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 R75.035 version.

Latest Documentation
The latest version of this document is at:
http://supportcontent.checkpoint.com/documentation_download?ID=12558
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>19 December 2011</td>
<td>First release of this document</td>
</tr>
<tr>
<td>8 February 2012</td>
<td>Adding VLAN enhancements</td>
</tr>
<tr>
<td>9 February 2012</td>
<td>Adding asg_info</td>
</tr>
<tr>
<td>14 February 2012</td>
<td>Update CMM debug section</td>
</tr>
<tr>
<td>19 February 2012</td>
<td>Adding official version names</td>
</tr>
<tr>
<td>13 March 2012</td>
<td>Adding 61000 R75.035 new commands</td>
</tr>
<tr>
<td>19 March 2012</td>
<td>Upgrade Procedure – adjustment</td>
</tr>
<tr>
<td>27 March 2012</td>
<td>Reset SIC adjustment</td>
</tr>
<tr>
<td>4 April 2012</td>
<td>Software blades updates</td>
</tr>
<tr>
<td>2 July 2012</td>
<td>Chassis HA - Link Preemption Mechanism</td>
</tr>
<tr>
<td>19 July 2012</td>
<td>Upgrade SSM60 to SSM160</td>
</tr>
<tr>
<td>22 July 2012</td>
<td>Radius, Chassis ID configuration, 61K LEDs</td>
</tr>
</tbody>
</table>

Feedback
Check Point is engaged in a continuous effort to improve its documentation.
Please help us by sending your comments
Important Information ............................................................................................................. 3
61000 Security Systems Monitoring and Information Gathering ............................................. 6
  Showing Chassis and Component State (asg stat) ............................................................. 6
  Setting the Required Number of Components per Chassis .............................................. 10
  Setting the Unit Weight .................................................................................................... 10
  Connecting to a specific SGM (blade command) ........................................................... 11
  Monitoring Chassis and Component Status (asg monitor) .............................................. 12
  Showing Hardware Information for Monitored Components (asg hw_monitor) ............ 14
  Showing Security Gateway Module Resource Information (asg resource) .................... 17
  Showing Interface Status (asg if) ....................................................................................... 18
  Showing the Routing Tables (asg_route) ......................................................................... 20
  Showing SGM Information (asg_blade_stats) .................................................................. 22
  Showing Traffic Information (asg_ifconfig) ..................................................................... 22
    Native (asg_ifconfig) ..................................................................................................... 24
    Analyze (asg_ifconfig analyze) .................................................................................... 24
    Banalyze (asg_ifconfig banalyze) ................................................................................ 26
    Internal interfaces ........................................................................................................ 29
  Showing SSM Traffic Statistics (asg_traffic_stats) ........................................................ 29
  Monitoring Key Performance Indicators and Load Statistics (asg perf) ......................... 30
  Searching for a Connection (asg search) .......................................................................... 31
  Configuring Alerts for SGM and Chassis Events (asg alert) ......................................... 32
    Redirecting Alerts Messages to External syslog server (asg_syslog) ......................... 34
  Command Auditing ............................................................................................................. 34
    Log Utility (asg log) ...................................................................................................... 35
  Displaying System Messages (asg_varlog) ....................................................................... 37
  Chassis Control (asg_chassis_ctrl) .................................................................................. 38
  System Under Load .......................................................................................................... 40
  Showing the firewall Database Configuration (asg config) ............................................. 42
  Showing the Number of Firewall and SecureXL Connections (asg conns) .................... 44
  Showing Software and Firmware versions (asg_version) ............................................... 45
  Showing the Auditlog (asg_auditlog) .............................................................................. 47
  gclish ................................................................................................................................. 49
  CP global commands....................................................................................................... 49
  OS global commands ...................................................................................................... 53
    Global Commands Generated by CMM ......................................................................... 57
  General global commands .............................................................................................. 58
  Tcpdump - multi-blade capture (tcpdump –mcap) ............................................................. 60
  Collecting System Information (asg_info) ...................................................................... 62
  Collecting System Diagnostics (asg diag) ...................................................................... 65
  Collecting System Serial Numbers .................................................................................. 67
61000 Security Systems Configuration ................................................................................ 69
  Configuring Security Gateway Modules as Up or Down (asg_blade_admin) ................. 69
  Configuring a Chassis as Up or Down (asg_chassis_admin) ........................................... 70
  Security Group (asg_security_group) ............................................................................... 71
  Distribution Modes .......................................................................................................... 72
    Manually Configuring Distribution Modes (asg_dxl dist_mode) ................................... 74
    Reconfiguring Distribution Modes ............................................................................... 76
    NAT and the Correction Layer .................................................................................... 77
  Configuring IPv6 Support (asg_dxl ipv6) ...................................................................... 77
  Configuring Link Aggregation (Bonding) ....................................................................... 80
    Creating a Bonding Group ............................................................................................ 80
    Setting a Bonding Mode ............................................................................................... 81
Chapter 1

61000 Security Systems Monitoring and Information Gathering

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</td>
<td>tasks</td>
</tr>
<tr>
<td></td>
<td>proc</td>
</tr>
<tr>
<td></td>
<td>all_ids</td>
</tr>
</tbody>
</table>
Showing Chassis and Component State (asg stat)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Shows</th>
</tr>
</thead>
<tbody>
<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>
<tr>
<td>chassis_monitor</td>
<td>Which SGM handles the chassis_monitor process (usually the local SGM)</td>
</tr>
</tbody>
</table>

Example 1

```
asg stat
```

Output

```
<table>
<thead>
<tr>
<th>System Status</th>
<th>Up time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>13:55:50 hours</td>
</tr>
</tbody>
</table>

| Current CPUs load average | 10 % |
| Concurrent connections   | 65   |
| Health                   |      |
| SGMs                     | 3 Inactive |

<table>
<thead>
<tr>
<th>Chassis</th>
<th>State</th>
<th>UP / Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>STANDBY</td>
<td>9 / 12 (1)</td>
</tr>
<tr>
<td></td>
<td>Ports</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>Fans</td>
<td>6 / 6</td>
</tr>
<tr>
<td></td>
<td>SSMs</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>CMMs</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>Power Supplies</td>
<td>5 / 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chassis</th>
<th>State</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>ACTIVE</td>
<td>12 / 12</td>
</tr>
<tr>
<td></td>
<td>Ports</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>Fans</td>
<td>6 / 6</td>
</tr>
<tr>
<td></td>
<td>SSMs</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>CMMs</td>
<td>2 / 2</td>
</tr>
<tr>
<td></td>
<td>Power Supplies</td>
<td>5 / 5</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
```
Output

| System Status | | |
| Chassis 1 | ACTIVE | | |

<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>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>

| Chassis Parameters | | |
|---|---|---|---|---|
| unit | chassis 1 | chassis 2 | Unit Weight |
| SGMs | 8 / 8 | 8 / 8 | 8 / 8 |
| Ports Priority | Standard | 2 / 2 | 2 / 2 | 11 |
| Other | 0 / 0 | 0 / 0 | 0 / 0 |
| Sensors | Fans | 6 / 6 | 6 / 6 | 6 |
| SSMS | 2 / 2 | 2 / 2 | 11 |
| CMMs | 2 / 2 | 2 / 2 | 11 |
| Power Supplies | 2 / 2 | 2 / 2 | 6 |
| Chassis Grade | 180 / 180 | 180 / 180 |

- Minimum threshold for traffic processing: 12
- Minimum grade gap for chassis failover: 11

Synchronization:
- Within chassis: Enabled (default)
- Between chassis: Enabled (default)
- Exception Rules: (default)
- Distribution: Controlled
- Control Blade: Disabled (default)
- Chassis HA mode: Active UP

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_blade_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 = (Unit Weight) \times (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>Chassis Parameters</th>
<th>chassis 1</th>
<th>chassis 2</th>
<th>unit weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGMs</td>
<td>8 / 8</td>
<td>8 / 8</td>
<td>12</td>
</tr>
<tr>
<td>Ports Priority</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td></td>
</tr>
<tr>
<td>Other</td>
<td>0 / 0</td>
<td>0 / 0</td>
<td>6</td>
</tr>
<tr>
<td>Sensors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fans</td>
<td>6 / 6</td>
<td>6 / 6</td>
<td>6</td>
</tr>
<tr>
<td>SGMs</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td>11</td>
</tr>
<tr>
<td>Downs</td>
<td>2 / 2</td>
<td>2 / 2</td>
<td>11</td>
</tr>
<tr>
<td>Power Supplies</td>
<td>3 / 3</td>
<td>3 / 3</td>
<td>6</td>
</tr>
<tr>
<td>Chassis grade</td>
<td>328 / 328</td>
<td>328 / 328</td>
<td></td>
</tr>
</tbody>
</table>

Minimum threshold for traffic processing: 12
Minimum grade gap for chassis failover: 11

- **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
Distribution: 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, you always have immediate access to the system through an SSH connection.
Setting the Required Number of Components per Chassis

Running `asg stat` shows the chassis and components 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 units</td>
<td><code>&gt;set chassis id 1 modules_amount power_units 3</code></td>
</tr>
<tr>
<td></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`

<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>Port</td>
<td><code>&gt;set chassis high-availability factors Port &lt;factor&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA Port Priorities Factors</td>
</tr>
<tr>
<td>Fans</td>
<td><code>&gt;set chassis high-availability factors sensor fans &lt;factor&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA Fans Sensor Factor</td>
</tr>
<tr>
<td>SSMs</td>
<td><code>&gt;set chassis high-availability factors sensor ssm &lt;factor&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA SSMs Sensor Factor</td>
</tr>
<tr>
<td>power_supplies</td>
<td><code>&gt;set chassis high-availability factors sensor power_supplies &lt;factor&gt;</code></td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA Power Supplies Sensor Factor</td>
</tr>
</tbody>
</table>
### 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 from `<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.

### Component Command

<table>
<thead>
<tr>
<th>Component</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMMs</td>
<td>&gt; set chassis high-availability factors sensor cmm &lt;factor&gt;</td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA CMMs Sensor Factor</td>
</tr>
</tbody>
</table>

- > set chassis high-availability factors pnote

<table>
<thead>
<tr>
<th>Component</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pingable_hosts</td>
<td>&gt; set chassis high-availability factors pnote pingable_hosts &lt;factor&gt;</td>
</tr>
<tr>
<td></td>
<td>Updates the Chassis HA pingable_hosts Factor</td>
</tr>
</tbody>
</table>
## 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: <code>asg monitor 3</code></td>
</tr>
<tr>
<td>-v interval</td>
<td>Monitors chassis parameters. For example: <code>asg monitor -v 3</code>.</td>
</tr>
<tr>
<td>-all interval</td>
<td>Monitors all SGMs and chassis parameters</td>
</tr>
</tbody>
</table>

### Example 1
```
asg monitor
```
```
garwdo30-ch01-01 > asg monitor
Thu Mar 29 15:56:13 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>
<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>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>
Comments

- 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 GW 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_blade_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.

**Example 2**

```
$ asg monitor -v
```

```
gmsd:100-601-01 > asg monitor -v
Thu Mar 29 16:06:41 IST 2012

------------------------------------------------------------------------
| Chassis Parameters         | Chassis 1 | Chassis 2 | Unit Weight |
------------------------------------------------------------------------
| Unit                      | 3 / 3     | 3 / 3     | 6           |
| Ports Priority            | Standard  | 2 / 2     | 11          |
|                           | Other     | 0 / 0     | 6           |
| Sensors                   | Fans      | 4 / 4     | 5           |
|                           | SMs       | 2 / 2     | 11          |
|                           | CMMs      | 2 / 2     | 6           |
|                           | Power Supplies | 6 / 6 | 6 |
| Chassis Grade             | 130 / 130 | 130 / 130 | -           |
------------------------------------------------------------------------

Minimum grade gap for chassis failover: 11

Synchronization
- N+1 in chassis: Enabled (Default)
- Between chassis: Enabled (Default)

