Red Hat Satellite 5.6

User Guide

Using and administering Red Hat Satellite

Edition 1

Last Updated: 2017-09-26

Red Hat Satellite 5.6 User Guide

Using and administering Red Hat Satellite

Edition 1

John Ha
Red Hat Engineering Content Services

Lana Brindley
Red Hat Engineering Content Services

Daniel Macpherson
Red Hat Engineering Content Services

Athene Chan
Red Hat Engineering Content Services

David O'Brien
Red Hat Engineering Content Services

Megan Lewis
Red Hat Engineering Content Services

Legal Notice

Abstract

This guide covers the use and administration of Red Hat Satellite. For further information, see the Red Hat Satellite Getting Started Guide and the Red Hat Satellite Client Configuration Guide.

Table of Contents

PREFACE 3

CHAPTER 1. MANAGING USER ACCOUNTS, GROUPS, AND SYSTEMS 4
1.1. MANAGING USER ACCOUNTS 4
1.1.1. Creating and Deleting User Accounts 4
1.1.2. Assigning Roles to User Accounts 6

CHAPTER 2. AUTOMATICALLY SYNCHRONIZING THE RED HAT SATELLITE SERVER REPOSITORY 8

CHAPTER 3. PLANNING FOR DISASTER RECOVERY 9
3.1. BACKING UP A RED HAT SATELLITE SERVER 9
3.2. BACKING UP AN EMBEDDED DATABASE 10
3.2.1. Performing Online Database Backups 10
3.2.1.1. Performing an Online Backup 11
3.2.1.2. Restoring a Database from an Online Backup 11
3.2.2. Performing Offline Database Backups 11
3.2.2.1. Performing an Offline Backup 12
3.2.2.2. Verifying the Backup 12
3.2.2.3. Restoring the Database 13
3.3. CLONING A RED HAT SATELLITE WITH AN EMBEDDED DATABASE 13
3.4. CREATING REDUNDANT SATELLITES WITH EXTERNAL DATABASES 14
3.5. AUTOMATING SATELLITE DATABASE BACKUPS 15

CHAPTER 4. CLONING SOFTWARE CHANNELS AND ERRATA 17
4.1. FEATURES 17
4.2. EXAMPLE USAGE 17

CHAPTER 5. MONITORING 18
5.1. MONITORING TABLESPACES 18
5.2. MONITORING RED HAT SATELLITE SERVER PROCESSES 18

CHAPTER 6. MAINTAINING SYSTEM SECURITY USING OPENSCAP 19
6.1. OPENSCAP FEATURES 19
6.2. OPENSCAP PREREQUISITES 19
6.3. RED HAT SATELLITE PREREQUISITES FOR USING OPENSCAP 20
6.4. PERFORMING AUDIT SCANS 20
6.4.1. Using the Web Interface to Perform Audit Scans 20
6.4.2. Using the API to Perform Audit Scans 21
6.4.3. Viewing the Results of SCAP Audits 22
6.5. OPENSCAP SATELLITE WEB INTERFACE 22
6.5.1. OpenSCAP Scans Page 22
6.5.1.1. All Scans 22
6.5.1.2. XCCDF Diff 23
6.5.1.3. Advanced Search 23
6.5.2. Systems Audit Page 23
6.5.2.1. List Scans 24
6.5.2.2. Scan Details 25
6.5.2.3. Schedule Page 25

CHAPTER 7. REPORTING CLIENT SOFTWARE FAILURES 27
7.1. VIEWING SOFTWARE FAILURES FOR A SINGLE CLIENT 27
7.2. GROUPING SIMILAR SOFTWARE FAILURES 27
7.3. CHANGING ORGANIZATION-WIDE SETTINGS FOR SOFTWARE FAILURE REPORTS 27

7.4. LOG FILES OF SOFTWARE FAILURES 28

CHAPTER 8. GENERATING RED HAT SATELLITE REPORTS 29

CHAPTER 9. SCHEDULING RED HAT SATELLITE ADMINISTRATIVE TASKS 31

9.1. SCHEDULING A RUN 32

APPENDIX A. REVISION HISTORY 34

PREFACE

Red Hat Network provides system-level support and management of Red Hat systems and networks. It brings together the tools, services, and information repositories needed to maximize the reliability, security, and performance of Red Hat systems. To use Red Hat Network, system administrators register software and hardware profiles, known as *System Profiles,* of their client systems with Red Hat Network. When a client system requests package updates, only the applicable packages for the client are returned.

Red Hat Satellite allows organizations to use the benefits of Red Hat Network without having to provide public Internet access to their servers or other client systems. System profiles are stored locally on the Satellite server. The Satellite website is served from a local web server and is only accessible to systems that can reach the Satellite. All package management tasks, including errata updates, are performed through the Satellite server.

Red Hat Satellite provides a solution for organizations that require absolute control over and privacy of the maintenance and package deployment of their servers. It allows Red Hat Network customers the greatest flexibility and power in keeping systems secure and updated. Modules can be added to the Satellite server to provide extra functionality.

This document provides guidance on essential operations for running Red Hat Satellite.

CHAPTER 1. MANAGING USER ACCOUNTS, GROUPS, AND SYSTEMS

1.1. MANAGING USER ACCOUNTS

1.1.1. Creating and Deleting User Accounts

The **Users** page on the Red Hat Satellite web server provides suitable tools to manage Satellite users. You can use this page to create, delete, activate, and deactivate user accounts, as well as assign roles and their associated permissions.

Creating User Accounts

Before Satellite users can register with the Satellite server to request product updates or to perform other maintenance, they need a suitable user account. Only certain Satellite **`Administrators`** can create user accounts.

Procedure 1.1. Creating User Accounts

To create a user account:

1. Navigate to the Satellite web server page, and click the **Users** tab on the navigation bar.

2. On the right side of the page, click **`create new user`** to open the **`Create User`** page.

3. Complete all of the required fields.

   NOTE

   The login value must be at least five characters long, and may only contain alphanumeric, hyphen, underscore, comma, period, and commercial at (@) characters.

4. Click **`Create Login`** to create the new user. An email will be sent to the user, using the address specified during creation, to inform them of the new account details. This will include the password in plain text.

5. When the account has been successfully created, you will be redirected to the **`User List`** page. To change permissions and set options for the new user, select their name from the displayed list to display the **`User Details`** page, and navigate to the appropriate tabs to make your changes.

Deleting User Accounts

Only Satellite **`Administrators`** can delete user accounts. Deleted accounts cannot be used to log in to the Satellite server interface, or to schedule actions.

WARNING

You cannot retrieve deleted user accounts. Consider deactivating the user account before deleting it, in order to assess the consequences.

Procedure 1.2. Deleting User Accounts

To delete a user account:

1. Navigate to the Satellite web server page, and click the **Users** tab on the navigation bar.
2. Click the user name of the account that you want to delete from the **`Username`** list. The **`User Details`** page displays.
3. Ensure that the user account is not a Satellite administrator.

   If the user is a Satellite administrator, clear the associated check box, and click **`Submit`**.

