
```

**Verification:**

In order to initiate manual garp sending:

On the chassis monitor blade, run:

```
fw ctl set int test_arp_refresh 1
```

This will cause garp sending (same as was failover)

**Debug:**

```
fw ctl zdebug -m cluster + ch_conf | grep fw_refresh_arp_proxy_on_failover
```

---

**Hardware Reference**

**Chassis Management Module (CMM) CLI**

The Chassis Management Module (CMM) monitors and controls hardware modules in the chassis. Communication with a CMM occurs via SNMP requests from the SMO SGM. If a hardware sensor reports a problem, the CMM automatically takes action or sends a report. CMMs also have a Command Line Interface.

There are two methods to connect to a CMM CLI:

1. **Connect to the serial port on the front panel of the CMM**
   a. In your terminal emulation program, set the baud rate to 9600
   b. Enter `admin` for the user name and password

2. **Open a telnet or SSH session from one of the SGMs**
   a. First make sure that you have connectivity to the CMMs by pinging both addresses:
      i. `198.51.100.33` (routed via SSM1)
      ii. `198.51.100.233` (routed from SSM2)
   b. Telnet or SSH from the SGM to the CMM
c. Enter **admin** for the user name and password

**When connected:**

- Modify the chassis configuration, including the chassis ID (1 or 2) by editing
  
  `/etc/shmm.cfg`

- Useful commands: (Should be run after entering CLI shell, via **clia** command)

- Display full list of commands:

  `CLI> help`

- Show alerts

  `CLI> alarm`

- Reset alerts

  `CLI> alarm 0`

- Show power consumption information

  `CLI> shelf pd`

- Retrieve event logs

  `CLI> sel`

- Reboot the CMM

  `CLI> reboot`

  **Note**: reboot initiates a failover to the standby CMM

- Check if a board is recognized in a slot

  `CLI> board`

- Reset the specified board

  `CLI> boardreset <slot number>`

- Display FRU information

  `CLI> fru [fru_id]`

**Example**: SGM 5 will be displayed as 8a on CMM

<table>
<thead>
<tr>
<th>Physical slot</th>
<th>IPMB</th>
<th>SGM/SSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9a</td>
<td>SGM1</td>
</tr>
<tr>
<td>2</td>
<td>96</td>
<td>SGM2</td>
</tr>
<tr>
<td>3</td>
<td>92</td>
<td>SGM3</td>
</tr>
<tr>
<td>4</td>
<td>8e</td>
<td>SGM4</td>
</tr>
<tr>
<td>5</td>
<td>8a</td>
<td>SGM5</td>
</tr>
<tr>
<td>6</td>
<td>86</td>
<td>SGM6</td>
</tr>
<tr>
<td>7</td>
<td>82</td>
<td><strong>SSM1</strong></td>
</tr>
<tr>
<td>8</td>
<td>84</td>
<td><strong>SSM2</strong></td>
</tr>
<tr>
<td>9</td>
<td>88</td>
<td>SGM7</td>
</tr>
<tr>
<td>10</td>
<td>8c</td>
<td>SGM8</td>
</tr>
<tr>
<td>11</td>
<td>90</td>
<td>SGM9</td>
</tr>
<tr>
<td>12</td>
<td>94</td>
<td>SGM10</td>
</tr>
<tr>
<td>13</td>
<td>98</td>
<td>SGM11</td>
</tr>
<tr>
<td>14</td>
<td>9c</td>
<td>SGM12</td>
</tr>
</tbody>
</table>

**CMM debug commands – How to activate the log function:**
• Log into the active ShMM
• Run
  o /etc/summary
can take several minutes
  o cat /tmp/debug.log
prints debug log with all basic information
  o i2c_test
tests the internal ShMM I2C and prints all devices connected on the I2C
  o cat /etc/shmm.cfg
prints ShMM’s custom configuration
  o clia fruinfo 20 x
17 times where x is 0 to 16
  o clia fruinfo y 0
16 times where y is 10,12,82,84,86,88,8a,8c,90,92,94,96,9a,9c
• Close your terminal program. /tmp/debug.log file will hold the debug information.

Security Switch Module (SSM) CLI

SSM60 CLI

The Security Switch Module (SSM):
• Distributes network traffic to the Security Gateway Modules (SGMs)
• Forwards traffic from the SGMs to the network
• Shares the load amongst the SGMs

Communication between the SSMs and SGMs occurs automatically via SNMP requests, but you can also connect directly to the SSM and run commands.

There are two ways to connect to the SSM CLI:
• Connect to a serial port on the front panel of the SSM.
  The SSM60 has two serial ports, one for the fabric switch (data ports) and one for the base switch (management ports).
  o In your terminal emulation program, set the baud rate to 9600.
  o Enter admin for the password.
  o Enter enable. This gives read and write permissions to the system. Not entering enable results in read-only permissions.
  o Enter ? for a list of available commands and usage.
  
Note - Load balancing commands are run on the fabric switch only.

• Open a telnet session from one of the SGMs.
  o First make sure that you have connectivity to the SSMs by pinging these addresses:
### SSM160 CLI

**Description**
The SSM (Security Switch Module) is the networking module of the gateway. The SSM transmits traffic to and from the SGM and performs the load distribution among the SGMs. The SSM includes two modules:
- Fabric switch - includes the Data ports
- Base switch - includes the Management ports.

Most of the communication with the SSM is done automatically by SNMP requests from the SGM but on some events connecting directly to the SSM can be useful.

**Configuration**
Connection to the SSM CLI can be established in two ways:
1. The administrator can connect with a serial console to the “CLI” port on the SSM front panel (baud rate 9600).
2. From one of the SGMs use SSH to connect to the SSM.
   The SSM IPs can be retrieved from CLISH/GCLISH:
   ```
   SHOW CHASSIS ID<1|2|ALL> MODULE SSM<1|2> IP
   ```

---

<table>
<thead>
<tr>
<th>SSM</th>
<th>Switch</th>
<th>IP address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Base</td>
<td>198.51.100.31</td>
</tr>
<tr>
<td></td>
<td></td>
<td>198.51.100.32</td>
</tr>
<tr>
<td>2</td>
<td>Base</td>
<td>198.51.100.231</td>
</tr>
<tr>
<td></td>
<td></td>
<td>198.51.100.232</td>
</tr>
</tbody>
</table>

- Telnet from the SGM to the SSM
- Enter admin for the password.
- Enter enable. This gives read and write permissions to the system. Not entering enable results in read-only permissions.
- Enter ? for a list of available commands and usage.

When connected, use these useful troubleshooting commands:

<table>
<thead>
<tr>
<th>To</th>
<th>Run:</th>
</tr>
</thead>
<tbody>
<tr>
<td>View the current configuration</td>
<td><code># show running-config</code></td>
</tr>
<tr>
<td>View current ports status</td>
<td><code># show interface</code></td>
</tr>
<tr>
<td>View interface statistics</td>
<td><code># show interface &lt;interface ID&gt; statistics [extended]</code></td>
</tr>
<tr>
<td>View SSM logs</td>
<td><code>#show log buffer</code></td>
</tr>
<tr>
<td>Modify the group of SGMs amongst which the load is distributed</td>
<td><code># configure terminal</code>&lt;br&gt;<code>(config)# load-balance mtx-bucket [SGM ID, SGM ID,]</code>&lt;br&gt;<code>(config)# load-balance apply</code>&lt;br&gt;<strong>Note:</strong> the command will not work if you have an odd number of SGMs in the group. For example, do not run:&lt;br&gt;<code>#load-balance mtx-bucket 1,2,3</code>&lt;br&gt;Run:&lt;br&gt;<code>#load-balance mtx-bucket 1,2,3,1,2,3</code></td>
</tr>
</tbody>
</table>
The password for the SSM is `admin`. Once connected to the SSM CLI you can do the following:

1. **View the current configuration**
   # show running-config [feature name]  
   Since the entire configuration is very long it is recommended to specify the feature which you are interested in its configuration.  
   For example `show running-config load-balance` to see the Load Balance configuration.  
   You can press `tab` to see a complete list of the features.

2. **View current ports status**
   # show port

3. **View detailed port information (speed, administrative state, link state, etc.)**
   # show port <port ID>

4. **View interface statistics**
   # show port <Port ID> statistics

```
T-HUB4#show port 1/3/1 statistics

Port Statistics

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast Packets</td>
<td>5003</td>
</tr>
<tr>
<td>Multicast Packets</td>
<td>568409</td>
</tr>
<tr>
<td>Broadcast Packets</td>
<td>122151</td>
</tr>
<tr>
<td>Flow Control</td>
<td>0</td>
</tr>
<tr>
<td>Discards</td>
<td>16</td>
</tr>
<tr>
<td>Errors</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>695563</td>
</tr>
</tbody>
</table>

Ethernet Statistics in Packets

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>RX CRC Errors</td>
<td>0</td>
</tr>
<tr>
<td>RX Undersize</td>
<td>0</td>
</tr>
<tr>
<td>Fragments</td>
<td>0</td>
</tr>
<tr>
<td>Oversize</td>
<td>0</td>
</tr>
<tr>
<td>Jabbers</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input and Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Octets</td>
</tr>
<tr>
<td>Packets</td>
</tr>
<tr>
<td>Packets of 64 Octets</td>
</tr>
<tr>
<td>Packets of 65 to 127 Octets</td>
</tr>
<tr>
<td>Packets of 128 to 255 Octets</td>
</tr>
<tr>
<td>Packets of 256 to 511 Octets</td>
</tr>
<tr>
<td>Packets of 512 to 1023 Octets</td>
</tr>
<tr>
<td>Packets of 1024 to 1518 Octets</td>
</tr>
<tr>
<td>Packets of 1519 or more Octets</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