Distribution
- Control Blade: Disabled (Default)
- Chassis HI mode: Active Up
```
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:
  
  o 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:
  o Within chassis - between SGMs located in the same chassis
  o Between chassis - between SGMs located in different chassis
  o Exception Rules - user configured exception rules. To configure, use the command g_sync_exception

---

**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.

**Example**
```
  asg hw_monitor
```
<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>Chassis 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CPUM</td>
<td>bay 1</td>
<td>0</td>
<td>0</td>
<td>&lt;$,&gt;0/$&gt;</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>bay 2</td>
<td>0</td>
<td>0</td>
<td>&lt;$,&gt;0/$&gt;</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 1, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 1, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 2, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 2, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 3, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 3, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 4, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 4, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 5, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 5, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 6, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 6, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 7, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 7, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 8, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 8, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 9, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 9, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 10, CPU0</td>
<td>38</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 10, CPU1</td>
<td>40</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 11, CPU0</td>
<td>39</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 11, CPU1</td>
<td>42</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 12, CPU0</td>
<td>39</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
<tr>
<td>CPUM</td>
<td>blade 12, CPU1</td>
<td>43</td>
<td>65</td>
<td>Celsius</td>
<td>1</td>
</tr>
</tbody>
</table>

Fan

<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>Fan</td>
<td>bay 1</td>
<td>6</td>
<td>11</td>
<td>Speed Level</td>
<td>1</td>
</tr>
<tr>
<td>Fan</td>
<td>bay 2</td>
<td>6</td>
<td>11</td>
<td>Speed Level</td>
<td>1</td>
</tr>
<tr>
<td>Fan</td>
<td>bay 3</td>
<td>6</td>
<td>11</td>
<td>Speed Level</td>
<td>1</td>
</tr>
<tr>
<td>Fan</td>
<td>bay 4</td>
<td>6</td>
<td>11</td>
<td>Speed Level</td>
<td>1</td>
</tr>
<tr>
<td>Fan</td>
<td>bay 5</td>
<td>6</td>
<td>11</td>
<td>Speed Level</td>
<td>1</td>
</tr>
</tbody>
</table>

PowerConsumption

<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>PowerConsumption</td>
<td></td>
<td>3050</td>
<td>4050</td>
<td>Watts</td>
<td></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>1</td>
</tr>
<tr>
<td>PowerUnit(AC)</td>
<td>bay 5</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnit(AC)</td>
<td>bay 6</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 1</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 2</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 3</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 4</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 5</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 6</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 7</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 8</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 9</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 10</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 11</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
<tr>
<td>PowerUnitFan</td>
<td>bay 12</td>
<td>0</td>
<td>0</td>
<td>NA</td>
<td>1</td>
</tr>
</tbody>
</table>

SSM

<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>SSM</td>
<td>bay 1</td>
<td>32480</td>
<td>0</td>
<td>Mops</td>
<td>1</td>
</tr>
<tr>
<td>SSM</td>
<td>bay 2</td>
<td>34184</td>
<td>0</td>
<td>Mops</td>
<td>1</td>
</tr>
<tr>
<td>Column</td>
<td>Meaning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>To identify the location, see the 61000 Security Systems Front Panel.</td>
<td></td>
<td></td>
<td></td>
<td></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 32).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Units</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>0 means the component does not exist.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Comments

<table>
<thead>
<tr>
<th>Location</th>
<th>Value</th>
<th>Units</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMH bay 1</td>
<td>0</td>
<td>0</td>
<td>&lt;Sx, Dy/A&gt;</td>
</tr>
<tr>
<td>CMH bay 2</td>
<td>1</td>
<td>0</td>
<td>&lt;Sx, Dy/A&gt;</td>
</tr>
<tr>
<td>CPUtemp blade 1</td>
<td>CPU1</td>
<td>28</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 2</td>
<td>CPU1</td>
<td>40</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 3</td>
<td>CPU1</td>
<td>37</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 4</td>
<td>CPU1</td>
<td>40</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 5</td>
<td>CPU1</td>
<td>38</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 6</td>
<td>CPU1</td>
<td>38</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 7</td>
<td>CPU1</td>
<td>42</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 8</td>
<td>CPU1</td>
<td>40</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 9</td>
<td>CPU1</td>
<td>44</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 10</td>
<td>CPU1</td>
<td>40</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 11</td>
<td>CPU1</td>
<td>47</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 12</td>
<td>CPU1</td>
<td>40</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 13</td>
<td>CPU1</td>
<td>43</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 14</td>
<td>CPU1</td>
<td>39</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 15</td>
<td>CPU1</td>
<td>42</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 16</td>
<td>CPU1</td>
<td>42</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 17</td>
<td>CPU1</td>
<td>43</td>
<td>65</td>
</tr>
<tr>
<td>CPUtemp blade 18</td>
<td>CPU1</td>
<td>42</td>
<td>65</td>
</tr>
<tr>
<td>Fan bay 1</td>
<td>Fan 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fan bay 2</td>
<td>Fan 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fan bay 3</td>
<td>Fan 3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POWERConsumption</td>
<td>W/A</td>
<td>2513</td>
<td>40</td>
</tr>
<tr>
<td>POWERUnit(AC)</td>
<td>bay 1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnit(AC)</td>
<td>bay 2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnit(AC)</td>
<td>bay 3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnit(AC)</td>
<td>bay 4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnit(AC)</td>
<td>bay 5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnit(AC)</td>
<td>bay 6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnitFan</td>
<td>bay 1</td>
<td>Fan 1</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnitFan</td>
<td>bay 2</td>
<td>Fan 2</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnitFan</td>
<td>bay 3</td>
<td>Fan 3</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnitFan</td>
<td>bay 4</td>
<td>Fan 4</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnitFan</td>
<td>bay 5</td>
<td>Fan 5</td>
<td>0</td>
</tr>
<tr>
<td>POWERUnitFan</td>
<td>bay 6</td>
<td>Fan 6</td>
<td>0</td>
</tr>
<tr>
<td>SGM</td>
<td>0</td>
<td>0</td>
<td>Mops</td>
</tr>
<tr>
<td>SEM</td>
<td>0</td>
<td>0</td>
<td>Mops</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>-b sgm</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>
<tr>
<td>-h</td>
<td>Shows usage and exits</td>
</tr>
</tbody>
</table>

**Example**  
`asg resource`
Output

gormio30-ch01-01 > asg resource

<table>
<thead>
<tr>
<th>Resource Name</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>82%</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>

Chassis 2

<table>
<thead>
<tr>
<th>Resource Name</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>81%</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%.

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:
Showing Interface Status (asg if)

- 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.</td>
</tr>
<tr>
<td>Note:</td>
<td>-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>5.5.5.10/24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eth1-Mgt4</td>
<td>00:0c:29:03:mate2</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></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></td>
</tr>
<tr>
<td>eth2-Sync</td>
<td></td>
<td>Bond Slave</td>
<td>up</td>
</tr>
<tr>
<td>(Sync)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Comments**

- 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 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><code>ipv6</code></td>
<td>Shows IPv6 routes (IPv4 is the default)</td>
</tr>
<tr>
<td><code>inactive</code></td>
<td>- Shows inactive routes</td>
</tr>
<tr>
<td></td>
<td>- Only these filters can be used with the inactive parameter:</td>
</tr>
<tr>
<td><code>filter</code></td>
<td>Customized the output with one of these filters:</td>
</tr>
<tr>
<td></td>
<td><strong>Shows inactive:</strong></td>
</tr>
<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><code>filter</code></td>
<td>Shows active</td>
</tr>
<tr>
<td>aggregate</td>
<td>Aggregate routes</td>
</tr>
<tr>
<td>bgp</td>
<td>BGP routes</td>
</tr>
<tr>
<td>destination <code>&lt;address&gt;</code></td>
<td>Route(s) to a specific destination</td>
</tr>
<tr>
<td>direct <code>&lt;address&gt;</code></td>
<td>Directly connected routes</td>
</tr>
<tr>
<td>exact <code>&lt;address&gt;</code></td>
<td>Specific route from a given address</td>
</tr>
<tr>
<td>less-specific <code>&lt;address&gt;</code></td>
<td>Less specific routes from a given address</td>
</tr>
<tr>
<td>more-specific <code>&lt;address&gt;</code></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><code>Summary</code></td>
<td>Summarizes the routing table</td>
</tr>
</tbody>
</table>
Example

```
asg_route -b 1_01,1_02
```

Output

```
Finding Routes info from SDNs:
1_01,1_02

Status: DB Routes info is NOT identical on all SDNs
DE Routes info is NOT identical on all SDNs

Identical DB Routes: (6 records)
  C  127.0.0.1/32   is directly connected, lo
  C  192.0.2.0/24  is directly connected, eth0-CIN
  C  172.16.6.0/24  is directly connected, eth0-Mnt4
  C  192.0.2.0/24  is directly connected, eth0-
  S  0.0.0.0/0     via 172.16.6.4, eth0-Mnt4, cost 0, see <time>

Inconsistent DB Routes:
  D  15.15.15.0/24  is directly connected, eth0-01
  S  200.17.122.0/24 via 172.16.6.4, eth0-Mnt4, cost 0, see <time>

Types: C - Connected, S - Static, R - REP, B - BGP,
O - OSPF IntraArea (IA) - InterArea, E - External, H - HSRP
P - Aggregate, R - Kernel Route, H - Hidden, P - Expressed
SR - Source-Based Routes

Identical OS Routes: (14 records)
  OSR 172.16.6.0/24 dev eth0-Mnt4 proto kernel scope link src 172.16.6.151
  OSR 5.5.5.0/24 dev eth0-01 proto kernel scope link src 5.5.5.10
  OSR broadcast 127.0.0.0 dev lo table 255 proto kernel scope link src 127.0.0.1
  OSR broadcast 172.16.6.0 dev eth0-Mnt4 table 255 proto kernel scope link src 172.16.6.151
  OSR broadcast 172.16.6.0 dev eth0-Mnt4 table 255 proto kernel scope link src 172.16.6.151
  OSR broadcast 172.16.6.0 dev eth0-Mnt4 table 255 proto kernel scope link src 172.16.6.151
  OSR broadcast 5.5.5.0 dev eth0-01 table 255 proto kernel scope link src 5.5.5.10
  OSR default via 172.16.6.4 dev eth0-Mnt4 proto gated
  OSR local 172.16.6.151 dev lo table 255 proto kernel scope host src 127.0.0.1
  OSR local 172.16.6.151 dev lo table 255 proto kernel scope host src 127.0.0.1
  OSR local 15.15.15.0/24 dev eth0-01 proto kernel scope link src 15.15.15.10
  OSR broadcast 15.15.15.0 dev eth0-01 table 255 proto kernel scope link src 15.15.15.10
  OSR broadcast 15.15.15.0 dev eth0-01 table 255 proto kernel scope link src 15.15.15.10
  OSR 15.15.15.0/16 dev eth0-Mnt4 scope link
  OSR 200.17.122.0/24 via 172.16.6.4 dev eth0-Mnt4 proto gated

Inconsistent OS Routes:
  D  15.15.15.0/24 dev eth0-01 proto kernel scope link src 15.15.15.10
  OSR broadcast 15.15.15.0 dev eth0-01 table 255 proto kernel scope link src 15.15.15.10

Note: Output can be found under:
/var/log/asg_route/Files/Inconsistent.1332465224.tar.gz
```
Showing SGM Information (asg_blade_stats)

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


<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>-a</td>
<td>Aggregate statistics from all SGMs</td>
</tr>
<tr>
<td>corr -p [-v]</td>
<td>Display correction layer statistics per service (for predefined services) for each SGM.</td>
</tr>
<tr>
<td>-p</td>
<td>Display advances statistics</td>
</tr>
<tr>
<td>-v</td>
<td>-a - Aggregate statistics from all SGMs</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>-v</td>
<td>-v – Breakdown to PPAK and FW-1 forwarded packets</td>
</tr>
<tr>
<td>6in4 [-v]</td>
<td>Display statistics on 6in4 forwarded packets</td>
</tr>
<tr>
<td>-v</td>
<td>-v – Breakdown to PPAK and FW-1 forwarded packets</td>
</tr>
<tr>
<td>gre [-v]</td>
<td>Display statistics on GRE forwarded packets</td>
</tr>
<tr>
<td>-v</td>
<td>-v – Breakdown to PPAK and FW-1 forwarded packets</td>
</tr>
<tr>
<td>icmp_error [-v]</td>
<td>Display statistics on ICMP ERROR forwarded packets</td>
</tr>
<tr>
<td>-v</td>
<td>-v – Breakdown to PPAK and FW-1 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>

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.
Showing Traffic Information (asg_ifconfig)

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>
<tr>
<td>-b SGMs</td>
<td>SGM values in one of these formats:</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 (the default option)</td>
</tr>
<tr>
<td></td>
<td>• chassis1</td>
</tr>
<tr>
<td></td>
<td>• chassis2</td>
</tr>
<tr>
<td></td>
<td>• chassis_active</td>
</tr>
<tr>
<td>-d delay</td>
<td>Delay between data samples - default: 5 seconds</td>
</tr>
<tr>
<td>-v</td>
<td>Verbose mode: show detailed information for each interface</td>
</tr>
<tr>
<td>-a</td>
<td>Show absolute values (default: rate values)</td>
</tr>
<tr>
<td>-h</td>
<td>Show help information and exit</td>
</tr>
</tbody>
</table>

`analyze`
- Shows accumulated traffic information.
- Add [-v] [-a] [-d delay] parameters to show traffic distribution between SGMs.

`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:

```
SGM ID     RX packets RX bytes RX dropped
1_03       70%         
1_02       20%         
1_01       10%         
```

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

Syntax  
asg_ifconfig [-b] [SGMs] [interface]

Example  
asg_ifconfig -b chassis1 eth2-01

Output  

```
1_01: Link encap:Ethernet  HWaddr 00:1C:7F:82:01:16
inet6 addr: fe80::2aa:aaff:fe01:201/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
 RX packets:6475 errors:0 dropped:0 overruns:0 frame:0
 collisions:0 txqueuelen:0
 RX bytes:33570 (345.2 KiB) TX bytes:802 (802.0 b)
```

```
1_02: Link encap:Ethernet  HWaddr 00:1C:7F:82:01:16
inet6 addr: fe80::2aa:aaff:fe01:201/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
 RX packets:5941 errors:0 dropped:0 overruns:0 frame:0
 TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:305710 (298.5 KiB) TX bytes:550 (550.0 b)
```

```
1_03: Link encap:Ethernet  HWaddr 00:1C:7F:82:01:16
inet6 addr: fe80::2aa:aaff:fe01:201/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
 RX packets:5786 errors:0 dropped:0 overruns:0 frame:0
 TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:302240 (295.1 KiB) TX bytes:550 (550.0 b)
```

```
1_04: Link encap:Ethernet  HWaddr 00:1C:7F:82:01:16
inet6 addr: fe80::2aa:aaff:fe01:201/64 Scope:Link
 UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
 RX packets:158 errors:0 dropped:0 overruns:0 frame:0
 TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:0
 RX bytes:99242 (98.2 KiB) TX bytes:6934 (6.7 Kib)
```

Comments

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

```
$ asg_ifconfig eth1 analyze -v
```

Output

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

eth1-01
  Link encap:Ethernet  HWaddr 00:3C:7F:18:01:BD
  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: 103 packets:1 bytes:851 (851 bps)  dropped:0
  TX: 102 packets:0 bytes:0 (0 bps)  dropped:0
```

