Secure Web Gateway
Version 11.8
High Availability

Legal Notice

The most current version of this document may be obtained by contacting:

Trustwave Technical Support:
Phone: +1.800.363.1621
Email: support@trustwave.com

Trademarks

Revision History

Version	Date	Changes
11.0	July 2013	First release
11.5	December 2013	Minor revisions
11.6	December 2014	Version update
11.7	March 2015	Version update
11.8	August 2016	Version update

Formatting Conventions

This manual uses the following formatting conventions to denote specific information.

Formats and Symbols	Meaning
Blue	Blue text indicates a Web site or e-mail address.
Bold	Bold text denotes UI control and names such as commands, menu items, tab and field names, button and checkbox names, window and dialog box names, and areas of windows or dialog boxes.
`Code`	Text in `Courier New 9 pt in blue` indicates computer code or information at a command line.
Italics	Italics denotes the name of a published work, the current document, name of another document, text emphasis, to introduce a new term, and path names.
[Square brackets]	Square brackets indicate a placeholder for values and expressions.

Notes, Tips, and Cautions

Note: This symbol indicates information that applies to the task at hand.

Tip: This symbol denotes a suggestion for a better or more productive way to use the product.

Caution: This symbol highlights a warning against using the software in an unintended manner.

Question: This symbol indicates a question that the reader should consider.

Table of Contents

1 Overview

To ensure continuous operation in case of a policy server failure, SWG supports High Availability, which is implemented by adding a secondary Passive Policy Server device to the system. Specific data is automatically replicated, updated and synchronized between the servers.

In the event of failure of the Active Policy Server, SWG automatically fails over to the Passive Policy Server, making it the primary Active Policy Server.

When the failed server can again be used, SWG designates it as the Passive Policy Server.

Note: To switch a Passive policy server to Active, you must manually perform the change on the active device using the **failover** Limited Shell command.
For more information on Limited Shell commands, see the *SWG Management Console Reference Guide*.

The high availability process includes:

- Deciding which device is active and which is passive.
- Switching automatically between active and passive devices when the active device is not functioning.
- Setting a virtual IP on the active device so that the user can view the active GUI without knowing the active device IP.

1.1 Requirements

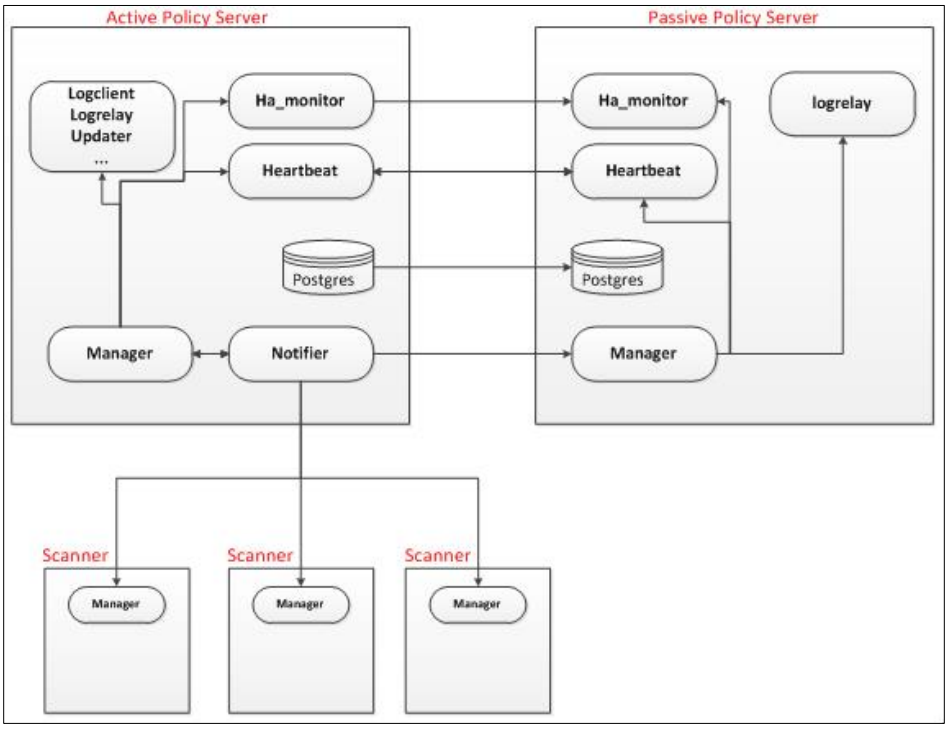
- Only one Active Policy Server is defined and only one Passive Policy Server is used for failover.
- The primary Active and secondary Passive Policy Servers are on separate devices, not on an All-In-One device.
- The device that houses the secondary Passive Policy Server is accessible and its IP address is known.
- Both policy servers are on the same network.
- Both policy servers are running the same SWG version.
- Linux-ha is installed on each policy server.
- A virtual IP given to the high availability system (the set of active and passive devices). Access to the GUI is recommended via the virtual IP. In addition, the scanners will send traps to the policy server via this virtual IP. Linux-ha is responsible for shifting the virtual IP to the active device.
- To prevent a split brain situation, the active and passive policy servers must be connected by two switches. This prevents a situation where both policy servers can communicate with the scanners but not to each other, thus thinking they are both active.