Rates in Bytes per Second

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate for last 10 sec</td>
<td>1477</td>
</tr>
<tr>
<td>Rate for last 60 sec</td>
<td>1435</td>
</tr>
</tbody>
</table>
```
Pay special attention to “Discards” and “Errors” fields which might indicate a problem if they are constantly increasing.

5. **View SSM logs**
   
   # unhide private (default password is “private”)  
   # show private shell  
   # tail /var/log/messages

6. **Modify load distribution SGM group**
   
   # configure terminal  
   (config)# load-balance mtx-bucket 1 buckets [<SGM ID><SGM ID>:<SGM ID><SGM ID>...]  
   (config)# commit  
   (config)# exit  
   # load-balance apply

   **Note**
   
   You need to provide a full list of the SGMs as the SGM list parameter to the `load-balance mtx-bucket` command. Otherwise, traffic might be dropped on the SSM.

7. **Switch between Ports modes for 40G ports (4X10G or 1X40G):**
   
   # unhide private (default password is “private”)  
   # show private shell  
   For switching to 1X40G mode:  
   # /batm/binux/bin/ub_util -s ahub4_40G yes  
   For switching to 4X10G mode:  
   # /batm/binux/bin/ub_util -s ahub4_40G  
   # exit  
   # config terminal  
   (config)# system reload

   **Note**
   
   This procedure requires to reload the SSM. It is recommended to do it one SSM at a time.

8. **View the current version information**
   
   # show version

9. **Logout from current session**
   
   # logout

10. **Changing SSM160 admin password**

    a. Login via SSH/Serial console to an SGM which resides in the same chassis you wish to change SSMs password
    b. From Expert shell Login to either of the SSMs in the chassis using:
       
       ssh admin@ssm<SSM ID>
    c. Enter `admin` password when prompted.
    d. In SSMs shell run the following commands:
       
       i. #conf t  
       ii. #system security user admin  
       iii. #password  
       iv. Enter new password  
       v. #commit  
       vi. #end  
       vii. #logout

   **Notes**
   
   - This procedure should be done separately on each SSM in the system.  
   - This procedure does not cause any traffic interruption
Example

```bash
[Expert@61000-gw]# ssh ssm2
admin@ssm2's password: BATM4-HUB4
T-HUB4#conf t
Entering configuration mode terminal
T-HUB4(config)#system security user admin
T-HUB4(config-user-admin)#password
(<MD5 digest string>): *****
T-HUB4(config-user-admin)#commit
Commit complete.
T-HUB4(config-user-admin)#end
T-HUB4#log
Connection to ssm2 closed.
```

Each port ID on the SGM maps to a port on the SSM. Below table maps SSM port ID to SGM port ID. Note that this table relates to SSM1. For SSM2 replace eth1-X with eth2-X:

<table>
<thead>
<tr>
<th>SGM</th>
<th>SSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth1-01</td>
<td>1/3/1</td>
</tr>
<tr>
<td>eth1-02</td>
<td>1/3/2</td>
</tr>
<tr>
<td>eth1-03</td>
<td>1/3/3</td>
</tr>
<tr>
<td>eth1-04</td>
<td>1/3/4</td>
</tr>
<tr>
<td>eth1-05</td>
<td>1/3/5</td>
</tr>
<tr>
<td>eth1-06</td>
<td>1/3/6</td>
</tr>
<tr>
<td>eth1-07</td>
<td>1/3/7</td>
</tr>
<tr>
<td>eth1-Sync</td>
<td>1/3/8</td>
</tr>
<tr>
<td>eth1-09</td>
<td>1/1/1</td>
</tr>
<tr>
<td>eth1-10</td>
<td>1/1/2</td>
</tr>
<tr>
<td>eth1-11</td>
<td>1/1/3</td>
</tr>
<tr>
<td>eth1-12</td>
<td>1/1/4</td>
</tr>
<tr>
<td>eth1-13</td>
<td>1/1/5</td>
</tr>
<tr>
<td>eth1-14</td>
<td>1/1/6</td>
</tr>
<tr>
<td>eth1-15</td>
<td>1/1/7</td>
</tr>
<tr>
<td>eth1-16</td>
<td>1/1/8</td>
</tr>
<tr>
<td>eth1-Mgmt1</td>
<td>1/5/1</td>
</tr>
<tr>
<td>eth1-Mgmt2</td>
<td>1/5/2</td>
</tr>
<tr>
<td>eth1-Mgmt3</td>
<td>1/5/3</td>
</tr>
<tr>
<td>eth1-Mgmt4</td>
<td>1/5/4</td>
</tr>
</tbody>
</table>

Verification
To verify that you have connectivity to the SSMs from the SGMs ping all the SSM modules IPs. You can also verify that SNMP connectivity is available by running from SGM shell:

```
asg_chassis_ctrl get_ssm_firmware all
```

61000 Security System LEDs

Security Gateway Module LEDs

<table>
<thead>
<tr>
<th>Item</th>
<th>LED</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
</table>

---
### Security Switch Module LEDs

<table>
<thead>
<tr>
<th>Item</th>
<th>LED</th>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Out of service</td>
<td>Red</td>
<td>SGM out of service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off (Normal)</td>
<td>SGM hardware is normal</td>
</tr>
<tr>
<td>2</td>
<td>Health</td>
<td>Green (Normal)</td>
<td>SGM core operating system is active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Green blinking</td>
<td>SGM core operating system is partially active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off</td>
<td>SGM operating system is in standby mode</td>
</tr>
<tr>
<td>3</td>
<td>Hot-swap</td>
<td>Blue</td>
<td>SGM can be safely removed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blue blinking</td>
<td>SGM is going to standby mode. Do not remove</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off (Normal)</td>
<td>SGM is active. Do not remove</td>
</tr>
<tr>
<td>4</td>
<td>Link</td>
<td>Yellow</td>
<td>Link enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yellow blinking</td>
<td>Link is active</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off</td>
<td>Link is disabled</td>
</tr>
<tr>
<td>5</td>
<td>Speed - data ports</td>
<td>Yellow</td>
<td>10 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Green</td>
<td>1 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off</td>
<td>100 Mbps</td>
</tr>
<tr>
<td></td>
<td>Speed - management port</td>
<td>Yellow</td>
<td>1 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Green</td>
<td>100 Mbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off</td>
<td>10 Mbps</td>
</tr>
<tr>
<td>6</td>
<td>L</td>
<td>LEDs 2 and 4 - Green</td>
<td>SGM is being configured. (Using First Time Wizard or adding a new SGM into a chassis)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All LEDs - Off</td>
<td>SGM is configured and ready</td>
</tr>
</tbody>
</table>
### Software Blades Support

61000 Security System supports the following software blades:

- Firewall
- IPSec VPN
- IPS
- Identity Awareness
- Anti-Virus (non proactive mode, HTTP and SMTP only)
- Application Control
- URL Filtering (legacy mode)

The capabilities of these software blades are similar to those that were provided by Check Point R75 release.

### Software Blades Updates

61000 Security System periodically updates Anti-Virus and URL Filtering databases, same as other Check Point products.

In order to manually update Anti-Virus and URL Filtering databases, use `g_avsu_update` command. This command is available from Expert shell only.

Upon execution, the command will update the database of the relevant SGMs.
Syntax:

```bash
g_avsu_update -b <blade string> <urlf/av/all>
```

Note:

Update configuration (proxy, username, etc.) should be set in SmartDashboard before issuing this command. Policy should be installed afterwards. Manual updates of Anti-Virus and URL Filtering from SmartDashboard are not supported.

## IPS Bypass under Load

Bypass under Load allows the administrator to define a gateway resource load level at which IPS inspection will temporarily be suspended until the gateway's resources return to acceptable levels. IPS inspection can make a difference in connectivity and performance. Usually, the time it takes to inspect packets is not noticeable; however, under heavy loads it may be a critical issue.

You have the option to temporarily stop IPS inspection on a gateway if it comes under heavy load.

For more about this feature, see the document R75 IPS Administration Guide at [http://downloads.checkpoint.com/dc/download.htm?ID=11663](http://downloads.checkpoint.com/dc/download.htm?ID=11663)

## IPS Cluster Failover Management

**Description**

You can configure how IPS is managed during a cluster failover (when one member of a cluster takes over for another member to provide High Availability).

**To configure failover behavior for a cluster:**

Run from expert shell: `asg_ips_failover_behavior [connectivity|security]`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connectivity</td>
<td>change the behavior upon cluster failover to prefer connectivity: Close connections for which IPS inspection cannot be guaranteed</td>
</tr>
<tr>
<td>security</td>
<td>change the behavior upon cluster failover to prefer security: Keep connections alive even if IPS inspections cannot be guaranteed</td>
</tr>
</tbody>
</table>
Improved handling of Software Blades traffic *(applies to R75.052)*

**Description**

In 61000 R75.052, Several optimizations for software blades traffic handling were integrated.

**Before implementing these optimizations in production network, please consult with Check Point Support.**

Optimizations consist of:

- Enabling SMT (HyperThreading), which utilizes each physical core as 2 logical cores. This stage requires BIOS update.
- Increasing network buffers and firewall queues, for better handling of intensive Software Blades traffic. For this stage, run `asg_swb_perf` command.

**To enable optimizations**

1. Contact Check Point Support in order to get the BIOS firmware, and copy it to one of the SGMs under `/var/log/`
2. Copy the BIOS firmware to all blades:

   `> asg_cp2blades /var/log/5322H054_cp_a20_hpm1bios.img`

3. Install the new BIOS firmware:

   `# g_alls smt_config enable /var/log/5322H054_cp_a20_hpm1bios.img`

   Note: This step may take a while (approx. 10-15 minutes)

4. Reboot all blades:

   `> reboot –b all`

5. Repeat step 3, in order to upgrade the backup BIOS firmware:

   `# g_alls smt_config enable /home/admin/5322H054_cp_a20_hpm1bios.img`

6. Reconfigure core distribution between CoreXL and SecureXL as desired:

   `# g_cpconfig corexl instances <num>`

   Note: The amount of cores assigned to SecureXL will be the total amount of cores (both physical and logical) minus the amount assigned to CoreXL.

7. Reboot all blades:

   `> reboot –b all`

8. Enable software blades optimizations:

   `# asg_swb_perf enable`

9. Reboot all blades:
> reboot –b all

**To disable optimizations**

1. Verify that core distribution between CoreXL and SecureXL is such, that the amount of SecureXL cores is smaller than the amount of physical cores on the machine (i.e. the amount of cores there will be after disabling SMT).

   E.g. for 12 physical cores (24 cores with SMT enabled), it is alright to have 4 or 8 SecureXL cores, but not 16.

   If current configuration is not as described, reconfigure and reboot all blades:

   ```
   # g_cpconfig corexl instances 20
   > reboot –b all
   ```

2. Revert to original BIOS firmware (taken automatically from $FWDIR/conf/hw_firmware):

   ```
   # g_alls -a smt_config disable
   ```

   Note: This step may take a while (approx. 10-15 minutes).

3. Reboot all blades:

   ```
   > reboot –b all
   ```

4. Repeat step 2:

   ```
   # g_alls -a smt_config disable
   ```

5. Reconfigure core distribution between CoreXL and SecureXL as desired:

   ```
   # g_cpconfig -a corexl instances <num>
   ```

6. Reboot all blades:

   ```
   > reboot –b all
   ```

7. Disable software blades optimizations:

   ```
   # asg_swb_perf disable
   ```

8. Reboot all blades:
To verify current state

To ascertain whether the optimizations are enabled, run the following commands:

```
# g_allc smt_config status
# asg_swbf_perf status
```

Note: activation must be done on both chassis simultaneously. Activation/deactivation of this functionality on each chassis separately is not supported.

Software Upgrade and Hardware Replacement

61000 Component Versions

The supported version for all 61000 Security System components:

<table>
<thead>
<tr>
<th>Component</th>
<th>R75.035</th>
<th>R75.050</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSM60 Base switch configuration</td>
<td>7.5.20</td>
<td>7.5.20</td>
</tr>
<tr>
<td>SSM60 Base switch firmware</td>
<td>7.5.20</td>
<td>7.5.20</td>
</tr>
<tr>
<td>SSM60 Fabric switch configuration</td>
<td>7.5.20</td>
<td>7.5.20</td>
</tr>
<tr>
<td>SSM60 Fabric switch firmware</td>
<td>7.5.20</td>
<td>7.5.20</td>
</tr>
<tr>
<td>SSM160 firmware</td>
<td>2.4.B6</td>
<td>2.4.B11</td>
</tr>
<tr>
<td>CMM firmware</td>
<td>2.74</td>
<td>2.74</td>
</tr>
<tr>
<td>SGMs BIOS image</td>
<td>1.3</td>
<td>1.3</td>
</tr>
<tr>
<td>IPMC bootloader image</td>
<td>1.42</td>
<td>1.42</td>
</tr>
<tr>
<td>IPMC firmware image</td>
<td>1.52</td>
<td>1.52</td>
</tr>
<tr>
<td>FPGA image</td>
<td>2.4</td>
<td>2.4</td>
</tr>
<tr>
<td>FPGA recovery image</td>
<td>2.4</td>
<td>2.4</td>
</tr>
<tr>
<td>SSD firmware</td>
<td>4PC10302</td>
<td>4PC10302</td>
</tr>
<tr>
<td>Check Point software</td>
<td>Take 55</td>
<td>Take 65</td>
</tr>
</tbody>
</table>
Upgrading Chassis Software

Description:
You can upgrade the 61000 Security System in Dual Chassis setup. This procedure assures network connectivity at all times during the upgrade by maintaining at least one Active Chassis that handles traffic. Old connections, that is, connections that were opened on the chassis with the old version, will survive the upgrade only when upgrading between minor versions.

Notes
1. The procedure is supported for all versions between R75.035 and R75.05X. (See the official version names in [http://supportcontent.checkpoint.com/solutions?id=sk77880](http://supportcontent.checkpoint.com/solutions?id=sk77880)).
2. This upgrade procedure converts the database to the new version. The conversion log file is in the directory /var/log/upgrade/
3. When exporting or importing a snapshot, you must export from or import to the /var/log directory.

Upgrade procedure:
1. Make sure that all SGMs are in the UP state.
   • To see the state of the SGMs, run `asg stat`.
   • To configure SGMs as Up or Down use `asg sgm_admin`.
2. Connect to the SMO SGM using SSH or the console. To open a console connection:
   a) Connect one end of a serial cable to the console port on the SGM front panel.
   b) Connect the other end of the serial cable to a computer,
   c) Open a console window. Use the default serial connection parameters: 9600, 8, N, 1.
3. Copy new_version snapshot file (*.tar) to SMO SGM, and from SMO to all SGMs. From clish run:
   `>asg_cp2blades <snapshot_file_name> /var/log/<snapshot_file_name>`
4. Verify MD5 sum on the copied file. Run:
   `g_md5sum <new_version_snapshot_file_path>`.
5. Import snapshot on all SGMs via gclish:
   `set snapshot import <new_version_snapshot_file_name (without .tar)> path /var/log/`
6. Put the standby chassis in an administrative DOWN state. From gclish run:
   `asg chassis_admin -c <stand_by_chassis_id> down`
7. Revert to the new snapshot on the down admin chassis via the shell:
   `g_snapshot <blade_string> revert <snapshot_name>`
   Example of reverting to snapshot: my_snapshot, for chassis 2:
   `g_snapshot -b chassis2 revert my_snapshot`
8. Wait until the chassis is UP and is running the new version. To verify this, in gclish run:
   `ver`
9. When upgrading to R75.052 or higher and initial configuration includes bond interface, run:
   `g_fw ctl set int fwha_ch_ha_port_monitor_legacy 1` from Expert shell.
   Note: Non upgraded SGMs will generate error after processing this command.
10. Put the upgraded chassis in an administrative UP state. From gclish run:
    `asg chassis_admin -c <chassis_id> up`
11. Validate that the backup chassis is activated as backup. On one of the connections run
    `asg search`
12. Put the active chassis in an administrative DOWN state. In gclish run:
    `asg chassis_admin -c <chassis_id> down`
13. Revert to the new snapshot on the DOWN admin chassis. In the the shell run:
    `g_snapshot <blade_string> revert <snapshot_name>`
14. When upgrading to R75.052 or higher and initial configuration includes bond interface, run:
    `g_fw ctl set int fwha_ch_ha_port_monitor_legacy 0` from Expert shell.
    Bond interfaces will now be considered as separate entities in `asg stat -v` output.
15. Wait until the chassis is UP and is running the new version. To verify this, in gclish run `ver`.

16. When upgrading from R75.035 to R75.05X: Upgrade SSM160 firmware to version 2.4.B11. See Upgrading SSM Firmware.

Verifying that the chassis software upgrade was successful:
1. Verify the current snapshot on all SGMs. In gclish run:
   ```
   show snapshots
   ```
2. Run `asg diag` to validate system integrity.

### Upgrading CMM Firmware

**Description:**
You can upgrade the CMM firmware. You can also move to a lower firmware version.

Before upgrading the CMM firmware, or moving to a lower software version, contact Check Point Support to get the firmware files. See Error! Reference source not found..

In a Dual Chassis configuration, run this procedure for each chassis, one CMM at a time. The chassis must be in down state when updating the CMM firmware.

**Upgrade procedure:**
Do this procedure on the standby chassis:

1. To put the chassis in standby state, run this command from gclish:
   ```
   asg chassis_admin -c <chassis id> down
   ```
2. Remove all CMMs from the chassis.
3. Insert the replacement CMM into the chassis.
4. Open a console connection to one of the SGMs on the chassis:
   a) Connect one end of a serial cable to the console port on the SGM front panel.
   b) Connect the other end of the serial cable to a computer,
   c) Open a console window. Use the default serial connection parameters: 9600, 8, N, 1.
5. Copy the firmware files to the SGM
6. Copy firmware files from the SGM to the /tmp/ directory on the CMM. From the expert shell, run:
   ```
   # scp sentry.shmm500.* admin@198.51.100.33:/tmp/
   ```
7. Open a console connection to the serial port on the CMM front panel.
8. On the CMM, run these commands:
   ```
   login: admin
   Password: admin
   # cd /tmp
   # setenv rc2 /etc/rc.asis
   # clia terminate
   # rupgrade_tool -s -v --r=sentry.shmm500.rfs --k=sentry.shmm500.kernel --u=sentry.shmm500.u-boot --hook=erase
   ```
9. To start the installation, run
   ```
   # install.sh
   ```
10. After the boot process has finished, select the appropriate chassis type.
   a) select option 2 if the chassis type is AC Telkoor PSU or a DC chassis
   b) select option 3 if the chassis type is AC Lambda.
Select one of following options.
1: Press 1 for 13U chassis (Telkoor PSU).
2: Press 2 for 14U chassis (Telkoor PSU).
3: Press 3 for 14U chassis (Lambda PSU).
Q: Press Q for to skipp.

(Press 1, 2, 3 or Q)?

11. If the chassis ID needs to be ‘2’ then edit the chassis ID configuration:

   login: admin
   Password: admin
   # sed -i 's/CHASSID="1"/CHASSID="2"/g' /etc/shmm.cfg
   # reboot

Upgrading SSM Firmware

Upgrading SSM160 firmware

Description:
You can upgrade the SSM 160 firmware in a dual chassis setup of the 61000 Security System. You can also move to a lower firmware version.

Before upgrading the SSM firmware, or moving to a lower software version, contact Check Point Support to get the firmware files.

Network connectivity is assured at all times during the procedure by having at least one Active Chassis.

Note – There is no need to upgrade firmware on the SSM60.

Upgrade procedure:
1. Before the upgrade check the configured distribution mode. This information is used in the validation phase. From gclish, run
   >asg dxl dist_mode verify
2. Copy the firmware to the SMO.
3. Make sure that the relevant chassis is in standby mode. From gclish, run
   >asg chassis_admin -c <chassis id> down
4. Copy the firmware to SGM on the relevant chassis. On the SMO, run
   >asg_cp2blades -b <blade_list> <file>
5. Connect to SGM on the standby chassis. From Expert shell run  
   # blade [chassis_]blade
   # blade [chassis_]blade
6. Copy the firmware image file to all SSMs on the Standby chassis. Run for each SSM:
   scp -P 2024 2.4.B6.T-HUB4.tar.bz2 root@<SSM_IP>:/batm/current_version/

   SSM_IP to get the SSM IP run from gclish:
   >show chassis id <1|2|all> module SSM<1|2>"

   6.1. In case of the read-only file system error do the following:
   Example of read-only error:
   # scp -P 2024 2.4.B11.T-HUB4.tar.bz2 root@ssm2:/batm/current_version/
   root@ssm2's password:
   
   a) Connect to the SSM via ssh. run from expert shell: ssh ssm<1/2>, password is “admin”
   b) From default shell, run “unhide private”. password is “private”
   c) Run the following commands:
d)  # show private shell

e)  # mount -rw -o remount /batm/

f)  # exit

g)  # logout

h)  Copy firmware file again (follow Step 3).

7. Activate the new image on the SSM. Repeat this step for all SSMs on the current chassis:

   a)  Connect to the SSM via ssh. Run from expert shell:`ssh ssm<1/2>`, password is “admin”

   b)  Run
        # file ls os-image
        and copy to clipboard the name of the new image file

   c)  Run
        # file activate-os-image <new image file name>

   d)  Move to config shell by running
        # config terminal

   e)  Reload the SSM with the new image. Run
        # system reload manufacturing-defaults

Example:

```
T-HUB4# file ls os-image
  1 May 01:05 8.5M 2.3.R1.T-HUB4.tar.bz2
  * 1 Jan 2010  8.6M 2.2.R1.T-HUB4.tar.bz2
  1 Jan 01:56 8.6M 2.2.R2.T-HUB4.tar.bz2

T-HUB4# file activate-os-image 2.3.R1.T-HUB4.tar.bz2
Image file 2.3.R1.T-HUB4.tar.bz2 is tested for validity, please wait... OK
Activating image 2.3.R1.T-HUB4.tar.bz2..
```

```T-HUB4# config terminal
Entering configuration mode terminal
T-HUB4(config)# system reload manufacturing-defaults
Are you sure that you want to delete existing configuration and
reload manufacturing default configuration (yes/no)? yes```

8. Validate SSM firmware version. From the SMO gclish shell, run

```
# asg_version
```

9. Perform Chassis admin up to the current chassis. From gclish, run

```
> asg chassis_admin -c <chassis_id> up
```

10. Repeat the Firmware upgrade procedure on the SSMs of the other chassis.

Validation

1. To verify the upgrade, run `asg_version` or do the following:

   ```
   > asg_chassis_ctrl get_ssm_firmware all
   Firmware version of SSM1 is 2.4.B11
   Firmware version of SSM2 is 2.4.B11
   ```

2. Validate that the correct distribution mode is set as before the upgrade. From gclish, run:

   ```
   > asg dxl dist_mode verify
   ```

**Upgrading SSM60 to SSM160**

**Procedure Summary:**

1. Disconnect Standby chassis from network and upgrade it
2. Disconnect Active Chassis from the network and reconnect the upgraded Chassis to the network
3. Upgrade the 2nd chassis and reconnect to the network

**Detailed steps:**

1. Pre Upgrade
a. Connect to the 61000 Security System, get and save the following chassis configuration for later verifications:
   - asg if
   - asg stat -v
   - asg diag
   - asg_version

b. Create a backup of the configuration by running the following command from shell:
   
   ```
   g_all backup_system backup ssm60
   ```

   **Important note:**
   Make sure the correct firmware version on the SSM160s is installed:
   - 61000 R75.035 is only compatible with SSM160 firmware 2.4.B6
   - 61000 R75.050 is only compatible with SSM160 firmware 2.4.B11

2. Disconnect Standby chassis (B) entirely from the Network and upgrade SSMs
   a. Disconnect the Standby chassis (B) entirely from the network (Management Interfaces, Traffic Interfaces and Sync Interface)
   b. Shutdown (physically) all SGMs in the disconnected chassis
   c. Replace both SSM60 with SSM160 in the disconnected chassis. Wait for both SSM160 to boot up (might take few minutes).
   
   **Verify** SSM160s have the correct firmware installed (see section 1), upgrade if needed.
   d. Power up all SGMs in the disconnected chassis
   e. Wait for all SGMs to go UP. Check this by running `asg_monitor` on one of the SGMs (via console).
   
   **Note:** SGMs may perform more than one reboot while booting up, therefore this step might take a while to finish.
   f. Re-apply previous distribution settings : run `asg dxl dist_mode set -f`
   g. Reconfigure management interfaces if needed. See SSM Upgrade Appendix – Reconfiguring Management Interfaces.
   h. Run verification tests:
      - Verify SSM firmware – run from gclish `asg_version`.
      - Validate Chassis configuration by comparing the following with configuration output from section 1
        1. `asg if` – Validate same IP addresses exist on the Interfaces.
           Pay attention to management interface changes according to the above reconfiguration (note that the new management interface will not appear until policy with the new topology is installed).
        2. `asg diag`
        3. `asg stat -v` – Verify all local chassis SGMs are up and running and chassis state is Active.
      - To verify successful creation of interfaces on all SGMs, run `ethtool eth1-[1-16]` and `ethtool eth2-[1-16]` from gclish and verify that all SGMs identify them.
      - Run `asg hw_monitor` to verify CIN health
i. Configure the GARPs refresh interval to 10 seconds by “g_fw ctl set int fwha_gratuitous_arp_timeout 100”

3. Disconnect the Active chassis (A) and reconnect the new upgraded Active chassis (B)
   a. The existing cables should be disconnected from the Active chassis (A).
   b. The existing cables should be reconnected to the new upgraded Active chassis (B).
      Take into account the change in the management interface which was configured in the previous section.
   c. In case of bond interfaces, it is recommended to run asg_chassis_ctrl clear_mac_learning.
   d. Verify that traffic is processed by the new Active upgraded chassis (B).
      At this point, rollback to the previous Active chassis can be performed if needed.

4. Upgrade SSMs on the previous Active (disconnected) chassis (A)
   a. Repeat steps 2.b – 2.i on the disconnected chassis.

5. Re-connect the previous Active chassis (A) to the network and to the other chassis
   a. Re-Connect the Sync network Cables between the chassis
   b. Re-Connect the other network Cables (Management Interfaces, Traffic Interfaces)
      Take into account the change in the management interface which was configured in the previous section.
   c. Verify both chassis communicate
   d. Run from gclish ‘asg stat –v’ and verify you have Active & Standby Chassis and all SGMs are UP.
   e. Get topology and install policy (make sure the new management port configuration takes effect).

6. Configure GARPs on chassis (B) back to their default
   c. Configure the GARPs refresh interval in the system back to its default (60 seconds) by running from shell “g_fw ctl set int fwha_gratuitous_arp_timeout 600”

SSM Upgrade Appendix – Reconfiguring Management Interfaces

Background:
On SSM60, ethX-Mgmt[1|2] are 1G and ethX-Mgmt[3|4] are 10G
On SSM160, ethX-Mgmt[1|2] are 10G and ethX-Mgmt[3|4] are 1G

When changing SSM60 to SSM160, ethX-Mgmt1 IP address automatically moves to ethX-Mgmt4.
In case the management interface that was being used in SSM60 is not ethX-Mgmt1, it will have to be reconfigured manually to a port with the same speed on SSM160.

For example, if the management port on SSM60 is eth1-Mgmt3 (10G port), it will have to be reconfigured, for instance, to port eth1-Mgmt1 on SSM160 (also 10G port), by running the following commands in gclish:

delete interface eth1-Mgmt3 ipv4-address
set interface eth1-Mgmt1 ipv4-address <Mgmt IP> mask-length <mask length>

Replacing the CMM

This procedure explains how to install a replacement CMM that you received through the Return Merchandise Authorization (RMA) process.
This procedure assumes that you have a Standby chassis.