Example 3

```
$ asg_ifconfig eth1-01 analyze -v -a
```

Output

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

eth1-01
  Link encap:Ethernet  HWaddr 00:3C:7F:18:01:BD
  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: 103 packets:1 bytes:851 (851 bps)  dropped:0
  TX: 102 packets:0 bytes:0 (0 bps)  dropped:0
```

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.
**Output**

```
command is executed on SGMs: chassis_active

1_01:  
etht01.  Link encap:Ethernet  Hwaddr 00:1C:7F:81:01:16
       inet6 addr: FE80::21c:7FFFF:FE81:116/64 Scope:Link
       UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500 Metric:1
       RX: packets:6272 bytes:314393 (316.8 Kbytes)  dropped:0
       TX: packets:351 bytes:41550 (40.7 Kbytes)  dropped:0

1_02:  
etht02.  Link encap:Ethernet  Hwaddr 00:1C:7F:81:01:16
       inet6 addr: FE80::21c:7FFFF:FE81:116/64 Scope:Link
       UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500 Metric:1
       RX: packets:6365 bytes:326403 (318.3 Kbytes)  dropped:0
       TX: packets:334 bytes:41028 (40.1 Kbytes)  dropped:0

1_03:  
etht03.  Link encap:Ethernet  Hwaddr 00:1C:7F:81:01:16
       inet6 addr: FE80::21c:7FFFF:FE81:116/64 Scope:Link
       UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500 Metric:1
       RX: packets:5997 bytes:298191 (285.3 Kbytes)  dropped:0
       TX: packets:335 bytes:41004 (40.0 Kbytes)  dropped:0

1_04:  
etht04.  Link encap:Ethernet  Hwaddr 00:1C:7F:81:01:16
       inet6 addr: FE80::21c:7FFFF:FE81:116/64 Scope:Link
       UP BROADCAST RUNNING SLAVE MULTICAST  MTU:1500 Metric:1
       RX: packets:5997 bytes:306413 (296.2 Kbytes)  dropped:0
       TX: packets:335 bytes:61454 (60.6 Kbytes)  dropped:0

```

**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][-rb][-rp][-rd][-tp][-tb][-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
```
Output

> asg_ifconfig banalyze
Processing system statistics for 5 seconds...

1_01:
RX: packets:501 bytes:363301 (354.8 kbps) dropped:0
TX: packets:136 bytes:163020 (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

Example 2

asg_ifconfig -b 1_01,1_02 eth1_mgmt banalyze -v

Processing system statistics for 5 seconds...

1_01:
eth1_mgmt1 Link encap:Ethernet  HWaddr 00:0B:AB:59:33:8A
inet addr:172.23.9.67 Bcast:172.23.9.255 Mask:255.255.255.0
inet6 addr: fe80::20b:abff:fe59:338a/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:27 bytes:16758 (16.4 kbps) dropped:0
TX packets:0 bytes:1172 (1.1 kbps) dropped:0

== Accumulative ==
RX: packets:27 bytes:16758 (16.4 kbps) dropped:0
TX: packets:0 bytes:1172 (1.1 kbps) 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>eth1_mgmt1</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

1_02:
eth1_mgmt1 Link encap:Ethernet  HWaddr 00:0B:AB:59:32:10B
inet addr:172.23.9.67 Bcast:172.23.9.255 Mask:255.255.255.0
inet6 addr: fe80::20b:abff:fe59:3210B/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:26 bytes:15574 (15.2 kbps) dropped:0
TX packets:0 bytes:0 (0 bps) dropped:0

== Accumulative ==
RX: packets:26 bytes:15574 (15.2 kbps) dropped:0
TX: packets:0 bytes:0 (0 bps) 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>eth1_mgmt1</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

== All Blades ==
RX: packets:53 bytes:32332 (31.6 kbps) dropped:0
TX: packets:0 bytes:1172 (1.1 kbps) dropped:0

== Traffic Distribution == (All Blades)

<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>eth1_mgmt1</td>
<td>100.0%</td>
<td>100.0%</td>
<td>0.0%</td>
<td>0.0%</td>
<td>100.0%</td>
<td>0.0%</td>
</tr>
</tbody>
</table>

Comments

Shows detailed and accumulated statistics (rates) for the Mgmt 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

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

Output

```
> asg_ifconfig -b 1_02-1_04 eth2-01 banalyze -v -a
1_02:
etcap:ethernet  Hwaddr 00:1c:7f:82:01:16
inet6 addr: fe80::2c5:aff:fe01:201/64 Scope:link
UP BROADCAST RUNNING MULTICAST MTU:1500  Metric:1
RX: packets:6381 bytes:333854 (326.0 KibiB) dropped:0
TX: packets:7 bytes:550 (350.0 b) dropped:0

== Accumulative ==
RX: packets:6381 bytes:333854 (326.0 KibiB) 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:
etcap:ethernet  Hwaddr 00:1c:7f:82:01:16
inet6 addr: fe80::2c5:aff:fe01:201/64 Scope:link
UP BROADCAST RUNNING MULTICAST MTU:1500  Metric:1
RX: packets:6165 bytes:323756 (316.2 KibiB) dropped:0
TX: packets:174 bytes:7564 (7.4 KibiB) dropped:0

== Accumulative ==
RX: packets:6165 bytes:323756 (316.2 KibiB) dropped:0
TX: packets:174 bytes:7564 (7.4 KibiB) 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>

```

== All Blades ==
RX: packets:18872 bytes:987004 (964.8 KibiB) dropped:0
TX: packets:188 bytes:8664 (8.5 KibiB) dropped:0

== Traffic Distribution == (all blades)
```

<table>
<thead>
<tr>
<th>Interface</th>
<th>RX packets</th>
<th>RX dropped</th>
<th>TX packets</th>
<th>TX dropped</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth2-01</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</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 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>
</table>
| SSM ID   | SSM ID: 1 or 2
          | Shows the total traffic statistics for a specified SSM |
| Interface name | The interface name: eth1-04 or eth1-Sync
          | Shows the total traffic statistics for a specified SSM |
| delay    | Time in seconds (optional, default equals 5). Traffic statistics are divided by the delay interval to show the average per second. |

Example1

`asg_traffic_stats eth1-04`

Output

```
eth1-04 statistics
---------------------
Incoming traffic:
---------------------
Throughput: 273.8 kbps
Packet rate: [total: 1099 pps], [unicast: 328 pps], [multicast: 547 pps], [broadcast: 224 pps]
```

Example2

`asg_traffic_stats 1`

Comments

Shows traffic passing through eth1-04.
### 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]
```

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

```
asg perf -v
```

Output

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>asg perf</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 perf -v</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.</td>
</tr>
</tbody>
</table>

Example 1

```
asg search <source IP> <Destination IP>
```
Output


Legend:
A - Active SGM
B - Backup SGM

Comments
Searching for connections from 14.14.1 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.

---

## 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
```

### 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 1. Full Configuration Wizard

#### Description repeat header rows

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:
   - **General Alerts**:
     - SGM State
     - Chassis State
     - Port State
     - Pingable Hosts State
- System Monitor Daemon
- Hardware Monitor events
  - Fans
  - SSM
  - CMM
  - Power Supplies
  - CPU Temperature
- Performance events
  - Concurrent connections
  - Connection rate
  - Throughput
  - CPU Load
  - Hard Drive Utilization
  - Memory Utilization

4. Alert mode - Switch between enable/disable/monitor modes.
   - Enable or disable the alert. You can also configure the alert in monitor-only mode. Monitor-only events are written to a log file instead of being sent.

2. Edit configuration
   Change the configuration of an alert

3. Show Configuration
   Show the current configuration of an alert

4. Run Test
   Run a test on an alert, to make sure that it works properly

Redirecting Alerts Messages to External syslog server (asg_syslog)

Description:
Whenever an alert message is logged (i.e. sent to SmartView Tracker), it is also sent to syslog.

asg_syslog command should be used in order to redirect these messages to an external syslog server.

This command allows configuring the external syslog server either by IPv4 address or by hostname. It also has an option to verify all SGMs have the same syslog configuration, for alerts purposes.

Command is only available from Expert shell.

Syntax:

```
asg_syslog config ipv4 <syslog server IPv4 address> - configure syslog server by IPv4 address
asg_syslog config host <syslog server hostname> - configure syslog server by hostname. This functionality will be applicable when hostname resolution can be made, either via DNS or by static configuration.
asg_syslog verify – verify that the same syslog server is defined on all SGMs.
```

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

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 (pnote) 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:

```bash
> 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 sgms: 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
```

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  
`asg log [-b <SGMs>] <log_name> [-tail [number]] [-f filter]`
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-b &lt;SGMs&gt;</td>
<td>A list of SGMs to show logs for. List the SGMs in one of these formats:</td>
</tr>
<tr>
<td></td>
<td>- 1_1,1_4</td>
</tr>
<tr>
<td></td>
<td>The first and fourth SGMs on chassis 1</td>
</tr>
<tr>
<td></td>
<td>- 1_1-1_4</td>
</tr>
<tr>
<td></td>
<td>SGMs 1-4 on chassis 1</td>
</tr>
<tr>
<td></td>
<td>- 1_1,1_3-1_7,2_08</td>
</tr>
<tr>
<td></td>
<td>First SGM on chassis1, SGMs 3-7 on chassis 1, SGM 8 on chassis 2.</td>
</tr>
<tr>
<td></td>
<td>- all</td>
</tr>
<tr>
<td></td>
<td>All SGMs</td>
</tr>
<tr>
<td></td>
<td>- chassis1</td>
</tr>
<tr>
<td></td>
<td>All SGMs on chassis1</td>
</tr>
<tr>
<td></td>
<td>- chassis2</td>
</tr>
<tr>
<td></td>
<td>All SGM on chassis2</td>
</tr>
<tr>
<td></td>
<td>- chassis_active</td>
</tr>
<tr>
<td></td>
<td>All SGMs on the active chassis</td>
</tr>
<tr>
<td>log_name</td>
<td>Enter one of these log types:</td>
</tr>
<tr>
<td></td>
<td>- audit</td>
</tr>
<tr>
<td></td>
<td>Shows the audit logs in /var/log, for example:</td>
</tr>
<tr>
<td></td>
<td>/var/log/asgaudit.log.1</td>
</tr>
<tr>
<td></td>
<td>- smd</td>
</tr>
<tr>
<td></td>
<td>Shows the System Monitor Daemon logs in /var/log, for example:</td>
</tr>
<tr>
<td></td>
<td>/var/log/sdm.log.2</td>
</tr>
<tr>
<td>-tail [number]</td>
<td>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 &quot;-tail 3&quot; limits the output to the last three lines of the log file.</td>
</tr>
<tr>
<td>-f filter</td>
<td>Word or phrase to filter by. For example: -f debug</td>
</tr>
</tbody>
</table>

Example 1:

```
* 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
```
Example 2:

```
# asg log smd -tail 3
Aug 03 09:07:02 1_03 check_dx1_consistency_with_of OK
Aug 03 09:08:05 2_02 check_dx1_consistency_with_of OK
Aug 03 09:10:03 1_03 check_dx1_consistency_with_of OK
Aug 03 09:11:03 2_02 check_dx1_consistency_with_of OK
Aug 03 09:13:03 1_03 check_dx1_consistency_with_of OK
Aug 03 09:14:03 2_02 check_dx1_consistency_with_of OK
Aug 03 09:14:09 1_03 check_date_time_configuration ok
Aug 03 09:14:09 1_01 check_dx1_configuration_consistency ok
Aug 03 09:14:09 1_01 check_debug_flags ok
Aug 03 09:15:00 2_01 check_dx1_configuration_consistency ok
Aug 03 09:15:00 2_01 check_debug_flags ok
Aug 03 09:15:00 1_02 check_dx1_configuration_consistency ok
Aug 03 09:15:29 1_02 check_debug_flags ok
Aug 03 09:15:29 1_02 check_dx1_configuration_consistency ok
Aug 03 09:15:29 1_02 check_date_time_configuration ok
Aug 03 09:15:37 2_03 check_date_time_configuration ok
Aug 03 09:15:38 2_03 check_debug_flags ok
Aug 03 09:15:38 2_03 check_dx1_configuration_consistency ok
```

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>] [-tail <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>-tail &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
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 31).
- 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>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 distribution modes: User, Network, General, and per port. The distribution mode affects the way an SSM distributes traffic among the SGMs.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>set_dist_mask</td>
<td>Sets the number of bits to be considered when calculating distribution. The 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 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.
System Under Load

Description

System Under Load feature (SUL) enables the GW to monitor high CPU load and also suspends setting remote SGMs to DOWN state when cannot receive CCP packets for a timeout of \textit{BLADE\_DEAD\_INTERVAL} (default is 3 sec) and when SUL state ON.

It enables every SGM to act differently when they/other 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 (\textit{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 (\textit{Number of sample}) – sample is taken every 2HTUs.

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
System Under Load

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 fw ha_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 fw ha_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-3: Cluster simpl is added
cluster_info [Bladed System] FW-1: [SUL] Changing SUL state to ON due to high CPU usage [15%] on remote Blade 1, threshold is 15% local CPU usage is 15% fw_message [1,1]
cluster_info [Bladed System] FW-1: [SUL] Changing SUL state to OFF, system is still underload but SUL timeout expired (35 seconds), fw_message [1,1]
cluster_info [Bladed System] FW-3: [SUL] Changing SUL state to OFF due to high CPU usage [15%] on remote Blade 1, threshold is 15% local CPU usage is 15% fw_message [1,1]
cluster_info [Bladed System] FW-1: [SUL] Changing SUL state to ON due to high CPU usage [15%] on remote Blade 1, threshold is 15% local CPU usage is 15% fw_message [1,1]
```

cluster_info [Bladed System] FW-1: [SUL] Notification completed on all Blades, fw_message [1,1]

**Tuning feature Parameters**

SUL feature can be modified and tuned to meet user specific needs.

