



HP-UX Secure Resource Partitions (SRP)

A.02.02 Administrator's Guide

HP-UX 11i v3

Table of contents

Preface.....	5
Intended Audience	5
Typographic Conventions.....	5
Related Information	6
Publishing History.....	6
HP Encourages Your Comments	6
1 Introduction.....	7
1.1 Product Overview	7
1.1.1 Securing SRP Compartments	8
1.1.2 Subsystems Configured by SRP	9
1.2 SRP Components.....	10
1.2.1 The SRP Manager.....	11
1.2.2 The <code>srp_sys</code> Utility	11
1.2.3 The <code>srp</code> Utility.....	11
1.2.4 The <code>srp_su</code> command	11
1.2.5 The <code>srp_ps</code> utility.....	11
1.2.6 SRP Templates and Services.....	11
1.2.7 Configuration Synchronization Manager (CMGR) Utility and Libraries	13
1.3 Planning Considerations and Best Practices.....	14
1.3.1 Compatibility with Other Partitioning Continuum Products	14
1.3.2 Coexistence with the <code>INIT</code> Compartment.....	14
1.3.3 Cross-Compartment Network Traffic.....	15
1.3.4 IP Routers and Strong End System (ES) Model	15
1.3.5 SRP Login Users.....	15
1.3.6 Compatibility with the Bastille Revert Feature	16
1.3.7 Compatibility with PRM SRP Commands.....	16
1.3.8 Serviceguard Support	16
1.4 Installing SRP	16
1.5 Migrating to A.02.02.....	16
2 Setting Up an SRP.....	18
2.1 The <code>srp_sys</code> Utility.....	18
2.2 Using <code>srp_sys -setup</code> to Set or modify system properties.....	18
2.3 Example: <code>srp_sys -setup</code>	19
2.4 Using <code>srp_sys -list</code> to Display System Properties	22
2.5 Example: <code>srp_sys -list</code>	22
3 Executing the <code>su</code> Command in the Target SRP	23
3.1 Using the <code>srp_su</code> Command.....	23
3.2 Allowing Additional Users to Use the <code>srp_su</code> Command	23
3.3 Example: Using the <code>srp_su</code> Command to Login to the Target SRP.....	23

4 Reporting Process Status for an SRP Compartment	25
5 Using SRP Manager	26
5.1 Configuring and Managing SRPs with SRP Manager	26
6 Getting Started with SRP	29
6.1 Sample SRP Lifecycle	29
6.1.2 Run Environment for the SRP Session	29
Step 1: Setting Up SRP	30
Step 2: Displaying Input Parameters for the base Template	30
Step 3: Creating an SRP Compartment	30
Step 4: Listing the Configuration Data	31
Step 5: Adding the sshd Template	32
Step 6: Listing the Configuration Data for the sshd Template	32
Step 7: Starting the SRP Compartment	33
Step 8: Listing SRP status information	34
Step 9: Replacing SRP Configuration Data	34
Step 10: Stopping the SRP Compartment	35
Step 11: Deleting the SRP Compartment	35
7 Using the SRP Environment	36
7.1 Establishing a User Session in the SRP	36
7.2 Managing SRP Startup and Shutdown Actions	36
7.3 Deploying Applications in an SRP Environment	37
7.3.1 Single Instance Applications	37
7.3.2 Multi-Instance Applications	37
7.3.3 Deploying Applications with the Application Templates	38
7.3.4 Ensuring access to application files located outside the SRP home directory	38
7.3.5 Best Practices for Application Deployment with SRP	38
8 Using the base Template	40
8.1 Creating a SRP Compartment	40
8.1.1 The <code>cmpt</code> Service	41
8.1.2 The <code>admin</code> Service	42
8.1.3 The <code>prm</code> Service	42
8.1.4 The <code>network</code> Service	44
8.1.5 The <code>init</code> Service	46
8.1.6 The <code>login</code> Service	46
8.1.7 The <code>ipfilter</code> Service	47
8.1.8 The <code>ipsec</code> Service	48
8.1.9 Completing the Configuration	50
8.2 Replacing or Deleting Base SRP Data	50
9 Using the apache Template	52
9.1 Adding the apache Template to an SRP Compartment	52
9.1.1 The <code>cmpt</code> Service	52
9.1.2 The <code>ipfilter</code> Service	53
9.1.3 The <code>provision</code> Service	54
9.2 Replacing or Deleting Apache SRP Data	56
10 Using the tomcat Template	57
10.1 Adding the tomcat Template to an SRP Compartment	57
10.1.1 The <code>cmpt</code> Service	57
10.1.2 The <code>ipfilter</code> Service	58
10.1.3 The <code>provision</code> Service	59
10.2 Replacing or Deleting Tomcat SRP Data	61
11 Using the custom Template	62
11.1 Adding the custom Template to an SRP Compartment	62
11.1.1 The <code>cmpt</code> Service	63
11.1.2 The <code>ipfilter</code> Service	63
11.1.3 The <code>provision</code> Service	64