2 How it Works

The Manager runs several roles on the passive Policy Server. These roles are written in the **/etc/Manager/commander_passive_module.xml** file:

The HA role added to **manager.conf.xml** holds two processes: **ha_manager** and **heartbeat**.

- ha_manager – The process that manages the high availability system.
- heartbeat – An open source Linux-ha process whose main component implements a Heartbeat protocol. For more information about Linux-HA, see http://www.linux-ha.org
- **logrelay** – The process that enables the Active Server to retrieve system logs.



The Manager treats the active Policy Server in the same way as a regular policy server. The role HA should be enabled in the **Commander/module.xml** file.

The Manager listens to the Notifier running on the active Policy Server. It stops listening to the local Notifier when the Notifier on the active device sends the **isPassive=1** flag within the **status** command.

Running the **Manager-ctl reload [passive]** command in Manager-ctl tells the Manager to stop all roles and start only roles listed as enabled in **commander_passive_module.xml** or **Commander/module.xml**.

When the Manager runs in passive mode, it creates the **/etc/Manager/passive_device** file. When the Manager starts, it checks for the existence of the file, and if it exists, it loads the configuration from **commander_passive_module.xml**. When the Manager runs in an active mode, it deletes the **Manager_passive** file if it exists.

The Policy Server saves its configuration in the database according to its device ID. When running **full_replicate**, the configurations in the database are copied to files located at **/var/policyserver/configuration/base/[global|deviceId]**.

In order for the active policy server configuration to be replicated to the passive Policy Server device, the device ID in the policy server database is changed to suit the passive device ID. This is done on failover, when passive becomes active, by calling **active_request_cli**. (The ha_manager is responsible for this.)

In order for the configuration located in **/var/policyserver/configuration/base/[passive_device_id]** to match the active policy server configuration, on failover after running active_request_cli, the ha_manager will call **full_replicate**.

2.1 ha_manager

ha_manager is the process designed for High Availability. Its main task is to:

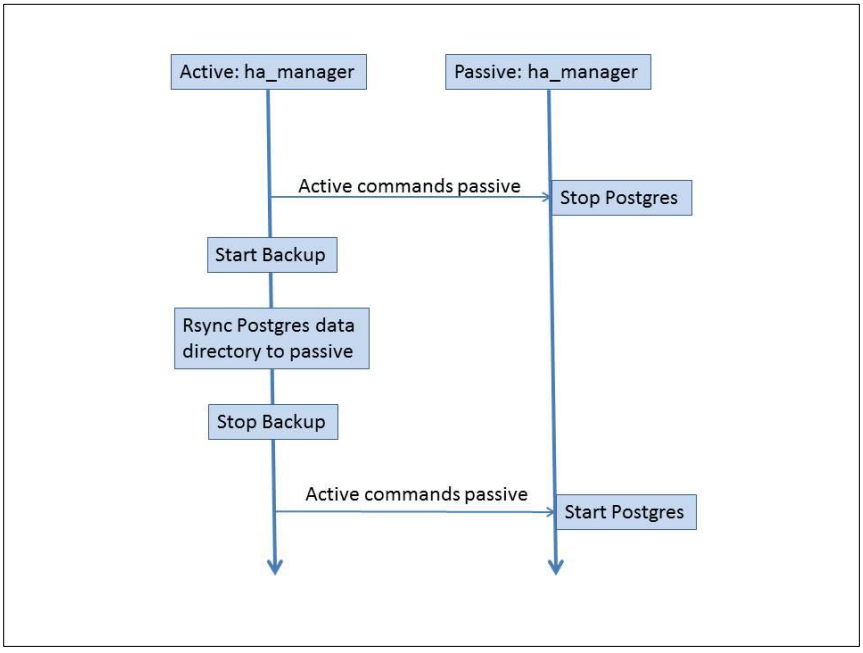
- Start, stop and monitor the PostgreSQL (Postgres) replication.
- Copy additional files to the passive Policy Server on demand.
- Perform failover when required.