   If the user is not a Satellite administrator, continue to the next step.
4. Click **`Delete User`**. The **`Confirm User Deletion`** page displays.
5. Ensure that you want to completely delete this user account, and click **`Delete User`**.

After the user account has been successfully deleted, you will be returned to the **`Active Users`** page. The user's name will no longer appear in the **`Active Users`** list.

Procedure 1.3. Activating and Deactivating Users

User accounts are automatically activated when they are created. They can be deactivated by administrators, or users can deactivate their own accounts. Deactivated user accounts cannot log in to the Satellite server interface, or schedule any actions. Any actions that were scheduled before the account was deactivated remain in the action queue until they are completed. Deactivated user accounts can only be reactivated by administrators.

NOTE

Administrator accounts can only be deactivated after the Administrator role has been removed from the account.

To deactivate a user account:

1. Select the user's name from the list in the **`Users`** tab, to display the **`User Details`** page.
2. Check to see if the user is a Satellite administrator.

   If the user is a Satellite administrator, uncheck the box next to that role, and click **`Submit`**.

   If the user is not a Satellite administrator, continue to the next step.

3. Click **`Deactivate User`**.

   You will be asked to confirm this action, by clicking it again. Check the details, and then click **`Deactivate User`** again to confirm.

4. Once the account has been successfully deactivated, the user's name will not appear in the **`Active Users`** list. Click the **`Deactivated`** link from the **`User List`** menu to view deactivated user accounts.

5. To reactivate the user account, view the **`Deactivated`** list, check the box next to the user to be reactivate, and click **`Reactivate`**.

1.1.2. Assigning Roles to User Accounts

User accounts can be managed through the **`Users`** tab at the top of the Satellite Server navigation bar. To change permissions and set options for a user, select their name from the displayed list to display the **`User Details`** page, and navigate to the appropriate tabs to make your changes. Modify account details by making the changes and clicking **`Submit`**.

User Roles

User roles are used to delegate responsibilities to user accounts. Each user role has a different level of responsibility and access.

To assign a user a new role, select the appropriate checkbox on the **`User Details`** page. Modify roles by making the changes and clicking **`Submit`**.

The user roles to choose from are

`Satellite Administrator`

A special role for Satellite administrative tasks such as creating organizations, managing subscriptions, and configuring global Satellite Server settings.

This role cannot be assigned on the **`User Details`** page. A user that already has the Satellite Server administrator role can assign the role to another user by going to **Admin** → **Users**.

`Organization Administrator`

Performs management functions such as managing users, systems, and channels within the context of their organization. Organization administrators are automatically granted administration access to all other roles, which are signified by the checkboxes for the other roles being selected and grayed-out.

`Activation Key Administrator`

Performs activation key functions for such as creating, modifying, and deleting keys within the account.

`Channel Administrator`

Provides complete access to the software channels and related associations within the organization. Performs functions such as making channels globally subscribable, and creating new channels, and managing the packages within channels.

`Configuration Administrator`

Has complete access to the configuration channels and related associations within the organization. Also has complete access to the kickstart profiles and associated items within the organization. Performs kickstart profile, channel and file management configuration functions in the organization.

`Monitoring Administrator`

Performs scheduling of probes and oversight of other monitoring infrastructure. This role is available only on Satellite Servers with monitoring enabled.

`System Group Administrator`

This role has complete authority over the systems and system groups to which it is granted access. Performs administrative functions such as creating new system groups, deleting assigned system groups, adding systems to groups, and managing user access to groups.

Satellite administrators can remove Satellite administrator rights from user accounts, including their own, but there must always be at least one Satellite administrator.

CHAPTER 2. AUTOMATICALLY SYNCHRONIZING THE RED HAT SATELLITE SERVER REPOSITORY

Manually synchronizing the Red Hat Satellite server repository with Red Hat Network can be an arduous task. The synchronization can be automated to occur randomly in a designated off peak window for best performance. You can use the **cron** utility to effectively automate synchronization.

Procedure 2.1. To Use the cron Utility to Automate Synchronization:

1. Switch to the root user, and run the following command to open the **crontab** in a text editor:

   ```
   # crontab -e
   ```

2. Create a suitable job definition to schedule the synchronization. To create a random synchronization time, use the following entry:

   ```
   0 1 * * * perl -le 'sleep rand 9000' && satellite-sync --email
   >/dev/null 2>1
   ```

   This entry runs the synchronization job randomly between 01:00 and 03:30, and discards ***stdout*** and ***stderr*** messages from the **cron** utility. This prevents duplicating messages from the **satellite-sync** command. Other options can be included as needed. See the **crontab** manual page **man crontab** for more information.

3. Exit the text editor to save the updated **crontab** file. The new rules take effect immediately.

NOTE

The **crontab** file opens in vi by default. To change this behavior, change the **EDITOR** variable to the name of the text editor you prefer.

CHAPTER 3. PLANNING FOR DISASTER RECOVERY

This chapter describes the recommended methods for backing up, verifying, and restoring Red Hat Satellite and embedded databases. If you are using an external database, consult your organization's database administrator. If you are using an embedded database, see Section 3.2, “Backing up an Embedded Database” for a complete description of this process and the options available.

You should create backups either nightly or weekly, depending on the amount of data being stored, and how much data can potentially be lost in the case of a system outage.

If you plan on performing offline, or "cold", backups, Red Hat recommends that you schedule these backups to occur during scheduled Satellite Server maintenance outages, because all website and client connection services will be unavailable during the backup. Satellite 5.6 and later include online, or "hot", backup functionality; it is not necessary to perform offline backups.

3.1. BACKING UP A RED HAT SATELLITE SERVER

Several methods exist for backing up your Red Hat Satellite system. The following methods are those recommended by Red Hat.

Minimal Backup

Red Hat recommends that you back up at least the following files and directories:

- **`/var/lib/pgsql/`**: Embedded database only.
- **`/etc/sysconfig/rhn/`**
- **`/etc/rhn/`**
- **`/etc/sudoers`**
- **`/var/www/html/pub/`**
- **`/var/satellite/redhat/[0-9]*/`** (This is the location of any custom RPMs)
- **`/root/.gnupg/`**
- **`/root/ssl-build/`**
- **`/etc/dhcp.conf`**
- **`/etc/httpd`**
- **`/tftpboot/`**
- **`/var/lib/cobbler/`**
- **`/var/lib/rhn/kickstarts/`**
- **`/var/www/cobbler`**
- **`/var/lib/nocpulse/`**

If possible, back up **`/var/satellite/`** as well. In case of failure, this will save lengthy download times. The **`/var/satellite/`** directory (specifically **`/var/satellite/redhat/NULL/`**) is primarily a duplicate of Red Hat's RPM repository, and can be regenerated using the **satellite-sync** command. Red Hat recommends that the entire **/var/satellite/** tree be backed up. In the case of disconnected satellites, **/var/satellite/** *must* be backed up.