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

---

**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.
Showing the firewall Database Configuration (asg config)

### Syntax

`asg config show` or
`asg config 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 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: asg conns 1_1, 1_3. When this parameter is not specified, only connections on the active chassis are shown.</td>
</tr>
</tbody>
</table>

Example
asg conns

Output
1_01: #VALS 385 #PEAK 1004 #SLINKS 385
1_02: #VALS 53 #PEAK 227 #SLINKS 57
1_03: #VALS 53 #PEAK 230 #SLINKS 57
1_04: #VALS 281 #PEAK 981 #SLINKS 285
1_05: #VALS 54 #PEAK 198 #SLINKS 58
1_06: #VALS 53 #PEAK 147 #SLINKS 57

Total (fwL connections table): 879 connections

1_01: There are 180666 connections in SecureXL connections table 1_02: There are 189256 connections in SecureXL connections table 1_03: There are 160212 connections in SecureXL connections table 1_04: There are 148598 connections in SecureXL connections table 1_05: There are 162430 connections in SecureXL connections table 1_06: There are 124090 connections in SecureXL connections table

Total (SecureXL connections table): 952272 connections
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 Use a comma to separate specified SGMs, for example:</td>
</tr>
<tr>
<td></td>
<td>asg_version -b 1_01</td>
</tr>
<tr>
<td>verify [-v]</td>
<td>Makes sure system hardware components are running approved software and</td>
</tr>
<tr>
<td></td>
<td>firmware versions Use -v for verbose mode Verbose mode also shows</td>
</tr>
<tr>
<td></td>
<td>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
```
+----------------+-----------------+----------+
| Component       | Configuration   | Firmware |
| Chassis 1       |                 |          |
| SSM1            | F3_v7.5_20_CP.cfg | 7.5.20   |
| SSM2            | F3_v7.5_20_CP.cfg | 7.5.20   |
| CMM             | N/A             | 2.70     |
| Chassis 2       |                 |          |
| SSM1            | F3_v7.5_20_CP.cfg | 7.5.20   |
| SSM2            | F3_v7.5_20_CP.cfg | 7.5.20   |
| CMM             | N/A             | 2.70     |
```

- The first part of the output shows software and firmware versions installed on SSMs and CMMs.
- The second part shows:
  - tp_cp> software versions installed on SGMs
  - Internal firmware versions, such as BIOS.
  - CPU related information, such as frequency and CoreXL allocation
Example 2 asg_version verify

Output

> asg_version verify
Chassis1
---
SSM1 Firmware version: database and hardware do not match:
database - 7.5.20, hardware - 7.5.19

blades
-----
05 version
--------
-=` 8 blades: 1.01 1.02 1.03 1.04 2.01 2.02 2.03 2.04 -*
05 build 18, 05 kernel version 2.6.18-92Epx80_04, 65 edition 64-bit
Firewall-1 version
-------------------
-=` 8 blades: 1.01 1.02 1.03 1.04 2.01 2.02 2.03 2.04 -*
This is Check Point VPN-1(TM) & Firewall-1(R) R75 - Build 017
kernel: R75 - Build 017

Performance Pack version
--------------------------
-=` 8 blades: 1.01 1.02 1.03 1.04 2.01 2.02 2.03 2.04 -*
This is Check Point Performance Pack version: R75 - Build 004
kernel version: R75 - Build 004

Hardware
--------
-=` 8 blades: 1.01 1.02 1.03 1.04 2.01 2.02 2.03 2.04 -*
IPMC: 1.42 BL: 1.42 FPGA: 2.4 FPGAR: 2.3B BIOS: 1.70
SSD
---
-=` 8 blades: 1.01 1.02 1.03 1.04 2.01 2.02 2.03 2.04 -*
Firmware Version: 4PCI0302

Number of cores
---------------
-=` 8 blades: 1.01 1.02 1.03 1.04 2.01 2.02 2.03 2.04 -*
12

Number of CoreX1 instances
---------------------------
-=` 8 blades: 1.01 1.02 1.03 1.04 2.01 2.02 2.03 2.04 -*
8

CPUs frequency
-------------
-=` 8 blades: 1.01 1.02 1.03 1.04 2.01 2.02 2.03 2.04 -*
2.4GHz

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.
**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 (-) an entries from the SGM database.

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

**Example**

```
Nov 22 08:42:30 admin localhost  t +  derived:hostname copmodule-ch1 01 [1 Blades: 1.01]
Nov 22 08:42:30 admin localhost  t -  hosts:4:gw-s11da7:address [1 Blades: 1.01]
Nov 22 08:42:31 admin localhost  p +  machine:hostname copmodule-ch1 01 [1 Blades: 1.01]
Nov 22 08:42:31 admin localhost  p -  hosts:4:gw-s11da7:1 [1 Blades: 1.01]
```

**Syntax:**

```
asg_auditlog [-b blades] [-d <n>] [-tail <n>] [filter <filter>]
```
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-b &lt;blades&gt;</code></td>
<td>Specifies SGMs</td>
</tr>
</tbody>
</table>
| `-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_auditlog
```

```
jul 11 12:09:06 admin localhost t +chassis.high-availability:factor:sensor:fan 1 [9 Blades: 1.01,1.02,1.03,1.04,1.05,2.02,2.03,2.04,2.05] 1
jul 11 12:09:09 admin localhost p -chassis.high-availability:factor:sensor:fan 11 [9 Blades: 1.01,1.02,1.03,1.04,1.05,2.02,2.03,2.04,2.05] 2
jul 11 12:09:10 admin localhost p -chassis.high-availability:factor:sensor:fan 1 [9 Blades: 1.01,1.02,1.03,1.04,1.05,2.02,2.03,2.04,2.05] 3
```

**Comments**

The output shows that:

- 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_auditlog -b 1.3-1.4 -d 50 -f cpu_load
```

```
jul 10 10:21:37 admin localhost t +chassis:alert_threshold:cpu_load_threshold_perc_low_ratio 80 [2 Blades: 1.03,1.04]
jul 10 10:21:37 admin localhost t +chassis:alert_threshold:cpu_load_threshold_perc_high 75 [2 Blades: 1.03,1.04]
jul 10 10:21:38 admin localhost p +chassis:alert_threshold:cpu_load_threshold_perc_high 75 [2 Blades: 1.03,1.04]
```

**Comments**

This output shows that:

- 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
**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.

Gclish 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:

gdual7-543-c02-02 > 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>-----------------------</td>
<td>-------</td>
<td>-----------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Name</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------</td>
<td>-----------------------</td>
<td>-------</td>
</tr>
</tbody>
</table>

Accelerated Path

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

Accelerated VPN Path

| C crypt conns        | 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 conns             | 0      |
| C PXL conns          | 0     | C PXL templates       | 0      |

Firewall Path

| F2F packets          | 10077862 | F2F bytes          | 1185051123 |
| F2F conns            | 38839    | C F2F conns        | 800       |
| TCP violations        | 0        | C partial conns     | 0         |
| C anticipated conns   | 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-s43-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
```
```
  gdual7-s43-ch02-02 > fw ctl iflist
  -* 6 blades: 1_01 1_02 1_03 2_01 2_02 2_03 -*
  0 : BPEth0
  1 : BPEth1
  2 : eth1-Mgmt4
  3 : eth2-Mgmt4
  4 : eth1-01
  5 : eth1-CIN
  6 : eth2-CIN
  8 : eth2-01
  16 : Sync
  17 : eth1-Mgmt1
  18 : eth2-Mgmt1
```

**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”.

Global command syntax:
<global command> [global command-flags] [native command arguments]

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 arguments] 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:

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

```
> md5sum /opt/CPsuite-R75/fw1/modules/fwkern.conf
-
0a3a446b638331e7815f49fdc794f9b7 /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:

```
gdual7-t43-ch02-02 > tcpdump -nnni bond1
[1_01]tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
[1_02]tcpdump: listening on bond1, link-type EN10MB (Ethernet), capture size 96 bytes
[1_03]tcpdump: listening on bond1, link-type EN10MB (Ethernet), capture size 96 bytes
[1_02]UDP  packetinya, ethertype Unknown (0x88cc), length 56:
[2_02]  0x0000:  0207 045c 260a 9f07 1f04 0705 312f 302f  ...
```

---

---
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:

a. asg_reboot <global command-flags> – power off and on SGMs
b. asg_hard_shutdown <global command-flags> – power off SGMs
c. 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.
General global commands

Description:
The global commands are utilities to run certain commands on multiple SGMs. This document is dealing with general purpose utilities,

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

The list of available commands is: update_conf_file, global, asg_cp2blades, asg_clear_table, asg_clear_messages, asg_blade_stats

Those commands are available in the gclish in addition they are available in bash:

<table>
<thead>
<tr>
<th>gclish name</th>
<th>bash name</th>
</tr>
</thead>
<tbody>
<tr>
<td>update_conf_file</td>
<td>g_update_conf_file</td>
</tr>
<tr>
<td>global</td>
<td>global_help</td>
</tr>
<tr>
<td>asg_cp2blades</td>
<td>asg_cp2blades</td>
</tr>
<tr>
<td>asg_clear_table</td>
<td>asg_clear_table</td>
</tr>
<tr>
<td>asg_clear_messages</td>
<td>asg_clear_messages</td>
</tr>
<tr>
<td>asg_blade_stats</td>
<td>asg_blade_stats</td>
</tr>
</tbody>
</table>

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

**update_conf_file:**

Usage: update_conf_file <file_name> <var>=<value>

Description: update_conf_file is a utility to add, update and remove variables from configuration files

(configuration file format is specified below)

Input parameters:
- file-name: Name/Path of .conf file to update. In case of known conf files full path is not required known conf files are: fwkern.conf, simkern.conf
- var: Variable name
- value: New value. An empty value will remove the variable from the .conf file (yet “=” sign must be specified)

Example:

```
> cat /home/admin/MyConfFile.txt
-- 3 blades: 2_01 2_02 2_03 --
cat: /home/admin/MyConfFile.txt: No such file or directory

> update_conf_file /home/admin/MyConfFile.txt var1=hello
> cat /home/admin/MyConfFile.txt
-- 3 blades: 2_01 2_02 2_03 --
var1=hello

> update_conf_file /home/admin/MyConfFile.txt var2=24h
> cat /home/admin/MyConfFile.txt
-- 3 blades: 2_01 2_02 2_03 --
var2=24h
var1=hello

> update_conf_file /home/admin/MyConfFile.txt var1=goodbye
> cat /home/admin/MyConfFile.txt
-- 3 blades: 2_01 2_02 2_03 --
var2=24h
var1=goodbye

> update_conf_file /home/admin/MyConfFile.txt var2=
> cat /home/admin/MyConfFile.txt
-- 3 blades: 2_01 2_02 2_03 --
var1=goodbye
```


Configuration file required format:
The configuration file is composed of lines of variable initialization where each line defines one variable.
Line format is: <variable>=<value>
Variable name must not include "=" sign
Note: fwkern.conf and simkern.conf are aligned with this definition

global help
Usage: global help
Description: shows the list of global commands accessible through gclish and their general usage
Example:
  gcpmodule-ch02-01 > global help
  Usage: <command_name> [-b SGMs] [-a -l -r] <native command arguments>
  Executes the given command on specified blades.

Optional Arguments:
- b  blades: in one of the following formats
      1_1,1_4 or 1_1-1_4 or 1_01,1_03-1_08,1_10
      all (default)
      chassis1
      chassis2
      chassis_active
- a  : Force execution on all blades (incl. down blades).
- l  : Execute only on local blade.
- r  : Execute only on remote blades.

Command list:
  arp cat cp cpconfig cplic cpstart cpstop dmesg ethtool fw fw6 fwaccel fwaccel6 fwaccel6_m
  fwaccel_m ls md5sum mv netstat reboot sim sim6 snapshot_recover snapshot_show_current tail
tcpdump top unlock update_conf_file vpn asg

asg_cp2blades
usage: asg_cp2blades [global command -flags] [-s] file-name-full-path [destination-full-path]
Description: this utility copies files from the current SGM to any specified SGMs
Input parameters:
Global command flags – the global flags which specify on which SGMs to be applied on
-s - flag that specify whether or not to save a local copy of the old file on each of the selected SGMs. The
saved copy will reside on the same directory as the original file and will end with .bak.<date>.<time>
file-name-full-path – full path to the file to be copied. If full-path is not specified the file will be searched in
current directory.
destination-full-path – full path to a destination location for the file. If destination was not specified, the file will
be copied to the source file location
example:
  gcpmodule-ch02-01 > cat /home/admin/note.txt
  "-" 1 blade: 2_01 "-"
  hello world
  "-" 2 blades: 2_02 2_03 "-"
  cat: /home/admin/note.txt: No such file or directory

  gcpmodule-ch02-01 > asg_cp2blades /home/admin/note.txt
  Operation completed successfully

  gcpmodule-ch02-01 > cat /home/admin/note.txt
  "-" 3 blades: 2_01 2_02 2_03 "-"
  hello world

asg_clear_table
usage: asg_clear_table [global command-flags]
Description: clears firewall connection table. This function will delete connections from fw connection table.
Its success indication is having less than 50 connections; it will repeat delete process for up to 15 times until
meeting this threshold.
Note: if connected to the machine by SSH, this command will delete current connection and user will need to
re-establish the connection
asg_clear_messages
usage: asg_clear_messages [global command-flags]
Description: clears all messages in /var/log/messages files
Example:

gcpmodule-ch02-01 > asg_clear_messages
This action will erase the messages in /var/log/messages
and will be executed on blades: all
Are you sure? (Y - yes, any other key - no) y
Command completed successfully

Examples:

> show interface eth1-01 ipv4-address
1_01:
ipv4-address 4.4.4.10/24

1_02:
ipv4-address 4.4.4.10/24

1_03:
ipv4-address 4.4.4.10/24

1_04:
ipv4-address 4.4.4.10/24

1_05:
Blade 1_05 is down. See "/var/log/messages".