11.2 Replacing or Deleting Custom SRP Data.....	65
12 Using the oracledb Template.....	66
12.1 Adding the oracledb Template to an SRP Compartment.....	66
12.1.1 The cmpt Service.....	66
12.1.2 The ipfilter Service.....	67
12.1.3 The provision Service.....	68
12.2 Replacing or Deleting Oracle SRP Data.....	68
13 Using the sshd Template.....	69
13.1 Adding the sshd Template to an SRP Compartment.....	69
13.1.1 The cmpt Service.....	69
13.1.2 The ipfilter Service.....	70
13.1.3 The provision Service.....	71
13.2 Replacing or Deleting SSHD SRP Data.....	72
14 Starting and Stopping SRP Compartments.....	73
14.1 SRP Startup and Shutdown Processing.....	73
14.2 Starting an SRP Compartment.....	74
14.3 Stopping an SRP Compartment.....	74
15 Managing SRP Data.....	76
15.1 Creating an SRP Compartment or Adding Data to an SRP.....	76
15.2 Deleting Configuration Data.....	77
15.3 Replacing Configuration Data.....	78
15.4 Displaying Help Text and Input Parameters.....	78
15.5 Listing Configuration Information About SRP Compartments.....	79
15.6 Displaying status of SRP Compartments.....	80
15.7 Using srp in Batch Mode.....	80
16 Customizing SRP Data.....	81
16.1 Modifying Provision Scripts.....	81
16.2 Modifying Compartment Rule Include Files.....	81
16.2.1 Securing SRP Compartments with Compartment Rule Include Files.....	81
16.3 Manually Editing SRP Configuration Data.....	82
16.3.1 Tag Formats.....	82
17 Exporting and Importing SRPs.....	85
17.1 Using the srp -export Command.....	85
17.2 Using the srp -import Command.....	86
17.3 Best practices for Exporting and Importing an SRP.....	87
18 Using Serviceguard with SRP.....	88
18.1 Choosing a Model.....	88
18.2 Creating an SRP to Use with Serviceguard.....	88
18.3 Adapting Serviceguard Scripts for the Classic Model.....	89
18.4 Creating Serviceguard Scripts for the SRP Package Model.....	90
19 Verifying and Troubleshooting SRP.....	91
19.1 Verification Procedures.....	91
19.1.1 Verifying SRP Subsystems.....	91
19.1.2 Verifying Security Containment Compartment Data.....	91
19.1.3 Verifying RBAC Data.....	92
19.1.4 Verifying PRM Data.....	92
19.1.5 Verifying Network Data.....	93
19.1.6 Verifying IPFilter Data.....	94
19.1.7 Verifying IPSec Data.....	94
19.2 Troubleshooting Procedures.....	95
19.2.1 Using the Security Containment Compartment Discover Feature.....	95
19.2.2 Removing or Disabling IPFilter.....	96
19.2.3 Removing or Disabling IPSec.....	96
19.3 Reporting Problems.....	97

Appendix A Configuration Example.....	98
A.1 Sample Base Configuration.....	98
The base.srp_incl File.....	99
Appendix B SRP Serviceguard Default Route Script.....	102

Preface

This document describes how to install, configure, and troubleshoot HP-UX Secure Resource Partitions (SRP).

Intended Audience

This document is intended for system and network administrators responsible for installing, configuring, and managing HP-UX SRP. Administrators are expected to have knowledge of operating system and networking concepts, commands, and configuration. Familiarity with the HP-UX Security Containment, HP Process Resource Manager (PRM), HP-UX IPFilter, and HP-UX IPSec products is useful. This document is not a tutorial.