Backing up only these files and directories does have some drawbacks. As part of the failure recovery process, you need to:

- Reinstall the Red Hat Satellite ISO RPMs.
- Reregister the server.
- Use the **satellite-sync** command to resynchronize Red Hat packages.
- Reinstall the **/root/ssl-build/rhn-org-httpd-ssl-key-pair-*MACHINE_NAME-VER-REL*.noarch.rpm** file.

Backup without Reregistration

Another method is to back up all of the files and directories mentioned above but reinstall the Satellite server without reregistering it. During the installation, cancel or skip the Red Hat Network registration and SSL certificate generation sections.

Full Machine Backup

The final and most comprehensive method is to back up the entire machine. This saves download and reinstallation time but requires additional disk space and back-up time.

IMPORTANT

Regardless of the back-up method used, when you restore the Satellite server from a backup, you must run the following command to schedule the recreation of search indexes the next time the **rhn-search** service is started:

```
# service rhn-search cleanindex
```

3.2. BACKING UP AN EMBEDDED DATABASE

Red Hat Satellite provides a specialized command-line utility to assist with embedded database management tasks. The **db-control** command provides features to create, verify, and restore backups, to obtain database status information and to restart the database when necessary. See the **db-control** manual page (**man db-control**) for a full listing of the features available.

The following sections are demonstrate how to create, verify, and restore Red Hat Satellite embedded and managed databases.

3.2.1. Performing Online Database Backups

Red Hat Satellite Server 5.6 introduces new functionality that enables online backups of your database, without the need to stop the Satellite Server. Additions to the existing **db-control** command make this functionality possible.

Three new options have been added to the **db-control** command:

- **online-backup *FILENAME*:** Performs an online backup of the Satellite database (embedded PostgreSQL only).

- **`reset-password`**: Resets the user password and unlocks the account.
- **`restore`** ***`DIRECTORY`*** **`|`** ***`FILENAME`***: Restores the database from either:
  - An offline backup taken by **`db-control backup`** and saved in the *DIRECTORY* directory. The database must be stopped for both the **`backup`** and **`restore`** operations in order to run successfully.
  - An online backup taken by **`db-control online-backup`** and saved as *FILENAME*. The database itself must be running for both the **`online-backup`** and **`restore`** operations in order to run successfully, but all other Satellite services must be stopped.

3.2.1.1. Performing an Online Backup

To create an online backup of an embedded Red Hat Satellite 5.6 server database, change to the root user, and run the following command. Replace the ***`FILENAME`*** option with the full path to the backup file that you want to create. This location needs to be writable by the PostgreSQL user:

```
# db-control online-backup FILENAME
```

NOTE

There is no need to stop either the database or the Satellite services to perform an online backup.

3.2.1.2. Restoring a Database from an Online Backup

Use the **`db-control restore`** ***`FILENAME`*** command to restore an embedded database from a backup created using the **`db-control online-backup`** command. Before you restore a database, you need to shut down all Satellite services except the database itself.

Procedure 3.1. To Restore a Database from an Online Backup:

1. Change to the root user, and run the following command to stop all Satellite services except the database:

   ```
   # rhn-satellite stop --exclude=postgresql
   ```

2. Run the following command to restore the database. Replace the ***`FILENAME`*** option with the full path to the backup file created with the **`db-control online-backup`** command:

   ```
   # db-control restore FILENAME
   ```

3. After the restoration is complete, run the following command to restart the database and all related services:

   ```
   # rhn-satellite start
   ```

3.2.2. Performing Offline Database Backups

Red Hat Satellite Server 5.6 provides the ability to perform online backup and restore operations. Red Hat recommends that you continue to perform offline backups during monthly or quarterly maintenance windows.

3.2.2.1. Performing an Offline Backup

The following procedure describes how to back up an embedded Red Hat Satellite server database.

Procedure 3.2. To Create an Offline Backup:

1. Change to the root user, and run the following command to stop the Satellite server:

   ```
   # rhn-satellite stop
   ```

2. Run the following command to create the backup:

   ```
   # db-control backup DIRECTORY
   ```

   Replace *DIRECTORY* with the absolute path to the location where you want to store your database backup. This process will take several minutes.

3. When the backup is complete, run the following command to restart the Satellite server:

   ```
   # rhn-satellite start
   ```

4. Copy the backup to another system using **rsync** or another file-transfer utility. Red Hat strongly recommends scheduling the backup process automatically using cron jobs. For instance, back up the system at 03:00 and then copy the backup to the separate repository (partition, disk, or system) at 06:00.

3.2.2.2. Verifying the Backup

Backing up the embedded database is useful only if you can ensure the integrity of the resulting backup. There are two approaches to this integrity check; you can *examine* the backup to check its time stamp and identify any missing files, or you can *verify* the backup, which involves a more thorough inspection and validation of the md5sum of each file in the backup. The first approach is quicker, but the second provides more thorough validation.

To examine the backup, run the following command as root:

```
# db-control examine DIRECTORY
```

To verify the backup, run the following command as root:

```
# db-control verify DIRECTORY
```

If the verification is successful, you can safely restore the database from the *DIRECTORY* directory.

NOTE

Users of external databases should also perform periodic backups. Consult your external database administrator for more information about supported backup procedures.

3.2.2.3. Restoring the Database

Use the **db-control restore** command to restore embedded databases from backup. Before you attempt to restore a database, you need to shut down the database and any related services.

Procedure 3.3. To Restore an Embedded Database from a Backup:

1. Run the following command to stop all of the Red Hat Satellite services:

   ```
   # rhn-satellite stop
   ```

2. Run the following command, including the directory containing the backup, to begin the restoration. Ensure that you replace *directory* with the absolute path to the location that contains the backup. This process will verify the contents of the backup before restoring the database. The process will take several minutes.

   ```
   # db-control restore directory
   ```

   This not only restores the embedded database but first verifies the contents of the backup directory using checksums.

3. After the restoration is complete, restart the database and related services:

   ```
   # rhn-satellite start
   ```

4. Regardless of whether you are backing up an external or embedded database, when the database is restored from a backup, you should schedule the restoration of search indexes the next time the **rhn-search** service is started:

   ```
   # service rhn-search cleanindex
   ```

3.3. CLONING A RED HAT SATELLITE WITH AN EMBEDDED DATABASE

You can limit outages caused by hardware or other failures by cloning the Red Hat Satellite server with an embedded database in its entirety. The cloned server can be prepared for use if the primary server fails.

Procedure 3.4. To Clone a Satellite Server with an Embedded Database:

1. Install Red Hat Satellite with an embedded database on a base install of Red Hat Enterprise Linux on a separate machine. That is, a machine separate from your primary Red Hat Satellite server. Omit the SSL Certificate generation step.

2. Back up the primary server's database daily using the commands described in Section 3.2.2.1, “Performing an Offline Backup”. This ensures that only changes made the day of the failure are lost.