2_01:
ipv4-address 4.4.4.10/24

2_02:
ipv4-address 4.4.4.10/24

2_03:
ipv4-address 4.4.4.10/24

2_04:
ipv4-address 4.4.4.10/24

2_05:
ipv4-address 4.4.4.10/24

Tcpdump - multi-blade capture (tcpdump –mcap)

Description:
Two new command line options were added to tcpdump:

1. tcpdump –mcap - supports capturing of packets from multiple blades and saving them into a single capture file.
2. tcpdump –view – reads packets from the file saved by tcpdump -mcap and displays the id of the blade on which the packet was captured

Syntax:

1. tcpdump –mcap
Arguments:
tcpdump \[-b\ blade string\] -mcap -w 'full path to capture file' [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 -nni eth1-Mgmt4
```
Capturing packets...

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

```
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-rw---- 1 admin root 46285 Nov 27 14:12 /tmp/capture
- rw-r--r-- 1 admin root 9500 Nov 27 14:12 /tmp/capture_1_1
- rw-r--r-- 1 admin root 6996 Nov 27 14:12 /tmp/capture_1_2
- rw-r--r-- 1 admin root 7541 Nov 27 14:12 /tmp/capture_1_3
- rw-r--r-- 1 admin root 7541 Nov 27 14:12 /tmp/capture_2_1
- rw-r--r-- 1 admin root 7286 Nov 27 14:12 /tmp/capture_2_2
- rw-r--r-- 1 admin root 7541 Nov 27 14:12 /tmp/capture_2_3

```
[Expert@cpmodule-ch01-01]#
```

Example:
Simple usage:
tcpdump -mcap -w /tmp/capture
On selected blades:
tcpdump \-b \_1\_1,\_3\_2\_1\ -mcap \-w \ /tmp/capture \-nni eth1-Mgmt4
On specific interface:
tcpdump -mcap -w /tmp/capture -nni eth1-Mgmt4
With filter:
tcpdump -mcap -nni eth1-Mgmt4 -w /tmp/capture proto http

2. tcpdump -view

Arguments:
tcpdump -view -r 'full path to capture file' [tcpdump cmdline]

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

```
gcpmodule-ch01-01 > tcpdump -view -r /tmp/capture
```
reading from file /tmp/capture, link-type EN10MB (Ethernet)
[1_3] 14:11:57.971587 IP 0.0.0.0.cp-cluster > 172.16.6.0.cp-cluster: UDP, length 45
[2_3] 14:12:07.625171 IP 0.0.0.0.cp-cluster > 172.16.6.0.cp-cluster: UDP, length 45
[2_3] 14:12:09.974195 IP 0.0.0.0.cp-cluster > 172.16.6.0.cp-cluster: UDP, length 37
[2_1] 14:12:09.989745 IP 0.0.0.0.cp-cluster > 172.16.6.0.cp-cluster: UDP, length 45
[2_3] 14:12:10.022995 IP 0.0.0.0.cp-cluster > 172.23.9.0.cp-cluster: UDP, length 32
...

Example:
tcpdump -view -r /tmp/capture proto 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 to display the captured packets. The packets should be prefixed with the blade id of the blade on which the packet was captured.

**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

```
asg_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 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
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 Diagnostics (asg diag)

Description:

This command displays system diagnostics. Upon execution, the command runs over predefined list of diagnostics utilities from different areas: installation, networking, routing, distribution, security enforcement and more.

asg_diag can be executed in two ways:

**asg diag list**

Display the predefined list of diagnostics utilities.

**asg_info**

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 cpminfo 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 SEL 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
asg diag verify

This functionality is divided to three stages:

1. Run the predefined diagnostics utilities and display it output on the screen
2. Display summary of the execution: for each command indicate whether it passed/failed/unable to determine (i.e. raw output should be examined)
3. Write the output of the previous two sections into a file, which has the following format: verifier_sum.<timestamp>.txt. File is located under /var/log/ directory
Example:
This example displays the last two stages of the verification:

```
> asg diag verify
.
.
.
```

==============================
Summary:
==============================

- Bond Verifier - Check raw output
- Bond Verifier Verbose - Check raw output
- Cores Data - Check raw output
- DXL Verifier - OK
- Distribution Mode Verifier - OK
- Dynamic Routing Verifier - Check raw output
- Dynamic Routing Verifier All - Check raw output
- General Status - Check raw output
- Hardware Status - Check raw output
- Installation Verifier - OK
- Local ARP Verifier - OK
- Local ARP Verifier Verbose - OK
- MAC Verifier - Fail
- MAC Verifier Verbose - Fail
- Policy Verifier - Check raw output
- Resource Status - Check raw output
- Security Group Verifier - Check raw output
- Syslog Verifier - OK
- Version Verifier - Check raw output

File: verifier_sum.1331661784.txt is located at: /var/log

**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 : AKO0789153
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

61000 Security Systems Configuration

Configuring Security Gateway Modules as Up or Down (asg_blade_admin)

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

Syntax
asg_blade_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>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>

- **p**
Persistent. The setting is kept after reboot.

- **h**
Display usage

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

### Output

You are about to perform blade_admin up on blades:

```
2_03
```

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

Blade_admin up requires auditing
Enter your full name: Fred
Enter reason for blade_admin up [Maintenance]: test
WARNING: Blade_admin up on blades: 2_03, User: Fred, Reason: test

```
Performing blade_admin up on blades: 2_03
[2_03]Setting blade to normal operation ...
[2_03]pulling configuration from: 192.0.2.16 (may take few seconds)
[2_03]Blade current state is ACTIVE
```

### Comments
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.

---

### 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`
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

```
> asg security_group
+--------------------------------------+
| Security Group Utility               |
+--------------------------------------+
Current Security Group:
+--------------------------------------+
<table>
<thead>
<tr>
<th>Chassis</th>
<th>Security Gateway Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,2,3</td>
</tr>
<tr>
<td>2</td>
<td>1,2,3</td>
</tr>
</tbody>
</table>
+--------------------------------------+
Choose one of the following options:
-------------------------------------
1) Add SGMs to Security Group
2) Remove SGMs from Security Group
3) Exit
```
Distribution Modes

Distribution modes refer 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:
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.

**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 96).

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.

**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>
</tbody>
</table>
### Manually Configuring Distribution Modes (asg_dx1 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_dx1 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_dx1 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
     >
     ```

   - 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:

5. 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 96)

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_dxl 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 SSMs.

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 SMM shows in the output.

```
   distribution mode: User/Network
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 <code>eth1-01</code> is set to the <em>User</em> distribution mode.</td>
</tr>
<tr>
<td>eth1-02=0</td>
<td>=0 means <code>eth1-02=0</code> is set to <em>network</em> distribution mode.</td>
</tr>
<tr>
<td>eth1-03=1</td>
<td>=1 means <code>eth1-03</code> is set to the <em>User</em> 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:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth2-01=0</td>
<td>=0 means the <code>eth2-01=0</code> is set to the <em>Network</em> distribution mode.</td>
</tr>
<tr>
<td>eth2-02=0</td>
<td>=0 means the <code>eth2-02=0</code> is set to the <em>Network</em> distribution mode.</td>
</tr>
<tr>
<td>eth2-03=0</td>
<td>=0 means the <code>eth2-03=0</code> is set to the <em>Network</em> distribution mode.</td>
</tr>
</tbody>
</table>

**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*.

---

**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_dxl 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>
</table>
| fwha_ch_nanda_dos_attack_enabled              | - Enables or disables the ND Adv DoS defense mechanisms  
|                                               | - Possible values: 0 - Disable  
|                                               | - 1 - Enable (default)  
|                                               | - 2 - Monitor  
|                                               | In monitor mode, the attack event is logged but no action taken.  
| fwha_ch_nanda_forwarding_threshold            | - Configures the forwarding threshold, the number of ND Adv packets forwarded to other SGMs during the forwarding interval.  
|                                               | - Default value: 5000  
| fwha_ch_nanda_forwarding_interval             | - Configures the forwarding interval (in milliseconds).  
|                                               | - Default value: 10000  |
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 gcli 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. There are two bonding modes available:

- **8023AD (LACP)**
  Do dynamic bonding according to the IEEE 802.3ad protocol

- **XOR**
  Do load sharing based on layer2, or 3 and 4.

**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
Configuring Chassis High Availability

Chassis High Availability section 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.

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>]
```
### Parameter | Description
--- | ---
SGM | - Sets the weight factor for an SGM  
- The weight factor must be between 0 and 1000  
- Example: set chassis high-availability factors sgm 100

port high | - A port has one of two grades: high or standard. This parameter sets a weight factor for the high grade  
- The factor must be between 0 and 1000  
- Example: set chassis high-availability factors Port high 70

This means that ports set to high grade have a weight of 70.

port standard | - A port has one of two grades: high or standard. This parameter sets a weight factor for the standard grade  
- The factor must be between 0 and 1000  
- Example: set chassis high-availability factors Port standard 50

This means that ports set to standard grade have a weight of 50.

Sensor CMMs | - Sets a weight factor for CMMs  
- The factor must be between 0 and 99  
- Example: set chassis high-availability factors sensor cmm 40

Sensor fans | - Sets a weight factor for fan units  
- The factor must be between 0 and 99  
- Example: set chassis high-availability factors sensor fans 30

Sensor Power Supplies | - Sets a weight factor for power supply units  
- The factor must be between 0 and 99  
- Example: set chassis high-availability factors sensor power_supplies 20

SSMs Sensor | - Sets a weight factor for SSMs  
- The factor must be between 0 and 99  
- Example: set chassis high-availability factors sensor ssm 45

pnote pingable_hosts | - Sets a weight factor for pingable hosts, a way of making sure ports are properly connected to their hosts.  
- The factor must be between 0 and 99  
- Example: set chassis high-availability factors pnote pingable_hosts 99

---

### 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>`
## Configuring Chassis High Availability

### Parameter | Description
---|---
0 | No primary Chassis (Active Up Mode)
   
   In this mode, the chassis which is UP stays up until the other chassis gets a higher grade.

1 | Chassis 1 is Primary Chassis

2 | Chassis 2 is Primary Chassis

### 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 -v` after chassis failover, you will be notified with the freeze time:

```
glanit-<CH01-01> \ asg stat -v
```
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.

**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 constantly Chassis fail-over and failback whenever there is an interface link flapping.

When Interface state has changed form down to up, it will be considered in the chassis grade only if the link state is up for X seconds (default is 10 sec).

**Configuration:**

The Link Preemption Mechanism is enabled by default with preemption time of 10 seconds

In order to set a different value, run from gclish:

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

For example, to set the preemption time to 20 seconds, from gclish run:
```
> fw ctl set int fwha_ch_if_preempt_time 20
> update_conf_file fwkern.conf fwha_ch_if_preempt_time=20
```
Deactivation:

In order to disable the feature run from gclish:

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

Verification:

To check what is the preemption time:

```
# fw ctl get int fwha_ch_if_preempt_time
```

## 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's unique IP.

**Note**

- Similar to the SMO mechanism, only one SGM owns the UIPC task
- The UIPC feature is disabled by default

### 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 management interfaces, for example eth1_mgmt1</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:
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
- Advertise those destinations to other routers in the network
- Determine the best path to each network
- Dynamically learn changes in routing topology

### 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>
<tr>
<td>area</td>
<td>Specifies the area ID. The area ID must be one of these:</td>
</tr>
<tr>
<td></td>
<td>An IPv4 address</td>
</tr>
<tr>
<td></td>
<td>A value between 1 and 4294967295</td>
</tr>
<tr>
<td></td>
<td>backbone</td>
</tr>
<tr>
<td></td>
<td>By default, the backbone area is enabled.</td>
</tr>
</tbody>
</table>

**Example**
```
> set ospf interface eth1-01 area backbone on
```
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]
```

<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></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`
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**

```bash
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 135).</td>
</tr>
<tr>
<td>full_sync</td>
<td>Runs full sync from a remote SGM. The <code>&lt;ip_addr&gt;</code> is the Sync IP of the remote SGM. The full Sync process synchronizes kernel tables between SGMs.</td>
</tr>
<tr>
<td>set_sync_start_ip</td>
<td>Changes the Sync start IP address from the local SGM to the specified address.</td>
</tr>
<tr>
<td>reset_uptime</td>
<td>Resets the system uptime on all SGMs to the current time.</td>
</tr>
<tr>
<td>reset_uptime_user</td>
<td>An interactive command that resets the uptime for all SGMs to a user configured time.</td>
</tr>
<tr>
<td>get_smo_ip</td>
<td>Returns the sync IP address of the Single Management Object defined in SmartDashboard. This address is not shown in SmartDashboard.</td>
</tr>
<tr>
<td>is_in_security_group</td>
<td>Checks whether the local SGM is in the security group.</td>
</tr>
<tr>
<td>is_in_pull_conf_group</td>
<td>Check whether the local SGM is in the Pulling Configuration Group (if not, the SGM won’t pull configuration and policy)</td>
</tr>
<tr>
<td>config fetch_smc</td>
<td>Fetches the policy from the Security Management server, and distributes it to all SGMs.</td>
</tr>
</tbody>
</table>

**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.

### Changing the Default VMAC

**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:
Changing the Default VMAC (asg_unique_mac_utility)

- 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**

```
<table>
<thead>
<tr>
<th>Hostname</th>
<th>Unique MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>[opmodule]</td>
<td>[254]</td>
</tr>
</tbody>
</table>
```

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
  ```
  # # 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_magic_magic
  -*- 4 sgms: 1_01 1_02 2_02 2_03 -*- 22
  fwha_magic_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)
```
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.

Syntax

asg log_servers

Example

asg log_servers

Output

Asg log_servers

Log Server 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 Server 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:
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:

- `>fw ctl set int cphwd_udp_selective_delay_ha <delay in seconds>`
- `>fwaccel off`
- `>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`

   ```
   > fwaccel templates
   
   Source  Sport Destination  DPort PR Flags Comm LCT  DLY CI2 i/f SCI i/f Inst Identity
   
   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 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
1_01: Success

Deleting the 6in4 Virtual Interface

Run: delete interface <physical-if> 6in4 <6in4-id>. For example:

> delete interface eth1-01 6in4 999
1_01: 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

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
### Configuring ECMP

#### 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. To reach the destination network defined in the static route, the packets must first go through one of the defined next-hop gateways.