Typographic Conventions

This document uses the following typographical conventions:

<code>%</code> , <code>\$</code> , or <code>#</code>	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells. A number sign represents the superuser prompt.
<code>audit(5)</code>	A manpage. The manpage name is <i>audit</i> , and it is located in Section 5.
Command	A command name or qualified command phrase.
Computer output	Text displayed by the computer.
Ctrl+x	A key sequence. A sequence such as Ctrl+x indicates that you must hold down the key labeled Ctrl while you press another key or mouse button.
ENVIRONMENT VARIABLE	The name of an environment variable; for example, <code>PATH</code> .
ERROR NAME	The name of an error, usually returned in the <code>errno</code> variable.
Key	The name of a keyboard key. Return and Enter both refer to the same key.
<i>Term</i>	The defined use of an important word or phrase.
User input	Commands and other text that you type.
<i>Variable</i>	The name of a placeholder in a command, function, or other syntax display that you replace with an actual value.
[]	The contents are optional in syntax. If the contents are a list separated by <code> </code> , you can choose one of the items.
{ }	The contents are required in syntax. If the contents are a list separated by <code> </code> , you must choose one of the items.
...	The preceding element can be repeated an arbitrary number of times.
:	Indicates the continuation of a code example.
	Separates items in a list of choices.
WARNING	A warning calls attention to important information that if not understood or followed results in personal injury or nonrecoverable system problems.
CAUTION	A caution calls attention to important information that if not understood or followed results in data loss, data corruption, or damage to hardware or software.
IMPORTANT	An important provides essential information to explain a concept or to complete a task.
NOTE	A note contains additional information to emphasize or supplement important

points of the main text.

Related Information

For more information about the products and subsystems used with SRP, see the following documentation:

- HP-UX Security Containment and Role-Based Access Control (RBAC), documented in the *HP-UX System Administrator's Guide: Security Management: HP-UX 11i Version 3*.
- HP-UX IPFilter
- HP-UX IPSec
- HP-UX Encrypted Volumes and File Systems (EVFS)

These documents are located at:

<http://www.hp.com/go/hpux-security-docs>

Select the **HP-UX Secure Resource Partitions (SRP) Software** product.

HP Process Resource Manager (PRM) is located at:

<http://www.hp.com/go/hpux-core-docs>

Select the **HP-UX 11i v3** product.

Publishing History

The document printing date and part number indicate the document's current edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The document part number will change when extensive changes are made. Document updates may be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new editions, you should subscribe to the appropriate product support service. See your HP sales representative for details.

This document is located at:

<http://www.hp.com/go/hpux-security-docs>

Select the **HP-UX Secure Resource Partitions (SRP) Software** product.

Manufacturing Part Number	Supported Operating Systems	Supported Versions	Publication Date
5900-0911	HP-UX 11i v3	Version 2.2	August 2010
5992-5172	HP-UX 11i v3	Version 2.01	December 2009
5992-4679	HP-UX 11i v3	Version 2.0	October 2008

HP Encourages Your Comments

HP encourages your comments concerning this document. We are committed to providing documentation that meets your needs. Send any errors found, suggestions for improvement, or compliments to:

<http://www.hp.com/bizsupport/feedback/ww/webfeedback.html>

Include the document title, manufacturing part number, and any comment, error found, or suggestion for improvement you have concerning this document.

1 Introduction

This chapter addresses the following topics:

- *1.1 Product Overview*
- *1.2 SRP Components*
- *1.3 Planning Considerations and Best Practices*
- *1.4 Installing SRP*
- *1.5 Migrating to A.02.02*

1.1 Product Overview

HP-UX Secure Resource Partitions (SRP) provides a lightweight workload consolidation environment that enables you to consolidate multiple workloads within a single instance of the HP-UX operating system. SRPs share a single Operating system kernel, system service daemons, administrative domain and file system namespace.. Each SRP compartment can have:

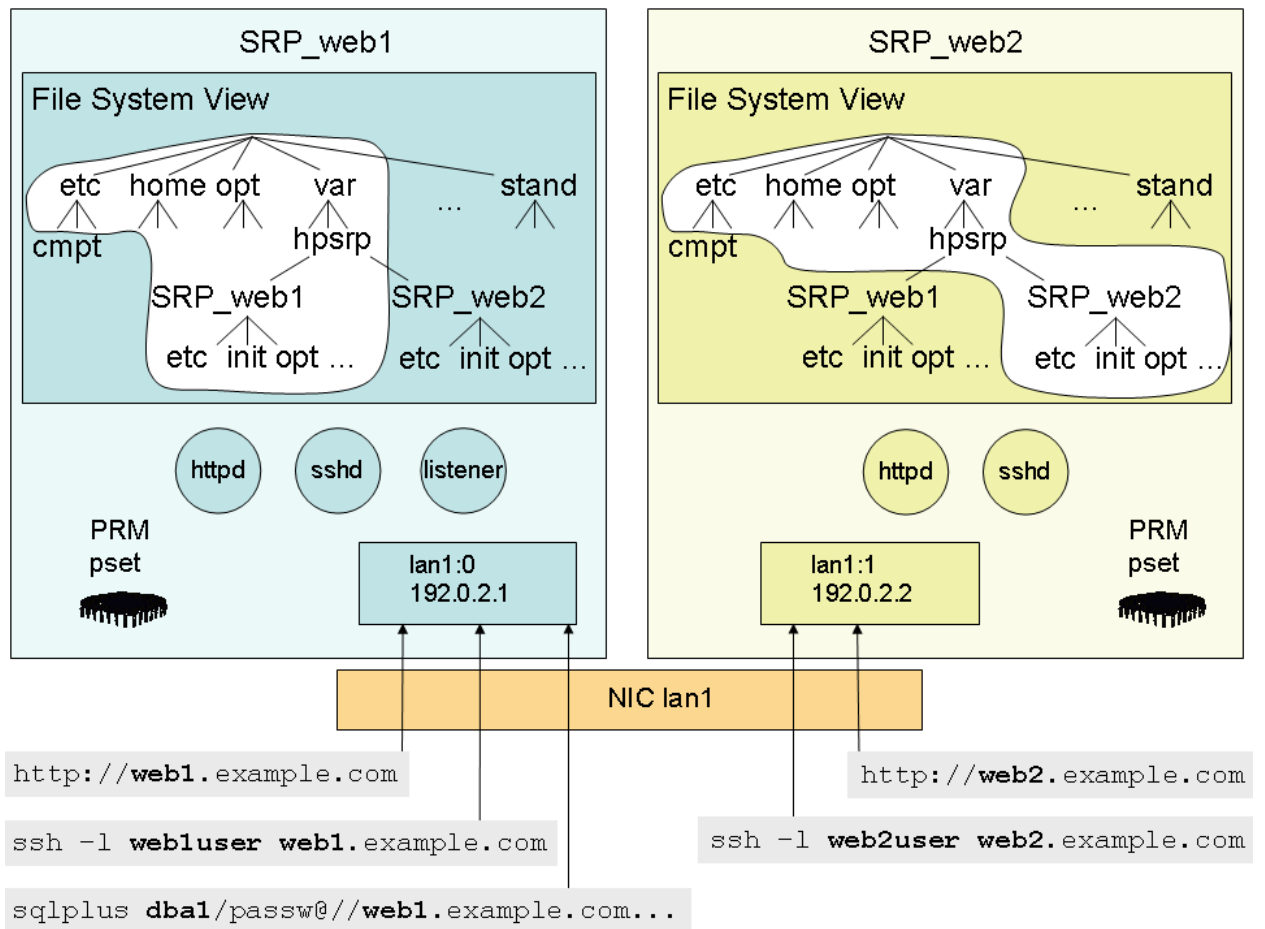
- A compartment home directory tree, which is isolated from other compartments.
- A dedicated IP interface.
- Isolated interprocess communication (IPC).
- A compartment-specific login environment.
- Dedicated CPU and memory resources.
- Per-compartment initialization and shutdown capabilities. You can start or stop an SRP compartment as you would start or stop a single system.
- Compartment-specific network security policies.

Because SRP enables you to configure and control these features on a per-compartment basis, each compartment forms an isolated execution environment. You can create multiple SRP compartments in a single image of an HP-UX operating system, which enables you to consolidate multiple applications on a single HP-UX OS image.

The configuration data for an SRP compartment encompasses data for multiple HP-UX subsystems and features, including HP-UX Security Containment and HP Process Resource Manager (PRM). SRP identifies this data using *tags*, or special text identifiers. This enables you to configure and manage the parameters for these subsystems as a single unit. Adding an SRP compartment creates configuration data for multiple HP-UX services, and deleting an SRP compartment removes all data configured for the compartment. For more information about SRP tags, see *16.3.1 Tag Formats*.

Figure 1.1 shows a system with two SRP compartments. Each compartment has a dedicated IP interface, isolated compartment home directory (`/var/hpsrp/srp_name`), compartment login group, dedicated processor set (`pset`), and separate instances of network daemons running.