3. Establish a mechanism to copy the backup to the secondary server. Keep these repositories synchronized using a file transfer program such as **rsync**. Copying is not necessary if using a *Storage Area Network (SAN).*

4. Use the **db-control restore** command to import duplicate data.

5. If the primary server fails, transfer the SSL key pair RPM package in **/root/ssl-build** from the primary to the secondary server, and install that package. This ensures that Red Hat Satellite clients can authenticate with and securely connect to the secondary server.

6. Update your DNS to reference the secondary server, or configure your load balancer appropriately.

3.4. CREATING REDUNDANT SATELLITES WITH EXTERNAL DATABASES

In keeping with the cloning option available to Red Hat Satellite with an embedded database, you can limit outages on Satellite servers with external databases by preparing redundant Satellite servers. Unlike clones, you can run redundant Satellite servers with external databases in either *active* or *standby* mode. This is entirely up to your network topology and is independent of the steps listed here.

IMPORTANT

Before you begin the following procedure, prepare the external database for failover using suitable recommendations for building a fault-tolerant database. Consult your database administrator.

Procedure 3.5. To Create a Redundant Satellite with an External Database:

1. Install Red Hat Satellite on a separate machine, but omit the database configuration, database schema, SSL certificate, and bootstrap script generation steps. Include the same Red Hat Network account and database connection information provided during the initial Satellite installation.

2. Register the new Satellite server. See the Red Hat Satellite *Installation Guide* for more information.

3. If your original SSL certificate does not take your high-availability solution into account, create a new one with a more appropriate **Common Name** value (see *The SSL Maintenance Tool* in the Red Hat Satellite *Client Configuration Guide*). In this case, generate a new bootstrap script (as defined in *Generating Bootstrap Scripts* in the *Red Hat Satellite Client Configuration Guide*) that captures this new value. Ensure the **Common Name** value represents the combined Satellite solution, not a single machine's host name.

4. After installation, copy the following files from the primary server to the secondary:

   - **/etc/rhn/rhn.conf**
   - **/etc/tnsnames.ora** (Oracle database only.)

5. Copy the server-side SSL certificate RPMs from the primary server and install them on the secondary server.

   If, during the installation process, you generated a new SSL certificate that included a new Common Name value, copy the SSL certificate RPMs from the secondary to the primary server and redistribute the client-side certificate. If you also created another bootstrap script, use it to install the certificate on all client systems.

6. - If you created a new bootstrap script, copy the contents of **/var/www/html/pub/bootstrap/** to the primary server.

- If you did not create a new bootstrap script, copy the contents of **/var/www/html/pub/bootstrap/** from the primary server to the secondary server.

7. Run the following command on the secondary server to stop the **Red Hat Network Task Engine** service:

```
# service taskomatic stop
```

You can use custom scripting or other means to establish automatic start-up/failover of the **Red Hat Network Task Engine** on the secondary server. Regardless, you need to ensure that it starts in the event of a failure.

8. Share channel package data (by default located in **/var/satellite**) and cache data (by default located in **/var/cache/rhn**) between the primary and secondary servers over some type of networked storage device. This eliminates data replication and ensures a consistent store of data for each server.

9. Make the various servers available on your network using a suitable Common Name and a method that suits your infrastructure. Options include round-robin DNS, a network load balancer, and a reverse-proxy setup.

3.5. AUTOMATING SATELLITE DATABASE BACKUPS

Backup tasks can be automated so that they occur in non-peak times, such as the late evening or early morning. This also ensures they are performed regularly, and are not forgotten. The most effective way to automate backups is using **cron**.

Procedure 3.6. To Automate Satellite Server Database Backups:

Create a new file called **backup-db.sh** containing the following script. This script will stop the satellite, perform a database backup, and restart the satellite:

```
#!/bin/bash
{
/usr/sbin/rhn-satellite stop
d=db-backup-$(date "+%F");
mkdir -p /tmp/$d;
db-control backup /tmp/$d
/usr/sbin/rhn-satellite start
} &> /dev/null
```

1. Create a new file called **move-files.sh** containing the following script. This script will use **rsync** to move the backup files to a directory to be stored:

```
#!/bin/bash
rsync -avz /tmp/db-backup-$(date "+%F") <destination> &> /dev/null
```

Replace *<destination>* with the path to the backup directory.

Alternatively, use the following script to achieve the same goal:

```
#!/bin/bash
scp -r /tmp/db-backup-$(date "+%F") <destination> &> /dev/null
```

2. Switch to the root user, and open the **crontab** file in a text editor:

   ```
   # crontab -e
   ```

   NOTE

   The **crontab** file opens in vi by default. To change this behavior, change the **EDITOR** variable to the name of the text editor you prefer.

3. Create a suitable job definition to schedule the backup scripts to run:

   ```
   0 3 * * * backup-db.sh
   0 6 * * * move-files.sh
   ```

   This **crontab** entry will run the backup at 03:00, and transfer the backup files at 06:00. Other options can be included as needed. You can also include a clean up script to remove older backup directories and prevent the backup storage from filling up.

4. Exit the editor to save the **crontab** file. The new rules take effect immediately.

CHAPTER 4. CLONING SOFTWARE CHANNELS AND ERRATA

Use the **spacewalk-clone-by-date** command to create custom cloned Red Hat Enterprise Linux channels based on the date an erratum was made available to the Red Hat Enterprise Linux system.

4.1. FEATURES

The following features are available with **spacewalk-clone-by-date**:

- Cloning the channel errata and associated package states as they were on a specific date
- Automating the cloning by scripts and template files
- Removing or blocking packages from channels
- Resolving package dependencies within the parent and child channels
- Filtering and acting on specific errata while ignoring others. For example, acting only on security errata and ignoring bugfixes and enhancements.

NOTE

You need to run the **spacewalk-clone-by-date** command as the *root user* and the **username** needs to be either an Organizational Administrator or Channel Administrator.

4.2. EXAMPLE USAGE

The example below clones the **rhel-i386-server-5** channel errata as it is on January 1st, 2012, into the channel named my-clone-RHEL-5.

```
# spacewalk-clone-by-date --username=your_username --
password=your_password --server=satellite_server_url --channels=rhel-i386-
server-5 my-clone-RHEL-5 --to_date=2012-01-01
```

IMPORTANT

Ensure the cloned channel name contains no spaces.

The example below will only clone security errata released on or before January 1st, 2012, ignoring any kernel updates or vim-extended packages. The command will also run the cloning process in the background on the Satellite.

```
# spacewalk-clone-by-date --username=your_username --
password=your_password --server=satellite_server_url --channels=rhel-i386-
server-5 my-clone-RHEL-5 --to_date=2012-01-01 --security_only --background
--blacklist=kernel,vim-extended --assumeyes
```

See the manual page for **spacewalk-clone-by-date** for more information about the available options and how to use them.

CHAPTER 5. MONITORING

Red Hat Satellite Server contains many different components, many of which can be monitored. This chapter describes how to monitor different areas of the system.

5.1. MONITORING TABLESPACES

Beginning with **Satellite Server 5.6**, the default database configuration is an embedded PostgreSQL database. PostgreSQL does not require that you manually allocate more space for the **Satellite** schema as it grows; PostgreSQL expands the schema of its own accord, provided that free disk space is available.