ha_manager supports the following signals:

1. **SIGHUP** – reloads configuration.
2. **SIGUSR1** – performs a failover (by restarting Heartbeat)

The Manager starts, stops and monitors the ha_manager process on both active and passive policy servers. It does not run on scanners. ha_manager will keep running until stopped by the Manager.

Postgres Replication:



The ha_manager checks the status of Heartbeat every x interval (as defined in **HA/module.xml**) by running the Linux-ha cli command **cl_status nodestatus**. According to the status, it decides if the device is active or passive. The cl_status can be one of the following:

1. **All:** This is the active device
2. **None:** This is the passive device
3. **Local:**
    a. If the device is the default active (the HA was configured on this device) then this is the Active device.
    b. If the device is the default passive (this device was first attached to the Active device):
        - This device will be passive if it was passive before or if the other device is Active.
        - This device will be active only if it was active already and the other device is passive. This can happen only if the device was active before, and the heartbeat was killed and restarted before the deadtime timeout.

If the device is active, the ha_manager will:

1. Create a file **/etc/ha_manager/active** indicating this device is the Active device.
2. Check if a failover occurred (If the device was passive before). If so, it will:
    a. Copy **/var/wasp/conf_ready** to **/var/policyserver/configuration/base**
    b. Run **manager-ctl reload**, which will tell the Manager to start all roles defined in **Commander/module.xml**.
    c. Move watched files from **/opt/finjan/configuration/ha** to their original location.
    d. Create a Postgres trigger file telling it that it should run in Active mode.
    e. Start the Policy Server and run **active_request_cli** which tells the policy server that this is the Active device.
    f. Run **full_replicate**.
3. Check the Postgres status of the active and passive device. If Postgres is not running in replication mode on both devices, the ha_manager will copy the Postgres data directory to the passive device as described above.
4. Copy files defined in **HA/module.xml** to the passive device directory **/opt/finjan/configuration/**.

If the device is passive, the ha_manager will:

1. Create a file **/etc/ha_manager/passive** indicating this device is the passive device.
2. Run **manager-ctl reload passive** which will tell the Manager to run only roles defined in **commander_passive_module.xml**.
3. Stop the Policy Server.
4. Listen to commands from ha_manager at the active device.

2.1.1 System Logs

The ha_manager will send system logs when it:

1. Finds out from the Heartbeat that the status of the device was changed to active or passive.
2. Starts Postgres replication.
3. Finishes the initialization of the Postgres replication.
4. Fails to connect to the ha_manager running on the passive Policy Server.

2.2 Heartbeat

The Heartbeat process runs on both active and passive policy servers (not on scanners). The Manager will start, stop and monitor the Heartbeat process.

In this process, interval messages are sent between devices. If a message is not received from a device then the device is assumed to have failed. In case the failed device is the active device, Heartbeat performs a failover and the passive device becomes the active device.

When running a proper shutdown of the Heartbeat using the **/etc/init.d/heartbeat stop** command, it will cause a failover. To avoid a failover every shutdown, SWG kills the heartbeat when restarting (using **killall -9 heartbeat**).