The procedure for replacing the CMM has the following steps:

1. **Insert RMA CMM to Standby chassis**
2. **CMM firmware validation.**
   - Validate that the same firmware version is installed on all CMM units in your system.
3. **Chassis Type validation.**
   - Validate that the same chassis type is configured on all CMM units in your system.
4. **Chassis ID validation.**
   - Validate that the chassis ID configured on the replacement CMM matches to the chassis it is plugged into.

**Note:** The CMM configuration information can be found on the label on the outside of the CMM packaging box.

![Warning]

**Note** - The replacement CMM must be on the standby chassis

1. **Insert RMA CMM to Standby chassis.**
2. **Validating CMM Firmware Version**

   **Important:** All CMMs in environment must have the same firmware version.

   A. Find out the firmware version of all CMMs in your system:
   Connect to the SMO gclish shell and run:

   ```
   asg_version
   ```
B. Make sure that the firmware version is the same as the firmware version on the label on the outside of the CMM packaging box.
C. If the firmware versions are not the same, follow sk92628 (61000 Security System Upgrading CMM firmware), for each CMM that needs to be upgraded.

3. Validating Chassis Type

Make sure that the Chassis Type on the label on the outside of the CMM packaging box is the same as your chassis.

The supported Chassis Types are:

4. DC chassis
5. AC Telkoor: The AC Chassis has two rows of three Telkoor power supplies in each row.
6. AC Lambda: The AC Chassis has one row of five Lambda power supplies

If your Chassis Type is not the same as the Chassis Type on the label on the outside of the CMM packaging box, do this procedure on the Standby chassis:

7. To put the chassis in Standby state, run this command from gclish:
   
   ```
   asg chassis_admin -c <chassis_id> down
   ```

8. Remove all CMMs from the chassis.
9. Insert the replacement CMM into the chassis.
10. Open a console connection to the CMM:
    
    a. Connect one end of a serial cable to the serial port on the CMM front panel.
    b. Connect the other end of the serial cable to a computer.
Replacing the CMM

Software Upgrade and Hardware Replacement

Page 221

For public distribution

11. To start the installation, run:
   
   # install.sh

12. After the boot process has finished, select the appropriate chassis type:
   
   a. Menu may vary according to CMM firmware. The following menu will be displayed when firmware 2.74 is installed.
   
   b. Select option 2 if the chassis type is AC Telkoor PSU or a DC chassis.
   
   c. Select option 3 if the chassis type is AC Lambda.

<table>
<thead>
<tr>
<th>Select one of following options.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Press 1 for 13U chassis (Telkoor PSU).</td>
</tr>
<tr>
<td>2: Press 2 for 14U chassis (Telkoor PSU).</td>
</tr>
<tr>
<td>3: Press 3 for 14U chassis (Lambda PSU).</td>
</tr>
<tr>
<td>Q: Press Q for to skip.</td>
</tr>
</tbody>
</table>

(Press 1, 2, 3 or Q)?

13. Insert the second CMM (which was removed in step 3-B)

4. Validating Chassis ID

Make sure that the chassis ID on the label on the outside of the CMM packaging box is the same as the label on the chassis.

If Standby chassis ID is different from RMA CMM chassis ID, then modify RMA CMM chassis ID as described in "Chassis ID Configuration".
RMA/Add New SGM Procedure

This topic explains how to update the OS version on an RMA/New SGM to suit the customer running OS version.

There are two methods to update the OS versions:
1. Create a snapshot image from one of the standby SGMs and revert the RMA/New SGM to this snapshot.
2. Install from a CD. Please contact Check Point support.

RMA/New SGM procedure using snapshot:

This procedure ensures that the same installation the user already has on its current environment, (including HF and RND patch) will be installed on the RMA/New SGM.

Procedure steps:
1. Create image snapshot for the existing configuration and export it.
2. Import the snapshot to the RMA/New SGM.
3. Add the RMA/New SGM to the security group.
4. Verification.

Create snapshot for the existing configuration and export it:

Note: In Dual Chassis configuration, it is recommended to create a snapshot on the Standby chassis.

1. Access the 61000 Security System, and switch to SGM on the standby chassis. [run from Expert shell blade <standby_chassis_id>_<blade_id>]
2. Set global mode off so that the new snapshot image will be created only on this SGM: [set global-mode off]
3. Create a new image snapshot: [add snapshot <snapshot_name> desc <snapshot description>]
4. Use the command 'show snapshots' to monitor creation progress: [show snapshots]
5. Insert a removable disk to the USB port of the SGM and mount it to /mnt/usb [see Mounting USB]
6. When creation process is done export the snapshot to a tar file under /mnt/usb directory: [set snapshot export <snapshot name without the .tar> path /mnt/usb]
7. Use the command 'show snapshots' to monitor exporting progress: [show snapshots]
8. Unmount /usb/mnt [umount /mnt/usb]
9. Remove the USB stick from the SGM.
Import the snapshot to the RMA/New SGM

1. Choose the standby chassis and insert the RMA/New SGM to a slot that does not participant in the security group
   a. If all the slots are taken, reconfigure the security group and remove one of the SGM from it:
      
      ```shell
      [asg security_group ]
      ```

2. Insert the removable disk to the USB port of the RMA and mount it to /mnt/usb [see Mounting USB]

3. Connect to the SGM by Console

4. Import the snapshot file: [ set snapshot import <filename without the .tar> path /mnt/usb/]

5. Use the command ‘show snapshots’ to monitor importing progress: [show snapshots]

6. Un mount the /mnt/usb and remove the removable disk [umount /mnt/usb]

7. Revert the RMA to the snapshot image: [ set snapshot revert <snapshotname> ]

8. The revert process take a while and includes reboot. When the reboot starts you can operate the next step.
Add the RMA/New SGM to the security group

1. Access the 61000 Security System and update the security group to include the RMA/New SGM: 
   [asg security_group ]

Verification:

1. Verify that the RMA/New SGM is up and enforcing the latest policy [asg monitor]
2. Verify that all the SGMs have the same OS version.: [asg_version ]

RMA/new SGM procedure using CD installation:

1. Choose the standby chassis and insert the RMA to an SGM that does not participant in the security group
   a. If all the SGMs participant in the security group, reconfigure the group and remove one of the SGM: [asg security_group ]
2. Connect with console to the RMA/New SGM
3. Remove the boot sector [eraseboot]
4. Insert the CD
5. Reboot the SGM [reboot]
6. While RMA/New SGM is in boot process, access the 61000 Security System and update the security group to include the RMA/New SGM [asg security_group ]

Mounting USB

1. Insert the removable disk to the USB port
2. Find the file system name for the USB in messages log file.[ in shell run tail /var/log/messages]
3. Create ‘usb’ directory under /mnt if not exist [mkdir /mnt/usb]
4. Mount the USB file system to your usb directory [mount /dev/sdb1 /mnt/usb]
5. Note: if you are copying files to the USB, you must do un mount before pulling out the removable disk. \textit{[umount /mnt/usb]}

6. Note: before rebooting the system don't forget to run \textit{`umount /mnt/usb`} and pull out the removable disk.
Licensing and Registration

61000 Security Systems have an initial 15-day evaluation license. After the evaluation license expires, you must license and register the system.

Each chassis is licensed separately. If you have dual chassis system, you must install two licenses.

The license key (CK) is the Chassis Serial Number.

Chassis has two Serial Numbers:

1. With format: XXXXXXX/XXX
2. With format: TPXXXXXXX
   This S/N is only available on chassis sticker, next to the Check Point logo

Note Due to Logistical issues, some of Chassis are registered in User Center using the first S/N and others are registered using the second S/N

To Retrieve Chassis Serial Number with format XXXXXXX/XXX:

A. If a policy is installed on the SGM:
   1. Connect to one of the SGMs on the chassis.
   2. Run asg_serial_info

   The output shows the Chassis Serial Number.

B. if no policy is installed on the SGM:
   1. Connect to one of the SGMs on the chassis.
   2. Get the IP address of the CMM by running (from gclish):
      show chassis id all module CMM1 ip
   3. Using the IP address, open an SSH connection to the CMM.
   4. On the CMM, run: clia fruinfo 20 254

   The output shows the Chassis Serial Number.

To license and register the 61000 Security System

1. Open the User Center Registration page (http://register.checkpoint.com/cpapp)
2. Search for the Chassis Serial Number.
3. Generate a license based on the IP address of the SSM interface connected to your Security Management Server

Note 61000 Security System has single Management IP address, thus in dual chassis environments, the Active and Standby Chassis should be bound to the same IP address in the license. Generate two licenses and enter the same IP address in each license.

4. Install the license on the system.
   a. If you use the cplic command, run it from GCLISH so that it applies to all SGMs. Run If you have a dual Chassis environment, run cplic twice. Once per Chassis.
   b. If using SmartUpdate, install the Policy.
Troubleshooting

Collecting System Information (asg_info)

Description:

Use this command to collect system information. The information consists of files and commands output. Major categories of collected information are:

- Log files
- Configuration files
- System status
- Indication for possible errors

The information is collected from all the SGMs and placed into a compressed folder named asg_report.<timestamp> located under /tmp.

Commands

The commands that are being run by the asg_info are clustered into three groups.

- System commands - run on SMO
- Commands that are executed only on one SGM of each chassis
- Commands that are executed on all blades

The output of the three groups is written to the file gasginfo_output.gz located in asg_report.<timestamp> folder.

Files

asg_info collects certain files from all SGMs.

SGM ID is added to file names, in order to indicate where data was collected from

For example:

Filename format for files that are part of coredump.tar.gz:

coredump_1_3.tar.gz

coredump_2_5.tar.gz

The first one was collected from SGM 3 in chassis 1, and the second was collected from SGM 5 in chassis 2.