To check the database's size, run the **`db-control report`** command as the **`root`** user.

To check the size of each table in the database, run the **`db-control tablesizes`** command as the **`root`** user.

If you are using an *external* Oracle database, consult your database administrator for information about how to monitor tablespaces.

5.2. MONITORING RED HAT SATELLITE SERVER PROCESSES

Use the **`rhn-satellite status`** command to verify that all services related to the **Satellite Server** are running:

```
# rhn-satellite status
```

CHAPTER 6. MAINTAINING SYSTEM SECURITY USING OPENSCAP

The *Security Certification and Authorization Package (SCAP)*is a standardized compliance checking solution for enterprise-level Linux infrastructures. It is a line of specifications maintained by the National Institute of Standards and Technology (NIST) for maintaining system security for enterprise systems.

Red Hat Satellite Server 5.5 and later use **OpenSCAP** to implement the SCAP specifications. **OpenSCAP** is an auditing tool that utilizes the Extensible Configuration Checklist Description Format (XCCDF). XCCDF is a standard way of expressing checklist content and defines security checklists. It also combines with other specifications such as Common Platform Enumeration (CPE), Common Configuration Enumeration (CCE), and Open Vulnerability and Assessment Language (OVAL), to create a SCAP-expressed checklist that can be processed by SCAP-validated products.

6.1. OPENSCAP FEATURES

OpenSCAP verifies the presence of patches by using content produced by the Red Hat Security Response Team (SRT), it checks system security configuration settings, and examines systems for signs of compromise by using rules based on standards and specifications.

6.2. OPENSCAP PREREQUISITES

To effectively use OpenSCAP, the following must be available:

- A tool to verify that a system conforms to a standard.

  Satellite Server 5.5 and later use OpenSCAP as an auditing feature. This allows you to use the web interface to schedule and view compliance scans for any system.

- SCAP content.

  You can generate your own SCAP content if you have an understanding of at least XCCDF or OVAL. XCCDF content is also frequently published online under open source licenses, and you can customize this content to suit your needs instead.

  > **NOTE**
  >
  > Red Hat supports the use of templates to evaluate your systems. However, custom content authoring of these templates is not supported.

  Some examples of bodies that publish XCCDF content are:

  - The United States Government Configuration Baseline (USGCB) : Official SCAP content for desktops within federal agencies that has been developed at NIST in collaboration with Red Hat, Inc. and the United States Department of Defense (DoD) using OVAL.
  - Community-provided content:
    - SCAP Security Guide: Active community-run content that sources from the USGCB requirements and widely-accepted policies and contains profiles for desktop, server, and FTP server. Suitable for **Red Hat Enterprise Linux 6** and **JBoss Enterprise Application Server 5.**

    - OpenSCAP Content for **Red Hat Enterprise Linux 6:** The openscap-content package from the **Red Hat Enterprise Linux 6** Optional Channel also provides default content guidance by means of a template.

SCAP was created to provide a standardized approach to maintaining system security, and the standards that are used will therefore continually change to meet the needs of the community and enterprise businesses. New specifications are governed by NIST's SCAP Release cycle in order to provide a consistent and repeatable revision workflow.

6.3. RED HAT SATELLITE PREREQUISITES FOR USING OPENSCAP

The following sections describe the prerequisites for using OpenSCAP on **Red Hat Satellite Servers** and **Satellite Clients.**

Package Requirements

- **Satellite Server: Satellite 5.5** or later.
- **Satellite Client:** spacewalk-oscap package (available from the Red Hat Network Tools Child Channel).

Entitlement Requirements

A Management entitlement is required for scheduling scans.

Other Requirements

Satellite Client: Distribution of the XCCDF content to all client machines.

You can distribute the XCCDF content to client machines using any of the following methods:

- Traditional methods, such as CD, USB, NFS, SCP, FTP.
- Satellite scripts.
- RPM packages.

  Custom RPMs are the recommended way to distribute SCAP content to other machines. RPM packages can be signed and verified to ensure their integrity. Installation, removal, and verification of RPM packages can be managed from the user interface.

6.4. PERFORMING AUDIT SCANS

OpenSCAP integration in **Red Hat Satellite Server** provides the ability to perform audit scans on client systems. This section describes the methods available for performing these scans.

6.4.1. Using the Web Interface to Perform Audit Scans

This section describes how to use the **Satellite** web interface to perform audit scans.

Procedure 6.1. To Perform an Audit Scan Using the Web Interface:

1. Log in to the Satellite web interface.
2. Click **Systems** → **system_name.**

3. Click **Audit** → **Schedule.**

4. Complete the ***Schedule New XCCDF Scan*** page. See Section 6.5.2.3, “Schedule Page” for information about the fields on this page.

> **WARNING**
>
> The XCCDF content is validated before it is run on the remote system. Specifying invalid command-line arguments can cause **spacewalk-oscap** to fail to validate or run. Due to security concerns the **oscap xccdf eval** command only accepts a limited set of parameters.

> **NOTE**
>
> You can run the **rhn_check** command to ensure that the action is being picked up by the client system.
>
> ```
> # rhn_check -vv
> ```
>
> If **rhnsd** or **osad** are running on the client system, the action will be picked up by these services. To check if they are running, run *one* of the following commands:
>
> ```
> # service rhnsd start
> OR
> # service osad start
> ```

To view the results of the scan, see Section 6.4.3, “Viewing the Results of SCAP Audits”.

6.4.2. Using the API to Perform Audit Scans

This section describes how to use the **Satellite** API to perform audit scans.

Procedure 6.2. To Perform an Audit Scan Using the API:

1. Choose an existing script or create a script for scheduling a system scan through **system.scap.scheduleXccdfScan**, the front-end API, for example:

```
#!/usr/bin/python
client = xmlrpclib.Server('https://satellite.example.com/rpc/api')
key = client.auth.login('username', 'password')
client.system.scap.scheduleXccdfScan(key, 1000010001,
    '/usr/local/share/scap/usgcb-rhel5desktop-xccdf.xml',
    '--profile united_states_government_configuration_baseline')
```

Where:

- 1000010001 is the ***system ID (sid)***.

   - `/usr/local/share/scap/usgcb-rhel5desktop-xccdf.xml` is the path to the content location on the client system. In this case, it assumes USGCB content in the `/usr/local/share/scap` directory.

   - **`--profile united_states_government_configuration_baseline`** is an additional argument to the **`oscap`** command. In this case, it is using the USGCB.

2. Run the script on the command-line interface of any system. The system needs the appropriate Python and XML-RPC libraries installed.

NOTE

You can run the **`rhn_check`** command to ensure that the action is being picked up by the client system.

```
# rhn_check -vv
```

If **`rhnsd`** or **`osad`** are running on the client system, the action will be picked up by these services. To check if they are running, run *one* of the following commands:

```
# service rhnsd start
OR
# service osad start
```

6.4.3. Viewing the Results of SCAP Audits

There are three methods of viewing the results of finished scans:

- Using the web interface. After the scan has finished, the results are available on the **Audit** page of specific system. See Section 6.5, “OpenSCAP Satellite Web Interface”.