2.2.1 Configuring Heartbeat

The HA configuration is saved in the file **/var/wasp/conf/ha/current/module.xml** like any other process in the system. The **module.xml** file holds the following parameters:

- **ha_enabled** - If HA is enabled, it will hold the value 1. Otherwise 0.
- **virtual_ip** - The virtual IP of the HA system. Can be empty if no virtual IP is defined.
- **default_active** - The IP and the name (as it appears in the **uname –n** command) of the default active Policy Server.
- **default_passive** - The IP and the name (as it appears in the **uname –n** command) of the default passive Policy Server.
- **device_uname** port and timeout - The **device_uname** is an Apache handler which returns the device uname. (Used for configuring Heartbeat).
- **ha_manager configuration** - such as ha_manager port and timeout.
- A list of files that are not located under the base directory and are required to be copied to the passive Policy Server. (This will be discussed later in this document.) For each file, we can configure whether it will be copied to the passive device as soon as it is modified using the **inotify** utility (inotify=1), or will be copied every X interval (inotify=0).

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<module id="174" name="HA" hide_local="0" type="module" local="1" active="1" display_name="HA"
    show_module_config="1">
  <init>
      <ha enabled="1"/>
      <default_active ip="192.168.120.166"/>
      <default_passive ip="192.168.120.28"/>
      <virtual_ip ip="192.168.120.187"/>
      <device_uname port="5222" timeout="20"/>
      <ha_manager_configuration connection_timeout="300" ha_manager_port="4589"
    heartbeat_cmd_interval="10" passive_status_interval="10"/>
      <files_to_replicate>
            <file inotify="1" name="/etc/policyserver/.license" permissions="www-data:www-data"/>
            <file inotify="1" name="/etc/shadow" permissions="root:shadow"/>
            <file inotify="0" interval="60" name="/etc/logserver/status.conf" permissions="root:root"/>
            <file inotify="0" interval="60" name="/var/logserver/archive" permissions="root:root"/>
      </files_to_replicate>
 </init>
       </module>
```

The Heartbeat configuration files are located at **/etc/heartbeat**/. Two files should be configured:

1. **Ha.cf** – holds the following:
   - The active and passive names.
   - Heartbeat debug file and debug level.
   - All kinds of Heartbeat configurations (for example autofailback).
   - Port through which both Heartbeats communicate.
2. **haresources** – holds the virtual IP of the HA system. The file is in the format: **[default active device name] [virtual ip]**

   For example: vs-166 192.168.120.185

2.2.2 HA Script

The **/usr/bin/ha** script performs the following:

1. **start -** Configures Heartbeat and Postgres, and starts the Heartbeat process.
2. **stop -** Configures Postgres and stops the Heartbeat process.
3. **restart -** Restarts the Heartbeat process. (performs **stop** and **start**)
4. **status -** Returns **1** if Heartbeat is running, **0** otherwise.
5. **amIactive -** Returns **1** if the device is active, **0** otherwise.
6. **Failover -** Performs a failover.

2.3 Notifier

The passive device is listed in the **devices.xml** file with **device_type** equal to **Management Server**.

The Notifier treats the passive device in the same way as it treats all other scanners, with a few exceptions:

1. The Notifier sends a new flag in the status command telling the passive Manager it is a passive device (isPassive=1).
2. The Notifier gives all policy servers a higher priority on scanners in the order of applying the configuration.
3. When there is a security update or maintenance release, the Notifier does not copy the **ps_debpackages** directory to scanners, though it should be copied to all policy servers.

On commit, the Notifier copies the base directory to the stable directory, and the stable directory to the **conf_ready** directory located at each device. The same happens with the passive device. On failover, when the passive becomes active, the passive will copy the **conf_ready** directory to the base directory so that the Notifier will be able to sync the configuration to the scanners.

The Notifier will get the passive policy server status in the same way as it gets the status of all scanners.

2.4 Replicating Data

The task of replicating data is divided between three utilities.

- **PostgreSQL 9:** Replicates the databases
- **notifier-Manager:** Replicates the files located under /var/policyserver/configuration/base
- **ha_Manager:** Copies all other files

All files are copied using **rsync** to directories located under **/opt/finjan/configuration**. This is because **rsync** has permissions to write only to that directory on a remote device. For this reason the Postgres data directory and the **conf_ready** directory are located at **/opt/finjan/configuration**. Files such as **/etc/logserver/status.conf** are copied to their original location only on failover.

The data that is replicated from the active to the passive policy server contains the following:

1. Databases - policy_server, logs, reports, system_logs
2. All the module configurations and dat files that are located at **/var/policyserver/configuration/base**. This directory also includes the debpackages in case of maintenance releases and hotfixes.