#### 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>

#### 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.
3. On the **Logs and Masters > Log Servers** page, select **Define Log Servers**.
4. Select the dedicated log server.
5. Install a policy.

 Müge-

- 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).
Example

```plaintext
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), O-ISA - OSPF InterArea, O-ASBR, O-AS
       A - Aggregate, K - Kernel, H - Hidden, P - Suppressed
S 0.0.0.0/0 via 172.16.6.4, mgmt, cost 0, age 6137
S 50.50.50.0/24 via 20.20.20.101, eth0-01, cost 0, age 15
    via 20.20.20.102, eth0-01
    via 20.20.20.103, eth0-01
```

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:

1. Open this file for editing:
   ```bash
   $PPKDIR/boot/modules/simkern.conf
   ```
   If `simkern.conf` does not exist, create it.

2. Add this line:
   ```plaintext
   sim_routing_by_source=0
   ```

3. Save the file and reboot.

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`
   
   - 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.

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)
### SMD Mode

<table>
<thead>
<tr>
<th>SMD Mode</th>
<th>Meaning</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 (<em>Configuring Alerts for SGM and Chassis Events (asg alert)</em> on page 32).</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><code>check_blade_down</code></td>
</tr>
<tr>
<td>Blade Admin Down</td>
<td>An SGM’s administrative state set using the <code>asg blade_admin</code> command is DOWN</td>
<td><code>check_blade_admin_down</code></td>
</tr>
<tr>
<td>SecureXL Down</td>
<td>SsecureXL's state set as DOWN</td>
<td><code>check_sxl_down</code></td>
</tr>
<tr>
<td>SecureXL Admin Down</td>
<td>SecureXL's administrative state has been set as down</td>
<td><code>check_sxl_admin_down</code></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></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The files listed in: <code>/etc/xfer_file_list</code> must be identical on both SGMs.</td>
</tr>
<tr>
<td></td>
<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>
</tr>
</tbody>
</table>
**Verification:**

<table>
<thead>
<tr>
<th>Verification:</th>
<th>Tests if:</th>
<th>Identifier in smd_user.conf</th>
</tr>
</thead>
<tbody>
<tr>
<td>DxI Configuration Consistency</td>
<td>The md5 checksum of the dxI configuration on the local SGM is identical to the md5 checksum of the dxI configuration calculated on the SMO SGM.</td>
<td>check_dxI_configuration_consistency_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_dxI_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>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>
<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>

1. Enable all the verification tests or configure each test manually.
2. 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>Meaning</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:**

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

**Note** - To enable or disable SMD:

(i) Run: `asg_system_monitor config`.
(ii) Select **Enable SMD** or **Disable SMD**.

**To enable or disable an SMD verification test:**
1. Run: `asg_system_monitor config`.
   The primary SMD menu opens.
2. Select **Edit Configuration**.
3. Select **Verification Tests**.
   The Verification Tests menu opens.
4. Select one of the verification tests.
5. Enter y or n to enable or disable the verification.

**To show the existing SMD configuration:**
1. Run: `asg_system_monitor config`.
   The primary SMD menu opens.
2. Select **Show Configuration**.
3. 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Contains the results of each test verification and the result of any</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>corrective action.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• When file the size of smd.log becomes larger than 1MB, it is</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>renamed smd.log.1 and a new smd.log opened. Up to 2 log files of</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>this type can be opened.</td>
</tr>
</tbody>
</table>
Verifying Port Connectivity (asg_pingable_hosts)

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

<table>
<thead>
<tr>
<th>Log file</th>
<th>Location</th>
<th>Max size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>smd_smo.log</td>
<td>/var/log/</td>
<td>500MB</td>
<td>• This log is created on the SMO SGM only.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Contains data about all the activities monitored by SMD.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• The log level can be modified using asg_system_monitor config &gt; Edit Configuration &gt; Debug Level. Select error, info or debug (default).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• smd_smo.log is created on all SGMs that have functioned as the SMO SGM.</td>
</tr>
</tbody>
</table>

Log command and options

<table>
<thead>
<tr>
<th>Command</th>
<th>Meaning</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>
<tr>
<td>watch asg log smd -tail 5</td>
<td>Periodically shows the last 5 logs from each SGM.</td>
</tr>
<tr>
<td>cat /var/log/smd_smo.log</td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>To find out when SGM 2_03 was UP run:</td>
</tr>
<tr>
<td></td>
<td>cat /var/log/smd_smo.log</td>
</tr>
<tr>
<td></td>
<td>To confirm if SGM 2_08 has been rebooted by the SMD, run:</td>
</tr>
<tr>
<td></td>
<td>cat /var/log/smd_smo.log</td>
</tr>
</tbody>
</table>

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.
Verifying Port Connectivity (asg_pingable_hosts)

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:</td>
</tr>
<tr>
<td></td>
<td>$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:</td>
</tr>
<tr>
<td></td>
<td>asg_pingable_hosts load_ips</td>
</tr>
<tr>
<td>disable</td>
<td>Disables the pingable hosts 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
>
```
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></th>
</tr>
</thead>
<tbody>
<tr>
<td>Uptime</td>
<td>02:34:06 hours</td>
</tr>
<tr>
<td>Current CPUs load average</td>
<td>14 %</td>
</tr>
<tr>
<td>Concurrent connections</td>
<td>32</td>
</tr>
<tr>
<td>Health</td>
<td>NORMAL</td>
</tr>
</tbody>
</table>

### Chassis 1

- **ACTIVE**
  - SMs: 3 / 3
  - Ports: 2 / 2
  - Fans: 4 / 4
  - SMs: 2 / 2
  - CMMs: 2 / 2
  - Power Supplies: 6 / 6
  - **Pingable Hosts**: 1 / 1

### Chassis 2

- **STANDBY**
  - SMs: 2 / 2
  - Ports: 2 / 2
  - Fans: 4 / 4
  - SMs: 2 / 2
  - CMMs: 2 / 2
  - Power Supplies: 6 / 6
  - **Pingable Hosts**: 1 / 1

- **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>
  ```

SNMP

Description

61000 Security Systems implements an SNMP agent for:

- Extracting requested SNMP parameter(s) by means of the CLI or an SNMP management tool
- Sending SNMP traps

61000 Security Systems use two MIB files located on the gateway at `/CPDIR/lib/snmp`:

<table>
<thead>
<tr>
<th>MIB Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| chkpnt.mib | - A MIB-file that specifies available 61000 SNMP parameters.  
- OID: `1.3.6.1.4.1.2620.1.44`. The prefix for 61000 Security Systems SNMP parameters.  
  **Note** - Other SNMP OIDs available under the Check Point MIB might not reflect the actual status of the 61000 Security Systems. |
### MIB Name | Description
---|---
chkpnt-mbs-trap.mib | • A MIB-file that specifies available SNMP traps.  
• Traps start with the OID prefix: 1.3.6.1.4.1.2620.1.2001

**To enable SNMP:**

1. In SmartDashboard, add a rule that accepts these services for the 61000 Security Systems gateway:
   - SNMP
   - SNMP traps
2. Install the policy on the gateway
3. On the gateway, enable the SNMP service by running from gclish:
   ```
   > set snmp agent on
   ```
   **Note** - To make the setting persistent, from gclish run:
   ```
   save config
   ```

**To activate SNMP Traps:**

Use asg alert utility to configure and enable SNMP traps. [Refer to asg alert section](#).

**To make sure the SNMP service is enabled:**

From gclish, run:
```
show snmp agent:
```

1_01:
SNMP Daemon: Enabled
1_02:
SNMP Daemon: Enabled
1_03:
SNMP Daemon: Enabled

**Note** - Run SNMP GET operations using:
- The SNMP Management tool
- The bash shell on the gateway

**To disable the SNMP agent:**

- From gclish, run:
  ```
  set snmp agent off
  ```
- Make sure the SNMP agent is disabled by running:
  ```
  show snmp agent:
  ```

1_01:
SNMP Daemon: Disabled
1_02:
SNMP Daemon: Disabled
1_03:
SNMP Daemon: Disabled

- To make this setting persistent, from gclish run:
  ```
  save config
  ```

**To find SNMP parameters using the CLI:**

To find names, descriptions, and OIDs for an SNMP parameter, search the `chkpnt.mib` file. For example, if you need the SNMP parameter that returns how much time has elapsed since the last system startup, the mib file shows:

```plaintext
asgSystemUp OBJECT-TYPE
SYNTAX  DisplayString
ACCESS  read-only
STATUS  mandatory
DESCRIPTION
"Time elapsed since the last system startup"
 ::= { asg 11 }
```
The parameter is asgSystemUp, with the parameter's OID given in the last line: 11.

To extract SNMP values using the CLI

To extract the value of this SNMP parameter, use the snmpwalk command with the specified OID:

```
snmpwalk -v 2c -c public localhost 1.3.6.1.4.1.2620.1.44.11
SNMPv2-SMI::enterprises.2620.1.44.11.0 = STRING: "04:57:27 hours"
```

To extract OIDs and their values on the CLI:

To extract a list of OIDs and values, run:

```
snmpwalk -v 2c -c public localhost 1.3.6.1.4.1.2620.1.44
```

Note - An SNMP management tool can also be used to access any SNMP parameter specified in Check Point MIB (chkpnt.mib) or to receive SNMP traps specified in the MIB trap file (chkpnt-mbs-trap.mib).

To validate SNMP connectivity between SGMs and the SNMP management tool:

1. Do an SNMP GET operation from the SNMP Management tool.
2. On the gateway, make sure the tcpdump utility uses UDP port 162 by running:
   ```
   > tcpdump -nnni Mgmt host 194.29.47.75 and port 162
   ```

   Note -
   - port 162 is the port for SNMP traps
   - 194.29.47.75 is the IP address of the SNMP management tool

3. Run: tcpdump.

SNMP GetRequest and GetResponse messages should be in the output:

```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on Mgmt, link-type EN10MB (Ethernet), capture size 96 bytes
04:16:18.920123 IP 194.29.47.75.52184 > 172.23.9.91.161: GetRequest(31)
   .1.3.6.1.4.1.2620.1.44.1.0
04:16:18.920507 IP 172.23.9.91.161 > 194.29.47.75.52184: GetResponse(34)
   .1.3.6.1.4.1.2620.1.44.1.0="ASG"
```

4. Make sure SNMP packets arrived on ports 161 and 162 are not dropped by running:
   ```
   > fw ctl zdebug + drop
   ```

   Search for: Rulebase drop - rule x, where x is the rule number in SmartDashboard. Make sure the rule allows SNMP and SNMP trap services.

---

**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:

<table>
<thead>
<tr>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Print sync exceptions table</td>
<td>This view displays the sync exception table. Each entry in this table consists of: 1. &lt;5-tuple, including wild cards&gt; 2. synchronization mode (none, within chassis only, between chassis only, both within and between chassis) 3. SecureXL delayed synchronization value In addition, global synchronization values are displayed</td>
</tr>
<tr>
<td>2) Add new sync exceptions rule</td>
<td>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</td>
</tr>
<tr>
<td>3) Delete old sync exception rule</td>
<td>Delete rule from the sync exceptions table</td>
</tr>
<tr>
<td>4) Set sync between chassis flag on / off</td>
<td>Global system setting: whether to synchronize connections to backup chassis</td>
</tr>
<tr>
<td>5) Set sync within local chassis flag on / off</td>
<td>Global system setting: whether to synchronize connections within active chassis</td>
</tr>
<tr>
<td>6) Configure sync between chassis blades ratio</td>
<td>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</td>
</tr>
<tr>
<td>7) Set default delay notifications</td>
<td>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</td>
</tr>
<tr>
<td>8) Enable / Disable unicast sync</td>
<td>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</td>
</tr>
</tbody>
</table>

**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:

```plaintext
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
>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)
>y
```

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>Tps</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
Default delays: chassis - 5

Sync Values:

| 0 = no sync
| 1 = sync only to local chassis
| 2 = sync only to other chassis
| 3 = sync to all chassis

Delay:

The time it takes for connections created from templates to synchronize

Press enter to continue
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
```
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
New glclish command asg_ntp_sync_config was implemented to allow configuring automatic time updates from NTP server.

```
asg_ntp_sync_config on <NTP Server IP | hostname> [-v <NTP version>] [-r <Refresh Timeout>] -
```  
configures automatic ntp time synchronization each Refresh Timeout seconds from the NTP server. If Refresh Timeout is not specified, it defaults to 5 minutes. If NTP version is not specified, NTPv4 will be used.