- Using the API functions in handler **`system.scap`**.

- Using the **`spacewalk-report`** command, as follows:

```
# spacewalk-report system-history-scap
# spacewalk-report scap-scan
# spacewalk-report scap-scan-results
```

6.5. OPENSCAP SATELLITE WEB INTERFACE

The following sections describe the pages in the Red Hat Satellite web interface that provide access to OpenSCAP and its features.

6.5.1. OpenSCAP Scans Page

Click the **`Audit`** tab on the top navigation bar to display the OpenSCAP Scans page. This is the "overview" page for all OpenSCAP functionality in Satellite Server. Use this page to view, search for, and compare completed scans.

6.5.1.1. All Scans

The **`All Scans`** page is the default page that appears on the **`Audit`** tab. This page displays all the completed OpenSCAP scans that the viewer has permission to see. Permissions for scans are derived from system permissions.

For each scan, the following information is displayed:

- System: the system that was scanned.
- XCCDF Profile: the evaluated profile.
- Completed: the time the scan was completed.
- Satisfied: the number of rules that were satisfied. A rule is considered to be Satisfied if the result of the evaluation is either Pass or Fixed.
- Dissatisfied: the number of rules that were not satisfied. A rule is considered to be Dissatisfied if the result of the evaluation is Fail.
- Unknown: the number of rules that failed to evaluate. A rule is considered to be Unknown if the result of the evaluation is Error, Unknown or Not Checked.

The evaluation of XCCDF rules may also return status results such as **`Informational, Not Applicable,`** or **`not Selected.`** In such cases, the given rule is not included in the statistics on this page. See **System Details** → **Audit** for information about these types of results.

6.5.1.2. XCCDF Diff

XCCDF Diff is an application which visualizes the comparison of two XCCDF scans. It shows metadata for two scans as well as the lists of results.

Click the appropriate icon on the **`List Scans`** page to access the **`diff`** output of similar scans. Alternatively, or you can specify the ID of arbitrary scans.

Items that show up in only one of the compared scans are considered to be "varying". Varying items are always highlighted in beige. There are three possible comparison modes: **`Full Comparison`** which shows all the scan items, **`Only Changed Items`** which shows items that have changed, and finally **`Only Invariant Items`** which shows unchanged or similar items.

6.5.1.3. Advanced Search

Use the Advanced Search page to search through your scans according to specified criteria, including:

- Rule results.
- Targeted machine.
- Time frame of the scan.

The search either returns a list of results or a list of scans which are included in the results.

6.5.2. Systems Audit Page

Use the **`Systems Audit`** page to schedule and view compliance scans for a particular system. Scans are performed by the OpenSCAP tool, which implements NIST's standard **`Security Content Automation Protocol (SCAP).`** Before you scan a system, ensure that the SCAP content is prepared and all prerequisites are met.

To display the `Systems Audit` page, click **Systems** → ***system_name*** → **Audit**.

6.5.2.1. List Scans

This page displays a summary of all scans completed on the selected system. The following columns are displayed:

Table 6.1. OpenSCAP Scan Labels

Column Label	Definition
XCCDF Test Result	The scan test result name. This is also a link to the detailed results of the scan.
Completed	The exact time the scan finished.
Compliance	The unweighted pass:fail ratio of compliance based on the standard that was used.
P	The number of checks that passed.
F	The number of checks that failed.
E	The number of errors that occurred during the scan.
U	Unknown
N	Not applicable to the machine.
K	Not checked.
S	Not selected.
I	Informational
X	Fixed
Total	Total number of checks.

Each entry begins with an icon indicating the results of a comparison to a previous similar scan. The icons indicate the following:

- ✓ No difference between the compared scans.
- ⚠ Arbitrary differences between the compared scans.
- ✖ Major differences between the compared scans. Either there are more failures than the previous scan or less passes.

- ◆ No comparable scan was found, and therefore no comparison was made.

6.5.2.2. Scan Details

The **Scan Details** page contains the results of a single scan. This page is divided into two sections:

Details of the XCCDF Scan

This section displays various details about the scan, including:

- ***File System Path:*** The path to the XCCDF file used for the scan.
- ***Command-line Arguments:*** Any additional command-line arguments that were used.
- ***Profile Identifier:*** The profile identifier used for the scan.
- ***Profile Title:*** The title of the profile used for the scan.
- ***Scan's Error output:*** Any errors encountered during the scan.

XCCDF Rule Results

The rule results provide the full list of XCCDF rule identifiers, identifying tags, and the result for each of these rule checks. This list can be filtered by a specific result.

6.5.2.3. Schedule Page

Use the **Schedule New XCCDF Scan** page to schedule new scans for specific machines. Scans occur at the system's next scheduled check-in that occurs *after* the date and time specified.

The following fields can be configured:

- **Command-line Arguments:** Optional arguments to the **oscap** command, either:
  - **--profile *PROFILE***: Specifies a particular profile from the XCCDF document.

    Profiles are determined by the ***Profile*** tag in the XCCDF XML file. Use the **oscap** command to see a list of profiles within a given XCCDF file, for example:

    ```
    $ oscap info /usr/share/openscap/scap-rhel6-xccdf.xml
    Document type: XCCDF Checklist
    Checklist version: 1.1
    Status: draft
    Generated: 2011-10-12
    Imported: 2012-11-15T22:10:41
    Resolved: false
    Profiles:
            RHEL6-Default
    ```

    If not specified, the default profile is used.

    NOTE

    Some early versions of OpenSCAP in Red Hat Enterprise Linux 5 require that you use the **--profile** option or the scan will fail.

- `--skip-valid`: Do not validate input and output files. You can use this option to bypass the file validation process if you do not have well-formed XCCDF content.

- **`Path to XCCDF Document:`** This is a required field. The ***`path`*** parameter points to the XCCDF content location on the client system. For example: `/usr/local/scap/dist_rhel6_scap-rhel6-oval.xml`

> **WARNING**
>
> The XCCDF content is validated before it is run on the remote system. Specifying invalid arguments can cause **`spacewalk-oscap`** to fail to validate or run. Due to security concerns, the **`oscap xccdf eval`** command only accepts a limited set of parameters.

For information about how to schedule scans using the Satellite web interface, see Procedure 6.1, “To Perform an Audit Scan Using the Web Interface:”.

CHAPTER 7. REPORTING CLIENT SOFTWARE FAILURES

You can take advantage of Red Hat Satellite's software failure reporting capabilities and the Automatic Bug Reporting Tool (ABRT) to extend the overall reporting functionality of your systems. This extended functionality allows your clients to automatically report software failures captured by ABRT to the Satellite server, and also to process the captured failures in a centralized fashion. You can use either the webUI or the API to process these failure reports.

For information about setting up Red Hat Satellite tools for ABRT on client systems, see the *Red Hat Satellite Client Configuration Guide*.