3. Licenses file - **/etc/policyserver/.license** - This file defines the SWG license and is replicated when the file is modified.
4. Shadow file - holds encrypted passwords such as root and administrator passwords. Is replicated when the file is modified.
5. Archive directory -**/var/logserver/archive**. Is replicated every X interval (defined in **HA/module.xml**).
6. LogServer status file - **/etc/logserver/status.conf**. Is replicated every X interval.

2.4.1 PostgreSQL (Postgres)

PostgreSQL is an object-relational database system with a built-in replication feature that replicates all databases in the device.

Note: To use the built-in replication in Postgres, it must be upgraded from PostgreSQL 8.4 to PostgreSQL 9.

Replication is asynchronous but occurs automatically (not on demand), and very close to the time of the changes in the active device. According to the Postgres manual: "*Streaming replication is asynchronous, so there is still a small delay between committing a transaction in the primary and for the changes to become visible in the standby. The delay is however much smaller than with file-based log shipping, typically under one second assuming the standby is powerful enough to keep up with the load.*"

The active databases remain read-write, while the passive databases are read-only.

Before Postgres starts replicating the databases, you must copy all files in the Postgres data directory (**/opt/finjan/configuration/data/postgresql/main**) from the active device to the passive device. Note that this sync can take a long time (depending on the size of the database), but Postgres requires this before starting continuous replication. (For example, copying a 1.5G database from one device to another using **rsync** takes 1m 24s.)

For more information about PostgreSQL replication, see http://wiki.postgresql.org/wiki/Binary_Replication_Tutorial

2.5 Version installation from scratch

When installing a new version, one must first disable the HA, then install the new version on both policy servers. HA can be re-enabled only after both policy servers have the same version installed.

2.6 System Updates

Normally, when you configure automatic update of Scanning Servers with the latest SWG updates, all Scanning Servers are updated at once.

However, the System Updates node lets you choose to update selected scanning servers with the latest Operating System update instead of sending the update to all the scanning servers at the same time. This ensures greater system stability and provides you greater control over the individual scanning servers in your configuration.

This feature is also useful when updating the policy server operating system in a High Availability configuration. In this scenario, some scanning servers can be left untouched, so that if the update fails, the Policy Server will still be able to control the selected scanning servers.

Note: To upgrade to SWG Version 11.0, 11.5, 11.6, 11.7 or 11.8 on a High Availability Setup, refer to the SWG Upgrade Release Notes.

2.6.1 Version Upgrades

Version upgrades are performed the same as version updates. The passive must first be disconnected from the active policy server.

2.6.2 Security Updates

Security updates work the same as configuration updates. The new files are copied to the passive policy server the same way as they are copied and installed at the Managers.

2.6.3 Hotfix / Maintenance Releases

Hotfix and maintenance releases will be copied to the passive policy server the same way as they are copied and installed at the Managers. However, the Notifier copies the directory ps_debpackages to the passive policy server although it is not copied to the scanners.

3 GUI

The passive policy server is added as a device in the Devices screen of the active policy server. It is written to the **devices.xml** file with the same Management Server device type. The GUI verifies that both policy servers are running the same SWG version.

The GUI shows the status of the passive policy server which includes:

Field	Description
Sync Status	whether the passive device is synced to base directory
Connection Status	whether the active is connected to the passive
Replication Status	whether the Postgres is running in replication mode on both devices. (The ha_manager writes this status to the file **/etc/ha_manager/ha_manager_status**)

1. The GUI should enable configuration of the following fields:
   - Passive IP
   - Virtual IP
2. A manual swap between active and passive can be done only by using the Limited shell on the active device by calling **/usr/bin/ha failover**.
3. If the user enters the passive IP at the URL browser, they should be redirected to the virtual IP.

3.1 Status Tab Fields in the High Availability Device IP Window

The following table describes the fields in the Status tab in the Device IP window of the High Availability (secondary) server.

Field	Description
Sync Status	Indicates whether the Device is synchronized with the Policy Server
Connection Status	Indicates if the device is available (Active)
Committing Status	Indicates whether the device is undergoing a Preparing to Commit status, Committing Changes status, or is Stable
Replication Status	Status of the replication
Last Connection Time	Indicates the last time this device was connected to the Policy Server

3.2 Implementing High Availability in SWG