```
asg_ntp_sync_config off - disables automatic time synchronization via NTP
```

The command updates all blades with the NTP server IP address and other settings into /config/active (equal to running the glcish command "set ntp server primary <IP Address> version <NTP version>" and schedules running of the script $FWDIR/bin/asg_ntp_update_time, which pulls the time from the NTP server. The scheduled task can be viewed by running the command 'cpd_sched_config print'.

Each Refresh Timeout seconds, the local time is updated on each blade by running the command 'ntpd -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 glcish, validate same time on all SGMs
- Execute tcpdump on port 123/UDP on the relevant interface and verify that all SGMs initiate NTP connections

Comments
- If new blade is added after configuring NTP time synchronization, the script needs to be re-run to schedule it also on the new blade.

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.

**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 1/2/1-1/14/1")
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 '^]'.

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.
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_chassis_ctrl jumbo_frames 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.

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>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.

# 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.

# 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.
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:
# 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
Configuration loaded

1_02:
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
Configuration loaded

1_03:
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
Configuration loaded

1_04:
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
Proxy ARP for Manual NAT – (local.arp file)

Description:
Proxy ARP is a mechanism that allows the configuration of a GW 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 61K GW:
1. Add any IPs for which 61k 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 61K appliance.
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.

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
Chapter 3

61000 Security Systems Miscellaneous Commands

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 Systems, 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 two ways
1. 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
   Unload policy requires auditing
   Enter your full name: jhon
   Enter reason for unload policy [Maintenance]:
   Unloading policy from remote SGMs succeed
   Unloading policy from local SGM. succeed
   Unloading policy from local SGM succeed
   ```
   
   **Note** - It is recommended to run this command over a serial connection.

2. From SmartDashboard **Policy > Uninstall**

**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:
   
   ```
   # 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.

  **Chassis ID: 1**

<table>
<thead>
<tr>
<th>Task (Task ID)</th>
<th>Blade ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMO (0)</td>
<td>1(local)</td>
</tr>
<tr>
<td>CH Monitor (3)</td>
<td>1(local)</td>
</tr>
<tr>
<td>UTPC (5)</td>
<td>1(local)</td>
</tr>
<tr>
<td>General (1)</td>
<td>2</td>
</tr>
<tr>
<td>DR Manager (4)</td>
<td>2</td>
</tr>
<tr>
<td>LACP (2)</td>
<td>3</td>
</tr>
</tbody>
</table>

  **Chassis ID: 2**

<table>
<thead>
<tr>
<th>Task (Task ID)</th>
<th>Blade ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH Monitor (3)</td>
<td>1</td>
</tr>
<tr>
<td>UTPC (5)</td>
<td>1</td>
</tr>
<tr>
<td>General (1)</td>
<td>2</td>
</tr>
<tr>
<td>LACP (2)</td>
<td>3</td>
</tr>
</tbody>
</table>

- asg monitor
Use this command to monitor policy installation.

Sun Jul 17 13:49:29 IDF 2011

```
<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**

This command makes sure the SGMs have the same policy installed.

```
<table>
<thead>
<tr>
<th>Blade ID</th>
<th>Policy Name</th>
<th>Policy Date</th>
<th>Policy Signature</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1_C1</td>
<td>any_any_accept</td>
<td>14Jul11 18:23</td>
<td>fde5e75c1</td>
<td>OK</td>
</tr>
<tr>
<td>1_C2</td>
<td>any_any_accept</td>
<td>14Jul11 18:23</td>
<td>fde5e75c1</td>
<td>OK</td>
</tr>
<tr>
<td>1_C3</td>
<td>any_any_accept</td>
<td>14Jul11 18:23</td>
<td>fde5e75c1</td>
<td>OK</td>
</tr>
<tr>
<td>2_C1</td>
<td>any_any_accept</td>
<td>14Jul11 18:23</td>
<td>fde5e75c1</td>
<td>OK</td>
</tr>
<tr>
<td>2_C2</td>
<td>any_any_accept</td>
<td>14Jul11 18:23</td>
<td>fde5e75c1</td>
<td>OK</td>
</tr>
<tr>
<td>2_C3</td>
<td>any_any_accept</td>
<td>14Jul11 18:23</td>
<td>fde5e75c1</td>
<td>OK</td>
</tr>
</tbody>
</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>

## 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.
Extending SecureXL Templates

Syntax:
```
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.

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 wildcard, 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_srcip_wildcard_for_template=1` to the file.
4. Copy the file to all SGMs by running:
   `g_cp2blades -a /etc/ppk.boot/boot/modules/simkern.conf`
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/bootstrap/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
   Instrumental Template: 1
   Source Port Destination DPort PR Flags Conns LCT DLY CTS i/f SIC i/f Inst Identity
   * 11.111.1100 22 6 ...... 1 2 7 2/5 3/1 1 0
   ```

   An asterisk (*) in the **Source** column and an increasing **Conns** counter means the extended template is being utilized.

### Resetting SIC (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: Reset 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: `g_cpconfig sic init <activation key>`.
      Reset is completed after 3-5 minutes.

2. In SmartDashboard:
   a) Open the gateway's **General Properties > Communication** window.
   b) Click **Reset**.
   c) Enter the same activation key used in step 1.
d) Click Initialize.

3. On the gateway, run: `g_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 <code>SmartDashboard &gt; General Properties &gt; Communication</code> 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.

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
- FW blade only is enabled
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:
  o Run g_fwaccel stats
  o Save the old value of the “Policy deleted tmpl” statistic
  o Install policy
  o Run g_fwaccel stats again
  o Make sure that templates were deleted
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
```

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`)

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;filename&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:
```
> 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
```
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:

`asg_tracert <IP Address> <tracert options>`

Example:

`asg_tracert 100.100.100.99`
`asg_tracert 100.100.100.99 --udp`

Output:

```
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: “tracert –z 500 100.100.100.99”

“asg_tracert 100.100.100.99 --udp” runs the following command: “tracert –z 500 100.100.100.99 --udp”
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 61k to work as a RADIUS client. 61k does not include RADIUS server functionality. You can configure 61k to authenticate users even when they are not defined locally. See Configuring Non-local RADIUS Users.

You can configure your 61k computer to connect to more 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.

Note: On R75.035 RADIUS server is accessed through data interface only. Radius Server cannon be accessed through the management interface. (This is supported from 61k R75.050)

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>priority</td>
<td>RADIUS server priority as an integer between 0 and 999 (default=0). When there two or more RADIUS servers, Gaia connects to the server with the highest priority. Low numbers have the higher priority.</td>
</tr>
<tr>
<td>new-priority</td>
<td>New RADIUS server priority as an integer between 0 and 999 (default=0). When there two or more RADIUS servers, Gaia connects to the server with the highest priority. Low numbers have the higher priority.</td>
</tr>
<tr>
<td>host</td>
<td>RADIUS server IP address in dot-delimited format.</td>
</tr>
<tr>
<td>port</td>
<td>UDP port on the RADIUS server. This value must match the port as configured on the RADIUS server. Typically this 1812 (default) or 1645 (non-standard but a commonly used alternative).</td>
</tr>
<tr>
<td>prompt secret</td>
<td>Shared secret (password) text string. The system prompts you to enter the value.</td>
</tr>
<tr>
<td>timeout</td>
<td>The number of seconds to wait for the server to respond. The default value 3 seconds.</td>
</tr>
<tr>
<td>secret</td>
<td>The shared secret used to authenticate the RADIUS server and the local client. You must define this value on your RADIUS server.</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 61k, 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 61k 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 61k 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>
<tr>
<td>Roles</td>
<td>The role to assign to the user.</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>
Chapter 4

61000 Security Systems - Appendix

Policy and Configuration Cloning

In 61000 GW, 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. FW1 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 Gateway Module, 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:

- `cpstart`
- `asg_blade_admin up`

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

<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>

#### 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.

### 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
```

**Output**

```
SGM : Policy Name: Policy Date: Policy Signature: Verification:
1.01 any_any_accept 14Jul11 18:23 fc60f75c1 OK
1.02 any_any_accept 14Jul11 18:23 fc60f75c1 OK
1.03 any_any_accept 14Jul11 18:23 fc60f75c1 OK
2.01 any_any_accept 14Jul11 18:23 fc60f75c1 OK
2.02 any_any_accept 14Jul11 18:23 fc60f75c1 OK
2.03 any_any_accept 14Jul11 18:23 fc60f75c1 OK
```

OK shows in the verification column.

### SW Upgrade Procedure

**Description:**

61K supports upgrade procedure on Dual Chassis setup. The procedure assures network connectivity at all times during the upgrade by maintaining at least one Active Chassis that handles traffic. Old connections, meaning connections that were opened on the chassis with the old version, will survive the upgrade only when upgrading between minor versions.

**Upgrade procedure:**
1. Copy new_version snapshot file (*.tar) to SMO SGM, and from SMO to all SGMs
   [asg_cp2blades].

2. Verify MD5 sum on the copied file: [g_md5sum <new_version_snapshot_file_path>].

3. Run snapshot import on all SGMs via gclish:
   [set snapshot import <new_version_snapshot_file (without .tar)> path <snapshot_directory>]

4. Perform down admin to standby chassis via gclish:
   [asg_chassis_admin -c <stand_by_chassis_id> down]

5. This step is relevant only when upgrading form R75.01 version:
   Copy "g_snapshot" script to all SGMs (the script should be supplied with the snapshot file):
   Copy the script to all SGM (to: $FWDIR/bin/), and apply it with executable permission on all SGMs:
   [asg_cp2blades $FWDIR/bin/g_snapshpt]
   [g_all chmod +x $FWDIR/bin/g_snapshpt]

6. Revert to the new snapshot on the down admin chassis via 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]

7. Wait until the chassis is UP, running the new version (verify with [ver] via gclish).

8. This step is relevant only when upgrading form R75.01 version:
   Set chassis CMM factor to be 6 on all SGMs:
   [set chassis high-availability factors sensor cmm 6]

9. Perform Chassis admin up to the upgraded chassis via gclish:
   [asg_chassis_admin -c <chassis_id> up]

10. Perform Chassis admin down to the active chassis via gclish:
    [asg_chassis_admin -c <chassis_id> down]

11. Revert to new snapshot on the down admin chassis via shell:
    [g_snapshot <blade_string> revert <snapshot_name>]

12. Wait until the chassis is be UP, running the new version (verify with [ver] via gclish)

13. This step relevant only when upgrading to R75.035 version:
    Change fans factor to be 5 (new default value) via gclish:
    [set chassis high-availability factors sensor fans 5]

Notes:
1. The procedure requires Dual Chassis setup
2. The procedure is supported for upgrade from R75.01 to R75.03 or to R75.035, and for upgrade from R75.03 to R75.035 (refer to official version names section in the CLI guide appendix).
3. All SGMs should be in UP state during the upgrade
4. Both SSH or console can be used

DB Changes between R75.01 to new versions:
Due to DB changes between R75.01 and newer versions, an automatic DB conversion is performed during the upgrade. That section is relevant only when upgrading form R75.01 version.

Following is a list of the changes in the DB that will be automatically performed:
- File: "/config/active"-"chassis:high-availability:factors:sensor:cmm"
  - Default CMM factor would be changed to 6 after the upgrade
  - eth1-Mgmt4 configuration will be back to new version default values
- File: "$FWDIR/conf/alert.conf"
  - Alerts configuration will not survive the upgrade but will be back to the new version default configuration. Note, old configuration is saved under "/var/log/upgrade/alert.conf.dbver100"
Conversion log file can be found under: "/var/log/upgrade/"

Verification:
Verify the current snapshot on all SGMs (via gclish):
"show snapshots"

Screenshots:
“show snapshots” output:
```
gcpmodue-ch02-01 > show snapshots
1_02:
Restore points:
----------------
ssm60_t2_pol
hp_d_t39_gaia

Creation of an additional restore point will need 2.532G
Amount of space available for restore points is 41.84G

1_03:
Restore points:
----------------
ssm60_t2_pol
hp_d_t39_gaia

Creation of an additional restore point will need 2.532G
Amount of space available for restore points is 41.84G

2_01:
Restore points:
----------------
ssm60_t2_pol
hp_d_t39_gaia

Creation of an additional restore point will need 2.532G
Amount of space available for restore points is 41.84G

2_02:
Restore points:
----------------
ssm60_t2_pol
hp_d_t39_gaia

Creation of an additional restore point will need 2.532G
Amount of space available for restore points is 41.84G
```

SSMs Upgrade Procedure - from SSM60 to SSM160

Procedure layout:
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 61k, get and save the following chassis configuration for later verifications:
      - `asg if`
      - `asg stat -v`
      - `asg diag`
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. Reconfigure management interfaces if needed. See **SSM Upgrade - Appendix** below.
   g. 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` – 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
   h. Configure the GARP’s 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`.