7.1. VIEWING SOFTWARE FAILURES FOR A SINGLE CLIENT

The following procedure shows how to view software reports for a single client system with Satellite's ABRT tools installed.

Procedure 7.1. To view software failures for a single client

1. Log into your Red Hat Satellite Web UI.
2. Click **Systems** → ***system_name*** → **Software** → **Software Crashes** to see the list of software failures that occurred on the registered system.
3. Click the required failure to see its details and the files captured for this software failure report.

The individual software failure report from the client system displays.

7.2. GROUPING SIMILAR SOFTWARE FAILURES

The Red Hat Satellite Web UI provides a page to group software failures across all systems by **`Crash UUID`**. This helps with identifying similar software crashes on your clients.

Procedure 7.2. To view similar software failures across clients

1. Log into your Red Hat Satellite Web UI.
2. Click **Systems** → **Software Crashes** to see a list of all software failures across all registered systems.
3. Click the on a **`Crash UUID`** to see the systems affected by the software failure.
4. Click on a specific system to see details and the files captured for the individual software failure report.

The software failure report from the client system displays.

7.3. CHANGING ORGANIZATION-WIDE SETTINGS FOR SOFTWARE FAILURE REPORTS

Red hat Satellite provides the ability to change the organization-wide settings for software failure reports. For example, with every software failure, clients upload the files captured by ABRT during the failure to your Satellite server. Because these files may be of arbitrary length, you can configure an organization-wide size limit for the upload of a single crash file.

The following procedure shows how to modify organization-wide settings for software failures.

Procedure 7.3. To change the organization-wide settings for software failures

1. In the Satellite Web UI, click **Admin** → ***<organization_name>*** → **Configuration**.
2. Modify the desired organization-wide and upload size settings, then click **`Update Organization`** to save.

The organization-wide settings for individual crash files is now changed to the chosen values.

7.4. LOG FILES OF SOFTWARE FAILURES

The log files captured by ABRT as a result of software failures are uploaded to your Satellite server for every failure report. You can download these files using either the Web UI or the API. On the Satellite server, these log files are physically stored in the **`/var/satellite/systems/$org_id/$system_id/crashes/$crash_name/`** directory.

CHAPTER 8. GENERATING RED HAT SATELLITE REPORTS

This chapter is designed to help you generate reports from Red Hat Satellite and Red Hat Satellite Proxy.

Red Hat Satellite and Red Hat Satellite Proxy contain a number of command-line reports:

- **`channel-packages`** - Packages in channels
- **`channels`** - Channel report
- **`custom-info`** - Display system custom info
- **`entitlements`** - Entitlement and channel list and usage
- **`errata-channels`** - List of errata in channels
- **`errata-list`** - Errata information based upon compliance checks against systems
- **`errata-list-all`** - List of all erratas
- **`errata-systems`** - Listing of each errata applicable to each affected system
- **`inactive-systems`** - Inactive systems in Satellite
- **`inventory`** - Inventory report
- **`kickstartable-trees`** - List of kickstartable trees
- **`packages-updates-all`** - List of packages that can be upgraded
- **`packages-updates-newest`** - List of packages that can be upgraded
- **`scap-scan`** - Results of OpenSCAP xccdf evaluation
- **`scap-scan-results`** - Results of OpenSCAP xccdf evaluation
- **`system-crash-count`** - Crash count for systems
- **`system-crash-details`** - Crash details for systems
- **`system-currency`** - System currency list
- **`system-groups`** - System groups in Satellite
- **`system-groups-keys`** - Activation keys for system groups
- **`system-groups-systems`** - Systems in system groups
- **`system-groups-users`** - System groups users report
- **`system-history`** - System event history
- **`system-history-channels`** - Channel event history
- **`system-history-configuration`** - Configuration event history

- **`system-history-entitlements`** - System entitlement event history
- **`system-history-errata`** - Errata event history
- **`system-history-kickstart`** - Kickstart event history
- **`system-history-packages`** - Package event history
- **`system-history-scap`** - OpenSCAP event history
- **`system-packages-installed`** - Packages installed on systems
- **`users`** - Users in the system
- **`users-systems`** - Systems administered by individual users

To generate a report, use the **`spacewalk-report`** command as follows:

```
# spacewalk-report report-name
```

This command generates the selected report as comma-separated value (CSV) output.

To more information, run the **`spacewalk-report`** command with the **`-h`** option.

CHAPTER 9. SCHEDULING RED HAT SATELLITE ADMINISTRATIVE TASKS

Red Hat Satellite allows organization administrators to regularly perform long-term operations using the `taskomatic` service. These operations are segregated into individual tasks and grouped logically into a *bunch* that is defined by schedules. You can modify these schedules to execute at specific time intervals. Satellite schedules are used to:

- Remove the administrative burden from the organizational administrator by automating tasks.
- Schedule operational tasks for time frames that will not tax the organization's daily network traffic.

Red Hat Satellite provides default schedules that trigger specific task bunches.

Table 9.1. Default Schedules in Red Hat Satellite 5.6

Schedule Name	Bunch Name	Bunch Function
channel-repodata-default	channel-repodata-bunch	Generates channel repository data.
cleanup-data-default	cleanup-data-bunch	Cleans up orphaned and outdated data.
clear-taskologs-default	clear-taskologs-bunch	Clears taskomatic run log history.
cobbler-sync-default	cobbler-sync-bunch	Applies any cobbler configuration changes.
compare-configs-default	compare-configs-bunch	Schedules a comparison of configuration files on all systems.
daily-status-queue	daily-status-bunch	Sends daily report.
errata-cache-default	errata-cache-bunch	Recalculates errata cache for a given server or channel.
errata-queue-default	errata-queue-bunch	Processes errata.
kickstart-cleanup-default	kickstart-cleanup-bunch	Cleans up stale kickstart files.
kickstartfile-sync-default	kickstartfile-sync-bunch	Synchronizes kickstart profiles that were generated using the wizard.
package-cleanup-default	package-cleanup-bunch	Cleans up orphaned packages.
sandbox-cleanup-default	sandbox-cleanup-bunch	Cleans up sandbox.

Schedule Name	Bunch Name	Bunch Function
satcert-check-default	satcert-check-bunch	Determines expiration status of Satellite certificate.
session-cleanup-default	session-cleanup-bunch	Deletes expired rows from the PXTSessions table to prevent it from growing too large.
sync-probe-default	sync-probe-bunch	Synchronizes probe state.

9.1. SCHEDULING A RUN

A *run* is a single execution of a bunch according to a configured schedule. You can schedule a run based on the default template provided by Red Hat Satellite, or you can create an entirely new schedule.