To create an HA system, the user adds a passive policy server to the Management Devices Group.

Note: The Management Console GUI is not accessible from the Passive Policy Server device.

When the change is committed, the policy server:

1. Adds the passive policy server IP to the **devices.xml** file.
2. Configures the file **HA/module.xml**.
3. Enables the role HA in the **Commander/module.xml** file.

   After the changes are committed, the Notifier on the active device sends a **get status** command to the passive device telling it to start listening to the active Notifier. This triggers the Notifier on the active device to send the configuration to the passive Manager which starts Heartbeat and ha_manager processes. The Heartbeat process sets which policy server is active and which is passive. The ha_manager on the active:
4. Tells the passive ha_manager to command the Manager to reload its configuration with the **Commander/module.xml.passive** file. This stops all roles and starts only the roles needed for a passive device.
5. Starts Postgres replication.

To implement High Availability:

1. Select **Administration | System Settings | SWG Devices**.
2. In the Devices tree, right-click the **Management Devices Group** node and choose **Add HA Device**.
3. In the main window, enter the mandatory Device IP, and optionally enter a description. Note that the device type is automatically set to **Passive Policy Server**.
4. Click **Save**.
5. Optionally, specify a virtual device IP, which will automatically route to whichever policy server is active at any given time, as follows:
   a. In the tree pane, select **Management Devices Group**.

      The Management Devices Group window contains only one editable field:

      Virtual IP: Enables you to specify a Virtual IP that will automatically resolve to your currently active policy server device. If you define a virtual IP value, you can use this value for access regardless of whether SWG has failed over to the previously passive policy server device.

   b. Specify a virtual Device IP and click **Save**.
6. To complete implementation of High Availability, including synchronization of the database and configuration files, click **Commit**  .

4 Other Considerations

If High Availability is enabled, you must disable the High Availability Policy Server feature before performing a restore.

Both active and passive policy servers must be synced with the same NTP server.

Ping Node — Ping node detects a situation in which there is a network communication between active and passive policy servers, but no network to scanners. If no ping between the active policy server and the ping node exists, the system will failover and the passive policy server will become active. It is recommended that the IP of the ping node be the default gateway.

The Management Console GUI is not accessible on the Passive Policy Server device.

Warning: When disabling HA, ensure that the Passive Policy Server is connected to the Active Policy Server.

5 Scenarios

5.1 Active Policy Server crashes

5.1.1 Passive Policy Server

The Heartbeat process:

1. Sets the device to active.
2. Starts the virtual IP.

The ha_manager process:

3. Changes its status to active.
4. Runs the **manager-ctl reload** command, which tells the Manager to stop all roles and start them according to the **Commander/module.xml** file.
5. Copies the directory /var/wasp/conf_ready to /var/policyserver/configuration/base
6. If there is a connection to the passive policy server, configures and starts Postgres replication.

If there is no connection to the passive policy server, steps 3-4 occur when the connection is resumed (the ha_manager checks every X seconds if the connection is resumed).

5.1.2 Active Policy Server

The following occurs when the connection between the two policy servers is resumed:

The Heartbeat process:

1. Sets the device to passive.
2. Stops the virtual IP.

The ha_manager process waits for commands from the new active ha_manager process.

5.2 Passive Policy Server Crashes

5.2.1 Passive Policy Server

When a passive policy server comes back up again:

1. The Manager comes up with the passive **Commander/module.xml.passive** configuration (because the manager_passive file exists) and listens to the Notifier at the active (as it did before the crash).
2. The Heartbeat process sets the device to passive.
3. The ha_manager process waits for commands from the active ha_manager process.

5.2.2 Active Policy Server

The ha_manager at the active policy server checks the status of the passive every X seconds. When it discovers the connection to the passive is resumed, it configures and starts Postgres replication.

About Trustwave

Trustwave helps businesses fight cybercrime, protect data and reduce security risk. With cloud and managed security services, integrated technologies and a team of security experts, ethical hackers and researchers, Trustwave enables businesses to transform the way they manage their information security and compliance programs. More than 2.7 million businesses are enrolled in the Trustwave TrustKeeper® cloud platform, through which Trustwave delivers automated, efficient and cost-effective threat, vulnerability and compliance management. Trustwave is a privately held company, headquartered in Chicago, with customers in 96 countries.

For more information, visit https://www.trustwave.com.