No other files exist in coredump folder, which means that all the other SGM didn’t have any information to send.

General

Information about core dumps created by the system can be found in core.txt.
**Syntax** asg_info [SGMs list] [-f] [-c] [-i] [-x] [-h]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| [SGMs list] | List of SGMs, default: all up SGMs  
Example: asg_info -a will attempt to collect information from all SGMs, including down SGMs |
| -f | Collect and zip information files |
| -c | Collect and zip cores |
| -i | Collect and zip cpinfo |
| -x | Collect and zip all above files - this operation may take several minutes |
| -h | Display usage message |
Example 1  `asg_info -f`

**Output**

```plaintext
easg_info -f

Collecting asg_info data to file
asg_info -b all: Starting..........................................
..............................................................Done

Collecting log info
Collecting policy info
Collecting info on core files from blades
Collecting blade_config files from blades
Collecting top info from blades
Collecting cpha_policy files from blades
Collecting routed.log files from blades
No files found

Collecting routed.conf file from blades
Collecting bond_init log files from blades
Collecting fwid_elg files from blades
Collecting distribution mode information from blades
Collecting anaconda logs files from blades
Collecting alert log files from blades
Collecting chassis.conf file from blades
Collecting start_mbs log file from blades
Collecting mbs log file from blades
Collecting cpd log file from blades
Operation finished successfully
File: asg_report.1324970910.tar.gz is located at: /tmp

2 cores were found, to collect them please run "asg_info -c"
To collect cpinfo please run "asg_info -i"
```

**Comments**  This option handles the collection of relatively light-weight information. It should finish within few minutes.

Example 2  `asg_info -x`
Collecting System Information (asg_info)

Output

```
  asg_info -x

  Collecting asg_info data to file
  asg_info -b all: Starting........................................
  ..............................................................
  ......................Done

  Collecting log info
  Collecting policy info
  Collecting core files from blades
  Collecting info on core files from blades
  Collecting ipinfo content from blades
  This may take a few minutes
  Collecting blade_config files from blades
  Collecting top info from blades
  Collecting cpha_policy files from blades
  Collecting routed.log files from blades
  No files found

  Collecting routed.conf file from blades
  Collecting bond_init log files from blades
  Collecting fwd.elg files from blades
  Collecting distribution mode information from blades
  Collecting anaconda logs files from blades
  Collecting alert log files from blades
  Collecting chassis.conf file from blades
  Collecting start_mbs log file from blades
  Collecting mbs log file from blades
  Collecting cpd log file from blades
  Collecting SLL info from CMM
  This may take a few minutes
  Operation finished successfully
  File: asg_report.1324971116.tar.gz is located at: /tmp
```

Comments

This command collects all available data. Its run time is relatively high and may exceed 10 minutes.

Example 3  

```
  asg_info -c
```

Comments

This command collects core dump from the SGM if available.
Verifiers

MAC Verifier (mac_verifier)

Description:
In the 61000 Security System system, each MAC address contains information about the Chassis ID, Blade ID and interface.

mac_verifier utility will verify that all vmac on ethx-yz interfaces and bond interfaces are the same on all blades on the same chassis.

Usage:
mac_verifier - verify MAC address consistency on both chassis
mac_verifier -l - verify MAC address consistency on local chassis
mac_verifier -v - verbose, display each interface MAC
mac_verifier -h - help screen

Output example:

[Expert@hpdcake35-ch01-01]# mac_verifier
Starting mac address verification on local chassis... (Chassis 1)
No inconsistency found on local chassis
Starting mac address verification on remote chassis... (Chassis 2)
No inconsistency found on remote chassis
[Expert@hpdcake35-ch01-01]#

Output example when inconsistency found:

[Expert@hpdcake35-ch01-01]# mac_verifier
Starting mac address verification on local chassis... (Chassis 1)
MAC address inconsistency found on interface eth1-03
Starting mac address verification on remote chassis... (Chassis 2)
No inconsistency found on remote chassis
[Expert@hpdcake35-ch01-01]#

Bond Verifier (asg_bond_verifier)

Description:
asg_bond_verifier is a utility which check if there are bond configuration problems.
The utility display configured bond interface, bond mode and bond slaves.
If bond mode is LACP, it also checks for sync with the remote switch.
L2 Bridge Verifier (asg_br_verifier)

Description:

asg_br_verifier is a utility which check if there are bridge configuration problems.

Usage:

```
asg_br_verifier  run bridge verification
asg_br_verifier -v  run bridge verification including table entries
```

Output example:

```
Checking for distribution mode conflicts:
Distribution mode is General
Status: OK
```

```
Number of entries in fdb_shadow table:
-**- 2 blades: 2_01 2_02 2_03 -**-
 15
Status: OK
```
Troubleshooting

Output of `asg_br_verifier -v` when there is a miss configuration:

```
Checking for distribution mode conflicts:
Distribution mode is General
Status: OK
Number of entries in fdb_shadow table:
  -x 2 blades: 2_01 2_02 -x-
    15
  -x 1 blade: 2_03 -x-
    0
Status: number of entries is different
Collecting table info from all SGMs, may take few seconds...
Table entries in fdb_shadow table:
  -x 2 blades: 2_01 2_02 -x-
    MAC address="AA:01:00:00:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:02:00:00:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:00:00:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:01:00:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:02:00:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:03:00:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:03:01:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:03:02:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:03:03:00:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:03:03:01:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:03:03:02:00" Interface="eth1-01" product="VPN-1"
    MAC address="AA:03:03:03:03:00" Interface="eth1-01" product="VPN-1"
  -x 1 blade: 2_03 -x-
    fdb_shadow table is empty
Status: Table entries in fdb_shadow table is different between SGMs
```

Policy Verification (asg policy verify)

Use this command to make sure that all Security Gateway Modules have the same firewall policy

Syntax  asg policy verify
Dynamic Routing Verifier (asg_dr_verifier)

Description:

This utility will collect information regarding dynamic routing protocols configured on the system, and will check for inconsistency among blades.

```
[Expert@CCDC-ch01-01]$ asg_dr_verifier

Verifiers

Troubleshooting

Page 235

Output  SGX : Policy Name:   Policy Date:   Policy Signature:   Verification:
1_01    any_any_accept  14Jul11 10:23  fc68872c81  OK
1_02    any_any_accept  14Jul11 10:23  fc68872c81  OK
1_03    any_any_accept  14Jul11 10:23  fc68872c81  OK
2_01    any_any_accept  14Jul11 10:23  fc68872c81  OK
2_02    any_any_accept  14Jul11 10:23  fc68872c81  OK
2_03    any_any_accept  14Jul11 10:23  fc68872c81  OK

OK shows in the verification column.
```

```
Verifiers

Troubleshooting

Page 235

Dynamic Routing Verifier (asg_dr_verifier)

Description:

This utility will collect information regarding dynamic routing protocols configured on the system, and will check for inconsistency among blades.

```
[Expert@CCDC-ch01-01]$ asg_dr_verifier

Verifiers

Troubleshooting

Page 235

```

```
If the "all" parameter is specified, factors which are not indicate on inconsistency problem will also be compared (disable smart compare).

In this example, the "dead" timeout in OSPF will cause inconsistency:

```
[Expert@ACCC-CH02-01# asg_dr_verifier all

General:
  DR Manager: SGM 2_01

Routes status:
  -- 8 blades: 1_01 1_02 1_03 1_04 2_01 2_02 2_03 2_04 -->

OSPF:

  OSPF interfaces -
  -- 8 blades: 1_01 1_02 1_03 1_04 2_01 2_02 2_03 2_04 -->

  Name    IP Address   Area ID  State  NC  DR Interface  BDR Interface Errors
  eth2-01 192.168.33.85 0,0,0,0  ERP 1 192.168.33.235 192.168.33.85 0

  Status: OK

  OSPF neighbors:
  -- 6 blades: 1_01 1_02 1_03 1_04 2_01 2_03 -->

  Neighbor ID            Pri  State  Dead   Address   Interface       Errors
  192.168.33.235         1    FULL/DR 33    192.168.33.235 192.168.33.86 0

  -- 2 blades: 2_02 2_04 -->

  Neighbor ID            Pri  State  Dead   Address   Interface       Errors
  192.168.33.235         1    FULL/DR 32    192.168.33.235 192.168.33.86 0

  Status: Inconsistency found on some of the SGMs

  BGP:

  BGP is not configured on this host
```

Verifiers
Verifying Port Connectivity (asg_pingable_hosts)

**Description**
Use this command to verify 61000 Security Systems ports are properly connected to their hosts. By enabling the pingable_hosts utility, the system constantly performs connectivity tests for each host configured per port. A fixed Chassis pnote factor (default: 50) is added to the chassis grade calculation. When all the port's hosts fail to respond, the chassis grade is lowered by that pnote factor.