Figure 1.1 SRP Compartments Example



1.1.1 Securing SRP Compartments

SRP provides a framework for managing compartment and networking security. This framework is primarily enforced with Security Containment compartment access rules. The default set of compartment access rules delivered with SRP has been developed to favor functional isolation, application compatibility and user session functionality over strong security containment. To meet the specific security requirements of your environment, you may need to replace these rules with security configuration to meet your application usage and local security policy, as described in 16.2.1 *Securing SRP Compartments with Compartment Rule Include Files*.

To secure the network packets for an SRP compartment, you can use the HP-UX IPFilter or HP-UX IPsec products. SRP can manage the configuration data for both these products, and you can use the SRP `srp_sys` utility to include these products in the default set of products configured by SRP.

You can also use HP-UX Encrypted Volume and File system (EVFS) to protect disk data at rest, or disk data that is not in use, such as when a disk device is physically transported. For more information on EVFS, see the *HP-UX Encrypted Volume and File system (EVFS) Administrator's Guide*.

1.1.2 Subsystems Configured by SRP

SRP can configure the following subsystems and HP-UX features:

- HP-UX Security Containment
- HP Process Resource Manager (PRM)
- IP interfaces
- Initialization and Shutdown Services
- HP-UX IPFilter
- HP-UX IPSec

1.1.2.1 HP-UX Security Containment

HP-UX Security Containment is a set of features that enhance HP-UX security. HP-UX Security Containment consists of the following components:

- Security Containment Compartments

A Security Containment compartment is an environment with a isolated file directory structure, isolated IPC, and isolated networking I/O for the processes and users in the compartment. If a process in a compartment is compromised, it cannot damage other parts of the system because it is isolated by the compartment configuration.

- HP-UX Role-Based Access Control (RBAC)

HP-UX Role-based Access Control (RBAC) is an alternative to the traditional "all-or-nothing" root user model, which grants permissions to the root user for all operations and denies permissions to non-root users for certain operations.

RBAC checks if an entity (such as a user or process) has the proper authorization value to perform an operation on a system resource. With RBAC, you can configure specific users to have access to specific resources such as files and executables. You can also configure the type of access allowed. For example, you can use RBAC so that only specific users can execute a given utility.

The RBAC configuration structure assigns authorization values to roles and assigns users (or subjects, which can also be executables) to roles. This structure enables you to assign a user to multiple roles, and therefore, have multiple authorization values. This also enables you to configure users that share some authorization values, but not necessarily share all of the same authorization values.

- Compartment Login

The compartment login feature enables you to control which compartment a user is allowed to log in to and which users are allowed to log in to a compartment. For example, you can configure the system so that only specific users can login to a given SRP compartment.

For more information, see *HP-UX System Administrator's Guide: Security Management*.

1.1.2.2 HP Process Resource Manager (PRM)

HP Process Resource Manager (PRM) manages CPU and memory allocation and enables you to configure dedicated resources for an SRP compartment. PRM can be used to set minimum and maximum allocations of system resources available to processes in an SRP compartment. When PRM is enabled for SRP, each SRP compartment is assigned a PRM group.

1.1.2.3 IP Interfaces

You can use SRP to create an IP interface for exclusive use by the compartment. You do not have to use a dedicated network interface card for this IP interface; you can create a logical IP interface on a network interface card.

An SRP compartment can also use an IP interface that is already in use by the system if it is not assigned to another compartment.

1.1.2.4 Initialization and Shutdown Services

You can use SRP to create an initialization and shutdown directory structure for the compartment with compartment control scripts that are automatically executed when the system starts up or shuts down. You can also execute a compartment control script to manually start or shut down an SRP compartment.

1.1.2.5 HP-UX IPFilter

HP-UX IPFilter is a host-based firewall software solution that enables you to restrict network traffic according to packet attributes, such as:

- Source IP address
- Destination IP address
- Protocol (such as TCP or UDP)
- TCP and UDP port numbers

1.1.2.6 HP-UX IPSec

HP-UX IPSec enables you to secure IP packets by encrypting and authenticating IP data. You configure IPSec to select packets for security according to packet attributes, such as:

- Source IP address
- Destination IP address
- Protocol (such as TCP or UDP)
- TCP and UDP port numbers

1.2 SRP Components

SRP includes the following components:

- The SRP Manager
- The `srp_sys` utility
- The `srp` utility
- The `srp_su` utility
- The `srp_ps` utility
- SRP templates, which manage configuration data for services
- The Configuration Synchronization Manager (CMGR) utility and libraries

1.2.1 The SRP Manager

The SRP Manager is integrated in the System Management Homepage (SMH) and provides a graphical user interface (GUI) to configure and manage HP-UX SRPs. See *5 Using SRP Manager* for more information.