---

**Related topics**
- **SSM Upgrade - Appendix**: Details on disconnecting and reconnecting Standby and Active chassis, update firmware, and verify configuration.
d. Very 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.g 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>
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 61k 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
```

VPN Packet Tracking

Use these commands to track IPSEC packet flow.

<table>
<thead>
<tr>
<th>To see:</th>
<th>Run:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source and destination IP addresses</td>
<td>• g_tcpdump for ip proto 50 (For Site-to-Site VPN)</td>
</tr>
<tr>
<td></td>
<td>• g_tcpdump for UDP port 4500 (For SecureClient and Endpoint VPN clients)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Which SGM receives encrypted traffic</td>
<td>asg_dxl calc &lt;src_ip,dst_ip&gt;</td>
</tr>
<tr>
<td>Which SGM encrypted packets are forwarded to</td>
<td>bcstats vpn -v</td>
</tr>
<tr>
<td>Which SGM holds the outbound SA</td>
<td>g_fw tab -t outbound_SPI -f</td>
</tr>
<tr>
<td></td>
<td>• Search for MSPI in the output. MSPI represents the Meta SA, and shows which SGM holds the outbound SA. For example:</td>
</tr>
</tbody>
</table>

# FW tab -t outboundСПІ -f
using_config
processing rule's data - this might take a while...
localhost

12:37:23  172.16.6.189 : (c) TABLE NAME: outbound_SPI (c) Attributes: dynamic, id 205, attributes: empty, ip, ip4 0.0, expire 3600, src 10.0.0.0, hashTo 379280, processor: 1 & 1
12:37:23  172.16.6.379 : (c) peer: 172.16.6.189; SPI 379280; CRYPTO: esp; id: 47056711; PROCESS: 1; cookie: c5e496d44a0b, cookie: c5e496d44a0b
95a7f880cf954460; expire: 379280; product: vpn-1 & 1

#
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@CCS-ch01-013# asg_dr_verifier

General:
DR Manager: SGN_1.0.1

Routes status:
  -> 8 blades: 1_01 1_02 1_03 1_04 2_01 2_02 2_03 2_04
RouteSource: Networks
  connected: 7
  kernel: 3
  static: 3
  bgp: 12
  ospf: 1101
  Total: 1123

OSPF:

OSPF interface
  -> 8 blades: 1_01 1_02 1_03 1_04 2_01 2_02 2_03 2_04

Name        IP Address   Area ID   State   DR Interface   BDR Interface
eth2-01     192.168.33.66 0.0.0.0   BDR     192.168.33.235 192.168.33.66

Status: OK

OSPF neighbors:
  -> 8 blades: 1_01 1_02 1_03 1_04 2_01 2_02 2_03 2_04

Neighbor  Pri  State   Address   Interface
192.168.33.235 1   FULL/DR 192.168.33.235 192.168.33.86

Status: OK

BGP:

BGP peers -
  -> 1 blade: 1_01 (DR Manager)

PeerID        AS  State   ActRts   Routes   InUrcs   OutUrcs   Uptime
192.168.33.230 30 Established 14 42       3       2       05:26:54

  -> 7 blades: 1_02 1_03 1_04 2_01 2_02 2_03 2_04

PeerID        AS  State
192.168.33.230 30   Idle

Status: OK

[Expert@CCS-ch01-013#]
```
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:

```bash
[Expert@ACDC-ch02-01]# mgc_dr_verifier all

General:
DR Manager:
SGM 2_01

Routes status:

- 8 blade: 1_01 1_02 1_03 1_04 2_01 2_02 2_03 2_04
RouteSource Networks

connected 7
kernel 0
static 2
gp 0
gp 0
Total 9

OSPF:

OSPF interfaces:

- 8 blade: 1_01 1_02 1_03 1_04 2_01 2_02 2_03 2_04

<table>
<thead>
<tr>
<th>Name</th>
<th>IP Address</th>
<th>Area ID</th>
<th>State</th>
<th>HC</th>
<th>DR Interface</th>
<th>BDR Interface</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth0-01</td>
<td>192.168.33.86</td>
<td>0,0,0,0</td>
<td>BGP</td>
<td>1</td>
<td>192.168.33.235</td>
<td>192.168.33.86</td>
<td>0</td>
</tr>
</tbody>
</table>

Status: OK

OSPF neighbors:

- 6 blade: 1_01 1_02 1_03 1_04 2_01 2_03

<table>
<thead>
<tr>
<th>Neighbor ID</th>
<th>Pri</th>
<th>State</th>
<th>Dead Address</th>
<th>Interface</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.33.235</td>
<td>1</td>
<td>FULL/DR</td>
<td>192.168.33.235</td>
<td>192.168.33.86</td>
<td>0</td>
</tr>
</tbody>
</table>

- 2 blade: 2_02 2_04

<table>
<thead>
<tr>
<th>Neighbor ID</th>
<th>Pri</th>
<th>State</th>
<th>Dead Address</th>
<th>Interface</th>
<th>Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.33.235</td>
<td>1</td>
<td>FULL/DR</td>
<td>192.168.33.235</td>
<td>192.168.33.86</td>
<td>0</td>
</tr>
</tbody>
</table>

Status: Inconsistency found on some of the SGMs

EIGRP:

EIGRP is not configured on this host.
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
- 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.
MAC Verifier (mac_verifier)

Description:
In 61K 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@hp1_35-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

Output example when inconsistency found:

[Expert@hp1_35-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

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.
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 ways to connect a CMM CLI:

- Connect to the serial port on the front panel of the CMM
  - In your terminal emulation program, set the baud rate to 9600
  - Enter `admin` for the user name and password
- Open a telnet or SSH session from one of the SGMs
  - First make sure that you have connectivity to the CMMs by pinging both addresses:
    - 198.51.100.33 (routed via SSM1)
    - 198.51.100.233 (routed from SSM2)
  - Telnet or ssh from the SGM to the CMM
  - 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`
- See alerts by running: `clia alarm`
- Reset alerts by running: `clia alarm 0`
- See power consumption details by running: `clia shef pd`
- Retrieve events logs by running: `clia sel`
- Reboot the CMM by running: `reboot` (which initiates a failover to the standby CMM)

CMM debug commands – How to active the log function:

- `loginto the active ShMM`
- run `/etc/summary` - this can take some minutes
- run `cat /tmp/debug.log` - this will print the debug log with all basic information
- run `i2c_test` - this will test the internal ShMM I2C and print all devices connected on the I2C
- run `cat /etc/shmm.cfg` - this will printout the ShMM custom configuration
- run `clia fruinfo 20 x` - 17 times where x is 0 to 16
- run `clia fruinfo y 0` - 16 times where y is 10,12,82,84,86,88,8a,8c,8e,90,92,94,96,98,9a,9c
- Close your terminal program. `/tmp/debug.log` file will hold the debug information.

**SMM60 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 SMM and run commands.

There are two ways to connect to the SMM CLI:

- Connect to a serial port on the front panel of the SMM.
  The SSM60 has two serial ports, one for the fabric switch (data ports) and one for the base switch (management ports).

  - In your terminal emulation program, set the baud rate to 9600.
  - 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.

  **Note** - Load balancing commands are run on the fabric switch only.

- Open a telnet session from one of the SGMs.
  - First make sure that you have connectivity to the SSMs by pinging these addresses:

<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>Fabric</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>Fabric</td>
<td>198.51.100.232</td>
</tr>
</tbody>
</table>

  - Telnet from the SGM to the SMM
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># show running-config</td>
</tr>
<tr>
<td>View current ports status</td>
<td># show interface</td>
</tr>
<tr>
<td>View interface statistics</td>
<td># show interface &lt;interface ID&gt; statistics [extended]</td>
</tr>
<tr>
<td>View SSM logs</td>
<td>#show log buffer</td>
</tr>
</tbody>
</table>
| Modify the group of SGMs amongst which the load is distributed | # configure terminal
(config)# load-balance mtx-bucket [SGM ID, SGM ID,]
(config)# load-balance apply |

Note: the command will not work if you have an odd number of SGMs in the group. For example, do not run:

#load-balance mtx-bucket 1,2,3
Run:
#load-balance mtx-bucket 1,2,3,1,2,3

SSM160 CLI

Description:
The SSM (Security Switch Module) is the networking module of the gateway. It transmits traffic to and from the SGM and performs the load distribution among the SGMs. The SSM includes two modules, the fabric switch which includes the data ports and the base switch which 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:
- The administrator can connect with a serial console to the “CLI” port on the SSM front panel (baud rate 9600).
- From one of the SGMs use ssh to connect to the SSM. The SSM IPs can be retrieved by “show chassis id <1|2|all> module SSM<1|2> ip” from clish/gclish.

The password for the SSM is “admin”.

Once connected to the SSM CLI you can do the following:
- 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.
- View current ports status:
  #show port
- View detailed port information (speed, administrative state, link state, etc.):
  #show port <port ID>
- View interface statistics:
  # show port <Port ID> statistics
Pay special intention to "Discards" and "Errors" fields which might indicate on 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 that 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 that the process 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
Each port ID on the SGM maps to a port on the SSM, the below table maps the SSM port IDs to the SGM port IDs. Note that it relates to SSM1, for SSM2 simply replace eth1-X with eth2-X:

<table>
<thead>
<tr>
<th>SGM</th>
<th>SSM</th>
<th>SGM</th>
<th>SSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>eth1-01</td>
<td>1/3/1</td>
<td>eth1-11</td>
<td>1/1/3</td>
</tr>
<tr>
<td>eth1-02</td>
<td>1/3/2</td>
<td>eth1-12</td>
<td>1/1/4</td>
</tr>
<tr>
<td>eth1-03</td>
<td>1/3/3</td>
<td>eth1-13</td>
<td>1/2/1</td>
</tr>
<tr>
<td>eth1-04</td>
<td>1/3/4</td>
<td>eth1-14</td>
<td>1/2/2</td>
</tr>
<tr>
<td>eth1-05</td>
<td>1/3/5</td>
<td>eth1-15</td>
<td>1/2/3</td>
</tr>
<tr>
<td>eth1-06</td>
<td>1/3/6</td>
<td>eth1-16</td>
<td>1/2/4</td>
</tr>
<tr>
<td>eth1-07</td>
<td>1/3/7</td>
<td>eth1-Mgmt1</td>
<td>1/5/1</td>
</tr>
<tr>
<td>eth1-Sync</td>
<td>1/3/8</td>
<td>eth1-Mgmt2</td>
<td>1/5/2</td>
</tr>
<tr>
<td>eth1-09</td>
<td>1/1/1</td>
<td>eth1-Mgmt3</td>
<td>1/5/3</td>
</tr>
<tr>
<td>eth1-10</td>
<td>1/1/2</td>
<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 "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>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Out of service</td>
<td>Red: SGM out of service</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Off (Normal): SGM hardware is normal</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Health</td>
<td>Green (Normal): SGM core operating system is active</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Green blinking: SGM core operating system is partially active</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Off: SGM operating system is in standby mode</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Hot-swap</td>
<td>Blue: SGM can be safely removed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Blue blinking: SGM is going to standby mode. Do not remove</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Off (Normal): SGM is active. Do not remove</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Link</td>
<td>Yellow: Link enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Yellow blinking: Link is active</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Off: Link is disabled</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>Speed - data ports</td>
<td>Yellow: 10 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Green: 1 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Off: 100 Mbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Speed - management port</td>
<td>Yellow: 1 Gbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Green: 100 Mbps</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Off: 10 Mbps</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>L</td>
<td>LEDs 2 and 4 - Green: SGM is being configured. (Using First Time Wizard or adding a new SGM into a chassis)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>All LEDs - Off: SGM is configured and ready</td>
</tr>
</tbody>
</table>

![Diagram of Security Gateway Module LEDs]
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>SSM out of service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off (Normal)</td>
<td>SSM hardware is normal</td>
</tr>
<tr>
<td>2</td>
<td>Power</td>
<td>On (Normal)</td>
<td>Power on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off</td>
<td>Power off</td>
</tr>
<tr>
<td>3</td>
<td>Hot-swap</td>
<td>Blue</td>
<td>SSM can be safely removed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blue blinking</td>
<td>SSM is going to standby mode. Do not remove</td>
</tr>
<tr>
<td>4</td>
<td>SYN ACT</td>
<td>On (Normal)</td>
<td>Normal operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Off</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>Link</td>
<td>On</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>
</tbody>
</table>

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. Disassemble the upper CMM
2. Log in to the 61000 Security System CMM.
   1. 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>
   PWR_SET="41 0 320"
   PWR_SET="42 0 320"
   PWR_SET="43 0 320"
   PWR_SET="44 0 320"
   PWR_SET="45 0 320"
   PWR_SET="46 0 320"
   PWR_SET="47 0 320"
   PWR_SET="48 0 320"
   PWR_SET="49 0 320"
   PWR_SET="4a 0 320"
   PWR_SET="4b 0 320"
   PWR_SET="4c 0 320"
   PWR_SET="4d 0 320"
   PWR_SET="4e 0 320"
   # add ... others
   # SNMP Credential
   SNMP_ruser="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. Disassemble the lower CMM that just was reconfigured
5. Assemble the upper CMM to the chassis
6. Run the step 1 - 3 on the upper CMM
7. Disassemble the upper CMM
8. Assemble both CMMs back to 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 disassemble and re-assemble all SGMs
Related debug files

List of 61000-related debug files:

<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/send_alert.*</td>
</tr>
<tr>
<td>Distribution</td>
<td>/var/log/dist_mode.log.*</td>
</tr>
<tr>
<td>Installation – OS</td>
<td>/var/log/anaconda</td>
</tr>
<tr>
<td>Installation – 61K</td>
<td>/var/log/start_mbs.log</td>
</tr>
<tr>
<td>Installation – 61K</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/fw.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>

Official 61k version names

The following is the list of 61K version names.

<table>
<thead>
<tr>
<th>Release Date</th>
<th>Functionality</th>
<th>Official Name</th>
<th>Take Number</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q4’11</td>
<td>First release, SSM60 support</td>
<td>61000 R75.010</td>
<td>Take #2</td>
<td>Obsolete</td>
</tr>
<tr>
<td>Q4’11</td>
<td>First official release, SSM160 support</td>
<td>61000 R75.030</td>
<td>Take #43</td>
<td>Obsolete</td>
</tr>
<tr>
<td>Q1’12</td>
<td>Customer fixes accumulator</td>
<td>61000 R75.035</td>
<td>Take #55</td>
<td>Supported</td>
</tr>
</tbody>
</table>

To verify which version is installed on 61K GW use gclish ‘ver’ command.

Example for ‘ver’, executed on 61000 R75.030:

glab-ch01-01 > ver
1_01:
Product version Check Point 61000
OS build 43
OS kernel version 2.6.18-92cpx86_64
OS edition 64-bit