**Syntax**
- `asg_pingable_hosts < status | load_ips | disable >`
- `asg_pingable_hosts enable [-i interval] [-monitor]`

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Shows the latest status</td>
</tr>
<tr>
<td>load_ips</td>
<td>Loads a user defined list of host IP addresses that SSM ports should be connected to.</td>
</tr>
<tr>
<td></td>
<td>- The pingable hosts IP file is located at: $FWDIR/conf/pingable_hosts.ips</td>
</tr>
<tr>
<td></td>
<td>- The file contains instructions on how to add new hosts to the list prior to running the load_ips command.</td>
</tr>
<tr>
<td></td>
<td>- After adding hosts to the list, run: <code>asg_pingable_hosts load_ips</code></td>
</tr>
<tr>
<td>disable</td>
<td>Disables the pingable host’s utility.</td>
</tr>
<tr>
<td>enable</td>
<td>Enables the utility. When this parameter is called by itself:</td>
</tr>
<tr>
<td></td>
<td>- Monitor mode is disabled</td>
</tr>
<tr>
<td></td>
<td>- The interval between arps is set to 4 seconds</td>
</tr>
<tr>
<td>-i &lt;interval&gt;</td>
<td>Sets the interval in seconds between arps</td>
</tr>
<tr>
<td>-monitor</td>
<td>Enables monitor mode only</td>
</tr>
</tbody>
</table>

**Example**
`asg_pingable_hosts enable`

**Output**
```
> asg_pingable_hosts enable
Command completed successfully
No additional settings, using default values:
  enable=1 interval=4 monitor=0
> 
```
### SIC reset (g_cpconfig sic init)

#### Description

Use this command to reset Secure Internal Communication (SIC) between the gateway and the Security Management server. For example, if you replace the management server, you must reset the SIC. Note: Rest SIC procedure cause traffic outage as all SGMs are rebooted while the local SGM performs cpstop and cpstart.

#### To reset SIC:

1. Using a console connection to the gateway
   a) Run: `asg stat -i tasks` to find out which SGM is the SMO MASTER.
   (During the sic reset procedure, the SMO SGM is the only SGM that does not reboot.)
   b) Exit gclish to the Bash shell.
   c) On the SMO SGM, run from gclish: `cpconfig sic init <activation key>`.
   Reset is completed after 3-5 minutes.

2. In SmartDashboard:
   a) Open the gateway's **General Properties > Communication** window.

#### Comments

- When running `asg stat` after enabling pingable hosts, each chassis shows the pingable hosts pnote factor:

<table>
<thead>
<tr>
<th>System Status</th>
<th>Up time</th>
<th>02:34:06 hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current CPUs load average</td>
<td>14 %</td>
<td>Concurrent connections</td>
</tr>
<tr>
<td>Health</td>
<td>NORMAL</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis 1</th>
<th>ACTIVE</th>
<th>UP / Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGMs</td>
<td>3 / 3</td>
<td></td>
</tr>
<tr>
<td>Ports</td>
<td>2 / 2</td>
<td></td>
</tr>
<tr>
<td>Fans</td>
<td>4 / 4</td>
<td></td>
</tr>
<tr>
<td>SSUs</td>
<td>2 / 2</td>
<td></td>
</tr>
<tr>
<td>CMMs</td>
<td>2 / 2</td>
<td></td>
</tr>
<tr>
<td>Power Supplies</td>
<td>6 / 6</td>
<td></td>
</tr>
<tr>
<td><strong>Pingable Hosts</strong></td>
<td>1 / 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis 2</th>
<th>STANDBY</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SGMs</td>
<td>3 / 3</td>
<td></td>
</tr>
<tr>
<td>Ports</td>
<td>2 / 2</td>
<td></td>
</tr>
<tr>
<td>Fans</td>
<td>4 / 4</td>
<td></td>
</tr>
<tr>
<td>SSUs</td>
<td>2 / 2</td>
<td></td>
</tr>
<tr>
<td>CMMs</td>
<td>2 / 2</td>
<td></td>
</tr>
<tr>
<td>Power Supplies</td>
<td>6 / 6</td>
<td></td>
</tr>
<tr>
<td><strong>Pingable Hosts</strong></td>
<td>1 / 1</td>
<td></td>
</tr>
</tbody>
</table>

- The **UP/Required** column shows the pnote status, not the number of pingable hosts up or required. The status means:
  - 1 / 1 when OK
  - 0 / 1 when one of the pingable hosts on the list fails to reply
- Pingable host log files are stored under `/var/log/pingable_hosts`
- Pingable hosts default factor is 50. That factor can be changed by:
  > set chassis high-availability factors pnote pingable_hosts <factor>
b) Click **Reset**.

c) Enter the same activation key used in step 1.

d) Click **Initialize**.

3. On the gateway, run from gclish: `cpconfig sic state`.
   Make sure that **Trust** is established.
   ```bash
   [Expert@cpmodule-ch01-01]# g_cpconfig sic state
   -- 4 blades: 1_01 1_02 2_01 2_02 --
   Trust State: Trust established
   ```

### Troubleshooting SIC reset

SIC reset requires 3-5 minutes. If SIC reset was interrupted (for example by loss of network connectivity), run: `g_cpconfig sic state` to get the SIC state. If the SIC State is:

<table>
<thead>
<tr>
<th>SIC state</th>
<th>Do this:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trust established</td>
<td>Repeat the SIC reset procedure</td>
</tr>
<tr>
<td>Initialized but Trust was not established</td>
<td>1. Reboot all SGMs</td>
</tr>
<tr>
<td></td>
<td>2. In <strong>SmartDashboard &gt; General Properties &gt; Communication</strong> window initialize SIC</td>
</tr>
<tr>
<td></td>
<td>3. Install the policy.</td>
</tr>
</tbody>
</table>

### SIC Cleanup

To resolve other SIC issues, do a SIC cleanup. There are two ways to do a SIC cleanup:

Run: `asg_blade_config reset_sic -reboot_all <activation_key>`. Or:

1. Shutdown all SGMs (but not the SMO SGM) using the **ccutil** command in the Bash shell.
2. Connect to the SMO SGM using a serial connection.
3. In **SmartDashboard > General Properties > Communication** initialize SIC.
4. Install a policy on the SMO SGM.
5. Turn on all SGMs.

### Hardware

This section describes known problems encountered with 61000 Security System hardware components along with its corresponding resolution.

**Hardware components:**

1. **Security Gateway Module (SGM)**
2. **Chassis Management Module (CMM)**
3. **Security Switch Module (SSM)**
4. **Fans**
5. **Power Supply Unit (PSU)**
Security Gateway Module (SGM)

Problem: SGM does not detect part of its RAM
Cause: One or more DIMMs are not properly installed
Resolution: Re-assemble DIMMs in problematic SGM
Validation: Run `asg resource` to verify all SGMs properly report its RAM size

Problem: SGM fails to boot and does not enter BIOS
Cause: SGM BIOS is corrupted
Resolution: CMOS reset:
   1. On the left side of the SGM there is a yellow jumper
      This jumper resides on the leftmost two pins (pins 1 and 2)
   2. Pull the jumper out and put it on pins 2 and 3
   3. Keep state 2 for 10 seconds
   4. Pull the jumper out and put it back on pins 1 and 2
Validation: SGM will start loading
   User will be able to enter BIOS

Problem: SGM fails to start, SSD not detected at boot time
Cause: SD is not properly assembled
Resolution: Re-assemble SSD connectors.
   1 connector to the SSD itself and the 2 others to the motherboard
Validation: SGM will start loading

Problem: SGM fails to boot and constantly searching for network installation (PXE)
Cause: There is no image loaded on the SGM
Resolution: Install image from CD/PXE or USB flash drive.
   Make sure BIOS setup is set to boot from the option you chose
Validation: SGM will start installing new image

Problem: Blue LED is on
Cause: SGM is not properly attached to its slot
Resolution: Reseat the SGM, tighten its thumb screws and make sure handles are firmly closed
Validation: Blue LED should disappear and SGM will start loading

Problem: SGM speed LEDs are not yellow/orange
Cause: SGM is connected to SSM with wrong speed
Resolution:
   1. Restore manufacturing-defaults on the associated SSM.
2. RMA the SGM

**Validation**
Verify that all speed LEDs on SGM are yellow/orange

**Problem**
SGM constantly boots after it was down and major configuration changes were made to the system

**Cause**
Old configuration conflicts with existing configuration

**Resolution**
1. Export snapshot from both problematic and stable SGMs and attach to support ticket
2. Make sure FCD image is aligned with existing SGMs
   Note: FCD is created during clean installation
3. Revert to FCD image

**Validation**
SGM should join security group, pull the configuration and become up

**Problem**
CPU type does not match customer's order, CPU type test fails in asg diag

**Cause**
Customer received SGM220 instead of SGM220T or vice versa

**Resolution**
MA the SGM

**Validation**
N/A

**Problem**
All SGMs beyond certain amount of time fail to boot and enter blue LED

**Cause**
Some power supply units are not connected. Minimum of 4 PSUs is required for fully populated system

**Resolution**
Make sure all PSUs are properly attached to the chassis by pushing the insertion latch. Verify that all power cords are plugged

**Validation**
All SGMs in the chassis should be able to start

---

**Chassis Management Module (CMM)**

**Problem**
Blue LED on the CMM

**Cause**
CMM is not properly assembled

**Resolution**
Reseat the CMM

**Validation**
Verify the Blue LED on the CMM turned off

**Problem**
Power Supplies are not monitored

**Cause**
Chassis type is not configured properly in the CMM

**Resolution**
1. Login via serial console to the CMM and repeat the CMM installation process (install.sh).
2. Select correct chassis type:
   - Telkor - 3 PSUs per 1U
   - Lambda - 5 PSUs per 1U
3. In case of dual CMM, perform it on each CMM individually.
Validation  Run asg stat -v and check the power supplies amount

Problem  Failed to install 2nd chassis in dual chassis setup
Cause  Chassis ID is identical on both chassis
Resolution
  6. Login via serial console to the CMM
  7. Set the chassis ID by editing the $SHMM_CHASSID$ in /etc/shmm.cfg
  8. Reboot the CMM
Validation  Check whether the installation on the 2nd chassis works