1.2.2 The `srp_sys` Utility

The `/opt/hpsrp/bin/srp_sys` utility manages system-wide configuration properties for SRP. It is required to run `srp_sys -setup` to configure the system for SRP prior to configuring individual SRPs on the system. You can also use `srp_sys` to view and modify system-wide configuration settings for SRP. See *2 Setting Up an SRP* for more information.

1.2.3 The `srp` Utility

The `/opt/hpsrp/bin/srp` utility is an interactive program that prompts you for information and creates an SRP compartment by configuring the subsystems described in the previous section. The input parameters and configuration data created is determined by the templates and services you use with `srp`, as described in *1.2.4 SRP Templates and Services*. The `srp` utility supports options to perform the following tasks:

- Create an SRP compartment
- List the status of SRPs
- List the SRP configuration contents
- Replace configuration information for an existing SRP compartment
- Delete all or part of the configuration for an existing SRP compartment
- Start up or shut down an SRP compartment
- Export (copy) the SRP into an SRP exchange package
- Import (create) an SRP from an SRP exchange package
- Display help information, including information about input parameter

1.2.4 The `srp_su` command

The `/opt/hpsrp/bin/srp_su` command allows a user in the INIT compartment to execute a `su` command in the specified target SRP compartment. This can be used by system administrators for the purpose of login or command execution within a SRP. See *3 Executing the `su` Command in the Target SRP* for more information.

1.2.5 The `srp_ps` utility

The `/opt/hpsrp/bin/srp_ps` utility reports process status for a compartment in a Secure Resource Partition environment. See *4 Reporting Process Status for a Compartment* for more information.

1.2.6 SRP Templates and Services

The input parameters and data configured by `srp` are determined by the templates and services used. SRP *templates* are XML documents that define the configuration actions performed by SRP. Configuration actions are grouped into SRP *services*. You can choose which services to apply to an SRP, and apply services individually or collectively to an SRP compartment.

1.2.6.1 Templates

SRP includes the following templates:

- `base`
Configures a base SRP compartment without any application-specific parameters. A base compartment consists of a Security Containment compartment, a compartment home directory subtree, a compartment file system view, and other configuration data. After you create a base SRP compartment, you can apply one of the following application templates to extend the base with parameters suitable for applications hosted by a compartment.
- `apache`
Manages the configuration and provisioning of an HP-UX Apache-based Web Server in an SRP compartment.
- `tomcat`
Manages the configuration and provisioning of an HP-UX Tomcat-based servlet Engine in an SRP compartment.
- `custom`
Manages customized configuration of the SRP compartment. You can use this template to specify additional Security Containment file access rules, IPFilter rules and Provisioning for an SRP compartment.
- `oracledb`
Manages configuration to enable the SRP compartment to access an Oracle database installation intended to be shared by multiple SRP compartments. If you intend to install a separate instance of the Oracle database software inside the SRP compartment, you do not need to use this template.
- `sshd`
Manages the configuration and provisioning of an HP-UX Secure Shell server daemon in an SRP compartment.

1.2.6.2 Services

SRP supports the following services:

- `cmpt`
Manages configuration data for an HP-UX Security Containment compartment, which forms the core of the SRP compartment. You must use the `cmpt` service when you create an SRP compartment.
- `admin`
Uses the HP-UX Security Containment RBAC feature to associate an HP-UX user with an RBAC role that has authorization to administer the compartment. By default, this authorization enables the administrator to execute the `srp start` and `srp stop` commands for the SRP from the `INIT` compartment.

- `login`
Defines the users and groups allowed to login to the SRP compartment. Uses the HP-UX Security Containment RBAC and compartment login features to configure the compartment login access for a set of HP-UX users and groups. If compartment login is enabled for the system with the default RBAC configuration and you do not configure the SRP `login` service, only the `root` (UID 0) user is allowed to log in to the compartment.
- `prm`
Configures a PRM group for the SRP compartment. You can specify the PRM group type and the CPU and memory allocations for the group.
- `provision`
Executes a script to deploy an application in an SRP compartment. HP provides provision scripts for Apache Web Server, Tomcat Servlet Engine, and Secure Shell daemon (`sshd`) templates.
- `network`
Configures an IP interface for use by a compartment. By default, SRP IP interfaces will not be shared between SRPs, however these interfaces are accessible by default from the INIT compartment.
- `init`
Creates compartment startup and shutdown scripts and a compartment-specific `init` directory structure that replicates the `/sbin/init.d` directory structure. By default, the scripts are automatically executed by the system startup and shutdown scripts.
- `ipfilter`
Configures IPFilter rules for the compartment. For the base template, SRP configures rules that restrict inbound IP packets to the compartment's IP interface. When used with application templates, SRP prompts you for local port numbers and configures rules that allow packets that match the specified ports.
- `ipsec`
Configures HP-UX IPSec policies for the compartment. SRP prompts you for the local and remote IP addresses and configures IPSec policies to encrypt and authenticate packets that match the address specifications. The `ipsec` service also configures an Internet Key Exchange (IKE) policy and an IKE preshared key.