Procedure 9.1. Creating a Schedule Template

1. Log in to Satellite as the Organization Administrator.
2. Click **Admin** → **Task Schedules** → **Create Schedule**.
3. Complete the following fields:
   - Schedule Name: must begin with a letter and contain only lowercase characters, hyphens, periods, underscores, or numerals.
   - Bunch: the default bunch of administrative tasks the administrator can choose from.
   - Frequency

     The following frequency options are available:

     - Disable Schedule: only recommended for administrators who have advanced knowledge of the scheduled tasks and their consequences. Disabling schedules can change Satellite behavior.
     - Daily: creates a daily schedule for a specific time of day.
     - Weekly: creates a weekly schedule for a specific day and time of day.
     - Monthly: creates a monthly schedule for a specific day and time of day.
     - Custom Quartz Format: this format relies on cron expressions to define the schedule. For more information about this format, see the crontab man page (**`man 5 crontab`**.)
4. Click **`Create Schedule`**.

Procedure 9.2. Editing Schedule Templates

As an alternative to creating a new schedule, you can edit the default templates. To edit one of the existing templates:

1. Log in to Satellite as the Organization Administrator.

2. Click **Admin** → **Task Schedules**.

3. Click the schedule that you want to modify.

4. Change the Frequency type as required.

5. Click **`Edit Schedule`**.

APPENDIX A. REVISION HISTORY

Revision 3-20.402 **Thu Aug 20 2015** **Dan Macpherson**

Mass publication of all Satellite 5.6 books

Revision 3-20.401 **Tue Mar 17 2015** **Dan Macpherson**

-m

Revision 3-20.400 **2013-10-31** **Rüdiger Landmann**

Rebuild with publican 4.0.0

Revision 3-20 **Fri Sep 27 2013** **Dan Macpherson**

Final version of documentation suite

Revision 3-19 **Wed Sep 11 2013** **Dan Macpherson**

Minor change

Revision 3-18 **Wed Sep 11 2013** **Dan Macpherson**

Modified tablespaces section to align with Satellite 5.6 requirements

Revision 3-17 **Wed Sep 11 2013** **Dan Macpherson**

Removing old screenshots

Revision 3-16 **Tue Sep 10 2013** **Dan Macpherson**

Revised Subtitle, Abstract and Preface for all Guides

Revision 3-15 **Thu Aug 29 2013** **Dan Macpherson**

First implementation of QE Review feedback

Revision 3-14 **Tues Aug 20 2013** **Megan Lewis**

Corrections for BZ#990387, BZ#990393, BZ#990398, BZ#990400, and BZ#990383

Revision 3-13 **Mon Jul 29 2013** **Dan Macpherson**

Adding Software Failure chapter based upon tech review feedback

Revision 3-12 **Sun Jul 28 2013** **Dan Macpherson**

Second implementation of tech review feedback

Revision 3-11 **Wed Jul 24 2013** **Dan Macpherson**

Corrections for BZ#987245

Revision 3-10 **Tue Jul 23 2013** **Dan Macpherson**

First implementation of tech review feedback

Revision 3-9 **Fri Jul 12 2013** **Dan Macpherson**

Final beta updates

Revision 3-8 **Fri Jul 12 2013** **Dan Macpherson**

Updates to beta packages

Revision 3-6 **Thu Jul 11 2013** **David O'Brien**

Move chapter on creating RPM files to Reference Guide.
Update section on failover. Include information about online backup and restore.
Review section on OpenSCAP.
Add section on scheduling administrative tasks.
Add section on disaster recovery.
Update section on cloning software channels.

Revision 3-5	**Wed Sept 19 2012**	**Dan Macpherson**

Final packaging for 5.5

Revision 3-4	**Fri Aug 31 2012**	**Athene Chan**

BZ#839798 Minor edit

Revision 3-3	**Fri Aug 24 2012**	**Athene Chan**

BZ#839798 Changed 4.3 example to a standard format

Revision 3-3	**Fri Aug 24 2012**	**Athene Chan**

BZ#839798 Changed 4.3 example to a standard format

Revision 3-2	**Fri Aug 24 2012**	**Athene Chan**

BZ#826501 QA-reviewed changes applied.
BZ#884313 QA-reviewed changes applied.

Revision 3-1	**Fri Aug 17 2012**	**Athene Chan**

BZ#848313 OpenSCAP chapter "How to View SCAP Results" added

Revision 3-0	**Thu Aug 9 2012**	**Athene Chan**

Staging for Review

Revision 2-5	**Wed Aug 1 2012**	**Athene Chan**

BZ#839798 Added spacewalk-clone-by-date chapter
BZ#826501 New OpenSCAP information added

Revision 2-0	**Fri Jul 6 2012**	**Athene Chan**

Prepared for RHN Satellite 5.5 release
BZ#826501 Added OpenSCAP Chapter
OpenSCAP Screenshots added

Revision 1-5	**Mon Aug 15 2011**	**Lana Brindley**

Folded z-stream release into y-stream

Revision 1-4	**Mon Jun 20 2011**	**Lana Brindley**

BZ#701900 - PAM Authentication

Revision 1-3	**Mon Jun 20 2011**	**Lana Brindley**

BZ#714029 - Fixed color in image

Revision 1-2	**Wed Jun 15 2011**	**Lana Brindley**

Prepared for publication

Revision 1-1	**Fri May 27 2011**	**Lana Brindley**

Updates from translators

Revision 1-0	**Fri May 6, 2011**	**Lana Brindley**

Prepare for translation

Revision 0-15	**Thu May 5, 2011**	**Lana Brindley**

BZ#701818 - QE Review

Revision 0-14 **Mon May 2, 2011** **Lana Brindley**

BZ#248465 - QE Review

Revision 0-13 **Fri Apr 29, 2011** **Lana Brindley**

BZ#692295 - QE Review

Revision 0-12 **Mon Apr 18, 2011** **Lana Brindley**

BZ#691985 - Updating image

Revision 0-11 **Mon Apr 18, 2011** **Lana Brindley**

BZ#691990 - QE Review

Revision 0-10 **Mon Apr 18, 2011** **Lana Brindley**

BZ#691985 - QE Review

Revision 0-9 **Thu Apr 14, 2011** **Lana Brindley**

Technical review feedback

Revision 0-8 **Wed Apr 13, 2011** **Lana Brindley**

BZ#692314 - QE Review
BZ#692294 - QE Review
BZ#692291 - QE Review
BZ#692290 - QE Review
BZ#691988 - QE Review
BZ#691986 - QE Review
BZ#691981 - QE Review

Revision 0-7 **Wed Mar 23, 2011** **Lana Brindley**

Preparation for technical review

Revision 0-6 **Mon Feb 19, 2011** **Lana Brindley**

RPMs
Boot Devices
Organizations

Revision 0-5 **Fri Feb 18, 2011** **Lana Brindley**

Monitoring
PAM Authentication

Revision 0-4 **Mon Jan 10, 2011** **Lana Brindley**

Backup and Restore

Revision 0-3 **Fri Jan 7, 2011** **Lana Brindley**

User Administration
Preface
Automatic Synchronization

Revision 0-2 **Wed Jan 5, 2011** **Lana Brindley**

User Administration

Revision 0-1 **Tue Jan 4, 2011** **Lana Brindley**

Completed new chapter structure

Revision 0-0 **Tue Dec 21, 2010** **Lana Brindley**

New document creation from original RHN Satellite Deployment Guide