Problem  CMM firmware is different after CMM failover
Cause  CMM firmware mismatch in a dual CMM
Resolution  Upgrade the faulty CMM individually and reseat the other CMM
Validation  Verify version by invoking asg_version

Problem  No connectivity to the CMM through one of the CIN interfaces
Cause  CMM interface is set to the front panel instead of the backplane
Resolution
  1. Remove CMM
  2. Change JP4 jumpers' position to 2-3
  3. Plug in the CMM
Validation  Run asg stat -v and check the CMM amount

Problem  Alarm LED is on
Cause  High temperature in chassis surroundings
Resolution
  1. Login via serial console to the CMM and reset LED by running: clia alarm 0
  2. Make sure that all open slots (missing SGMs/CMMs/PSUs) are covered with blanks
  3. Verify that all fans are properly attached
  4. Make sure there is proper cooling in chassis surroundings
Validation  Alarm LED should remain off

## Security Switch Module (SSM)

Problem  Blue LED on the SSM
Cause  SSM is not properly assembled
Resolution  Reseat the SSM
Validation  Verify the Blue LED on the SSM turned off
**Problem:** asg dxl dist_mode verify failed and there are traffic issues on pseudo interfaces

**Cause:** SSM distribution configuration is not set properly

**Resolution:** re-set distribution mode and verify it

In case it didn't solve, login into the appropriate SSM and invoke
load-balance apply

As a last resort, invoke system reload manufacturing-defaults

**Validation:** Invoke asg dxl dist_mode verify

---

**Problem:** Connectivity issues between SGMs

**Cause:** Invalid SSM configuration on Sync interfaces

**Resolution:** Login into the appropriate SSM and invoke
system reload manufacturing-defaults

**Validation:** Verify connectivity between the SGMs by invoking asg monitor

---

**Problem:** Connectivity issues between SGMs on different chassis

**Cause:** Sync ports are connected through 1G link/transceivers are not compatible with distance

**Resolution:** Connect the Sync ports to 10G links, using LC fiber optic. Make sure to use SR/LR transceivers, according to distance

**Validation:** Verify connectivity between SGMs on different chassis by invoking asg monitor

---

**Problem:** No link on SSM traffic port

**Cause:** Uncertified transceivers or incorrect port speed

**Resolution:**

1. Check if the transceivers are certified by invoking the command
   asg diag verify and check whether the "Media Details" Passed
2. Verify the port speed by invoking:
   asg_chassis_ctrl get_port_admin_speed <SSM ID> <port ID>

   If needed, set by invoking
   asg_chassis_ctrl set_port_speed <SSM ID> <port ID> <speed>

**Validation:** Verify link on the port and connectivity (if possible)

---

**Problem:** No link on SSM management port

**Cause:** Uncertified transceivers or incorrect port speed

**Resolution:**

1. Check if the transceivers are certified by connecting them to one of the traffic ports and invoke transceiver_verifier
2. Verify the port speed
   a. login to the relevant SSM
   b. Run show port 1/5/<management port index, 1-4>
3. Reset port speed
   a. login to the relevant SSM
b. Enter `conf t`

c. Set port speed to different value:
   `port 1/5/<management port index> speed 100`

d. Set port speed back to the desired value:
   `port 1/5/<management port index> speed 1000`

**Validation**  Verify link on the port and connectivity (if possible)

**Problem**  Silent installation on other SGMs does not work after FTW finished

**Cause**  SSM version is incorrect or has an invalid configuration

**Resolution**  Login into the appropriate SSM and verify version by invoking `show version`
   In case the version is incorrect, please upgrade the SSM
   Otherwise, invoke `system reload manufacturing-defaults`

**Validation**  Verify that silent installation on other SGMs completed after 5-10 minutes

## DC deployment

**Wiring instructions:**  Each PEM unit is capable of providing enough power to support a fully populated chassis. Each PEM consists of four circuits that power different parts of the chassis (combination of SGMs, SSMs, fans, etc.). Each of the four circuits on each PEM unit is backed by a corresponding circuit on the other PEM unit. Therefore, in order to guarantee consistent power supply in case of power source failure each PEM unit should be connected to a different power source.

**Monitoring mechanism:**  The chassis monitor daemon queries the CMM for the PEM unit’s “presence” value. This allows the daemon to determine whether a PEM unit is present and is connected to a power source.

**Current limitations:**  The PEM’s presence value does not allow the chassis monitor daemon to determine the status of any specific power circuit. Therefore, it is unable to detect whether only one of the power circuits is disconnected.

   Monitoring through “asg hw_monitor” and “asg stat –v” is limited to an entire PEM unit. Also, “power supply down” message will only be received in case an entire PEM unit is disconnected.

**Examples:**

1. Shutting down power circuit #1 on one of the PEM units will cause no power failure because circuit #1 on the other PEM unit will provide enough power to support the required components. But, although power redundancy is lost, chassis monitor daemon will not be able to detect this.
   Further shutting down power circuit #1 on the other PEM unit will cause a power failure, as some of the components will lose their power supply. The chassis
Hardware

Troubleshooting

Page 245

[Protected] For public distribution

monitor daemon will now detect that those components are down, but will not report about any PEM unit issues.
2. In case of an entire PEM unit failure, chassis monitor daemon will detect that the unit is not present and will report about the problem.

Fans

<table>
<thead>
<tr>
<th>Problem</th>
<th>Blue LED on fan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause</td>
<td>Fan is not properly assembled</td>
</tr>
<tr>
<td>Resolution</td>
<td>Reseat the fan and lock the captive screw (where applicable)</td>
</tr>
<tr>
<td>Validation</td>
<td>Verify the blue LED on the fan is turned off and all fans rotate at normal speed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem</th>
<th>Rotation speed is too high, fans are extremely noisy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause</td>
<td>High temperature in chassis surroundings</td>
</tr>
<tr>
<td>Resolution</td>
<td></td>
</tr>
<tr>
<td>1. Make sure that all open slots (missing SGMs/CMMs/PSUs) are covered with blanks.</td>
<td></td>
</tr>
<tr>
<td>2. Verify that all fans are properly attached.</td>
<td></td>
</tr>
<tr>
<td>3. Make sure there is proper cooling in chassis surroundings</td>
<td></td>
</tr>
<tr>
<td>Validation</td>
<td>asg hw_monitor indicates that fans rotation speed is normal and the threshold is not crossed</td>
</tr>
</tbody>
</table>

Power Supply Unit (PSU)

<table>
<thead>
<tr>
<th>Problem</th>
<th>Blue LED on PSU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause</td>
<td>PSU is not properly assembled</td>
</tr>
<tr>
<td>Resolution</td>
<td>Make sure all PSUs are properly attached to the chassis by pushing the insertion latch</td>
</tr>
<tr>
<td>Validation</td>
<td>Run asg stat -v and check PSUs amount</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem</th>
<th>After chassis RMA: rightmost chassis components (SGMs, PSUs) are not monitored</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause</td>
<td>DC PEMs are missing</td>
</tr>
<tr>
<td>Resolution</td>
<td>Move DC PEMs from old (RMAed) chassis to new chassis</td>
</tr>
<tr>
<td>Validation</td>
<td>Verify that all chassis components are monitored</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem</th>
<th>asg diag reports that power unit is misplaced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause</td>
<td>Telkor PSUs should be placed from upper right to bottom left</td>
</tr>
<tr>
<td>Resolution</td>
<td>Place 5 Telkor PSUs as follows: 3 in upper tray, 2 in bottom tray. Leave the leftmost bay in bottom tray empty and covered with blank</td>
</tr>
<tr>
<td>Validation</td>
<td>asg diag should not warn about PSU misplacement</td>
</tr>
</tbody>
</table>
Debug files

These are the debug files that relate to the 61000 Security System:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Debug File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy</td>
<td>/var/log/cpha_policy.log.*</td>
</tr>
<tr>
<td>SGM Configuration / Pull Configuration</td>
<td>/var/log/blade_config.*</td>
</tr>
<tr>
<td>Alerts</td>
<td>/var/log/dist_mode.log.*</td>
</tr>
<tr>
<td>Installation – OS</td>
<td>/var/log/anaconda</td>
</tr>
<tr>
<td>Installation – the 61000 Security System</td>
<td>/var/log/start_mbs.log</td>
</tr>
<tr>
<td>Installation – the 61000 Security System</td>
<td>/var/log/mbs.log</td>
</tr>
<tr>
<td>Dynamic Routing</td>
<td>/var/log/routed.log</td>
</tr>
<tr>
<td>CPD</td>
<td>$CPDIR/log/cpd.elg</td>
</tr>
<tr>
<td>FWD</td>
<td>$FWDIR/log/fwd.elg</td>
</tr>
<tr>
<td>General</td>
<td>/var/log/messages*</td>
</tr>
<tr>
<td>SMD</td>
<td>/var/log/smd_smo.log</td>
</tr>
<tr>
<td>SMD</td>
<td>/var/log/smd.log</td>
</tr>
<tr>
<td>Log servers</td>
<td>/var/log/log_servers*</td>
</tr>
<tr>
<td>Pingable hosts</td>
<td>/var/log/pingable_hosts*</td>
</tr>
<tr>
<td>Clish auditing</td>
<td>/var/log/auditlog*</td>
</tr>
<tr>
<td>Command auditing</td>
<td>/var/log/asgaudit.log*</td>
</tr>
<tr>
<td>VPND</td>
<td>$FWDIR/log/vpnd.elg*</td>
</tr>
</tbody>
</table>