1.2.7 Configuration Synchronization Manager (CMGR) Utility and Libraries

The Configuration Synchronization Manager (CMGR) product is included in the SRP bundle. The CMGR product includes the `cmgr` utility and libraries, which enables SRP to coordinate the configuration of multiple subsystems. The `srp` utility invokes the `cmgr` utility.

For more information about CMGR, refer to the *HP-UX CMGR Administrator's and Developer's Guide*.

1.3 Planning Considerations and Best Practices

This section contains information to consider when planning an SRP deployment and best practices to follow when managing a system with SRP compartments.

1.3.1 Compatibility with Other Partitioning Continuum Products

HP-UX SRP is a component of the Partitioning Continuum for HP-UX and is compatible with HP-UX nPartitions, HP-UX vPar, and Integrity Virtual Machine (VM) solutions. You can create an SRP in any HP-UX OS image; the OS image can exist in an nPartition, vPar, Integrity VM, or directly on non-partitioned server hardware.

1.3.2 Coexistence with the `INIT` Compartment

The `INIT` compartment is a permanent, default compartment defined by the Security Containment product. The `INIT` compartment provides a comprehensive host-based view of the system. By default, all system processes and services (all processes started by the `init` process) run in the `INIT` compartment, and the `INIT` compartment has access to all files and processes. The `INIT` compartment also has access to all interfaces configured in other compartments, including the `ifaces` compartment and all SRPs. (When you run `srp_sys -setup`, the Security Containment product is initially enabled, it creates the `ifaces` compartment and assigns all network interfaces currently installed on the system to `ifaces`.)

1.3.2.1 Using the `INIT` Compartment

You must perform system administration activities in the `INIT` compartment. By default, a login to the system console or a network based session (`ssh`, `telnet`) to the `iface` compartment IP addresses will result in a session in the `init` compartment. To verify that your session is in the `init` compartment, you can use the following command to return the name of the compartment you are running in:

```
getprocxsec -c
```

1.3.2.2 Address Collisions with `INADDR_ANY` and `IN6ADDR_ANY` Sockets in the `INIT` Compartment

Because the `INIT` compartment has access to all network interfaces configured in other compartments, it is possible for a socket owned by a process running in the `INIT` compartment that binds to the wildcard IP address `INADDR_ANY` or `IN6ADDR_ANY` to bind to the specified port number on all IP or IPv6 interfaces on the system. This means that socket owned by a process in the `INIT` compartment can bind to an IP address that is configured for another compartment. (Note that compartments other than `INIT` can bind only to IP addresses for which they been explicitly configured access.)

An address collision can occur if a process in the `INIT` compartment and a process in an SRP compartment attempt to use the same port number and either process attempts to bind the socket to the `INADDR_ANY` or `IN6ADDR_ANY` address. If both sockets have the `SO_REUSEADDR` option set, both bind calls will succeed, but either socket may receive a given connection request. If both sockets do not have the `SO_REUSEADDR` option set, the second bind call will fail.

1.3.2.2.1 Address Collisions with `sshd` Daemons

One example of a network daemon that might have problems with address collisions is the `sshd` daemon. By default, the `sshd` daemon binds its socket to TCP port 22 on the wildcard IP address `INADDR_ANY` (or `IN6ADDR_ANY`, if the IPv6 address family is specified). If an `sshd` daemon is runs in the `INIT` compartment with the default configuration and a second `sshd` daemon starts in an SRP compartment and attempts to bind its socket to TCP port 22 on the compartment IP address, the bind

will succeed. However, the `sshd` daemon running in the SRP compartment might not receive SSH connection requests on its socket.

To prevent `sshd` address collisions, the `srp_sys` utility prompts for the system `sshd` configuration file name (the configuration file that the `sshd` daemon running in the `INIT` compartment would use) and checks if this file configures the daemon to listen on a wildcard IP address. If so, `srp_setup` asks if you want to set the `ListenAddress` variable to specific addresses instead of a wildcard IP address.

1.3.2.3 Recommendations

Because of the `INIT` compartment properties, HP recommends that you:

- Do not use the `INIT` compartment to run non-system management applications or non-essential services. Any application or service that is not intended to be shared by SRPs should be run in an SRP and not in `INIT`.
- Manage system resources when logged in to the `INIT` compartment. If a utility manages system-wide resources or configuration files, such as `SMH`, run the utility from the `INIT` compartment. The SRP utilities manage system resources and should be executed from the `INIT` compartment.
- Run `swinstall` and `swremove` from the `INIT` compartment. Do not install system software or utilities from within an SRP compartment. By default, an SRP compartment will have file access rules that prevent you from successfully installing system software.
- Execute associated applications from within the same SRP compartment. This enables the processes to share common file system directories, IPC facilities, and network security rules.

1.3.3 Cross-Compartment Network Traffic

SRP compartments provide isolated networking environments. By default, an SRP compartment is configured so that the only networking traffic allowed is through the compartment-specific IP interface, through the physical network layer, or through the network loopback layer to other SRPs on the same server. You can manually configure compartment access rules (network block rules) to prevent loopback networking to a second SRP compartment.

NOTE: Configuring cross-compartment rules can interfere with the ability to import compartments to another system. See *17 Exporting and Importing SRPs* for more details.

1.3.4 IP Routers and Strong End System (ES) Model

To ensure proper routing, SRP configures the system to use the strong end system (ES) model, as described in RFC 1122 to provide symmetric routing of connection based network traffic. When the strong ES model is used, a system cannot act as an IP router. A system with the strong ES model silently drops incoming IP packets with destination IP addresses that do not match the interface address. Outbound IP packets must use the interface address as the source IP address.

1.3.4.1 Application Gateway Servers

Although SRP systems cannot be used as IP routers, they can be used as application gateway servers. Application gateway servers receive IP packets sent to a local IP address, process the packets at an upper layer, and retransmit the packets using the local IP address as the source address.

1.3.5 SRP Login Users

The SRP `login` service assigns a set of HP-UX users and groups the RBAC authority to log in to the compartment. Only users in this set will be allowed to login to the SRP. HP recommends that you create a group for each SRP and apply this group to the SRP login service.

NOTE: By default, RBAC configuration also authorizes the `root` user to log in to all compartments.

1.3.6 Compatibility with the Bastille Revert Feature

If you use the `bastille -r` command to revert to the Bastille baseline configuration, you may lose any IPFilter rules configured using SRP that are not in the baseline. HP recommends that you do not configure the IPFilter service with SRP if you are using Bastille to manage IPFilter rules. If Bastille is managing IPFilter rules, the `/etc/opt/ipf/ipf.conf` or `/etc/opt/ipf/ipf.conf` file contains a statement similar to the following:

```
# WARNING: This file was generated automatically and will be replaced
# the next time you run Bastille. DO NOT EDIT IT DIRECTLY!!!
```

1.3.7 Compatibility with PRM SRP Commands

The HP PRM product includes the following commands to associate a Security Containment compartment with a PRM group:

- `prm2scomp`
- `scomp2prm`
- `srpgen`

HP recommends that you use the `srp` utility instead of the PRM SRP commands. You cannot use the `srp` utility to manage with Security Containment compartments and PRM groups created with the above commands, but SRP compartments can coexist with these compartments and PRM groups.

1.3.8 Serviceguard Support

All Serviceguard daemons must run in the `INIT` compartment. See *18 Using Serviceguard with SRP*, for more information on using Serviceguard with SRP.

1.4 Installing SRP

The HP-UX-SRP bundle consists of two products: CMGR and SRP. To use SRP, you must install both products in the bundle.

For system and environment requirements, see the *HP-UX SRP A.02.02 Release Notes* located at: www.hp.com/go/hpux-security-docs

Select the **HP-UX Secure Resource Partitions (SRP) Software** product.

You can acquire and install HP-UX Secure Resource Partitions free of charge from Software Depot: <http://www.software.hp.com>

1.5 Migrating to A.02.02

No manual migration steps are required to migrate from a previous version of HP-UX SRP. If you are upgrading from a previous SRP version, ensure that all the SRPs on the system are stopped before running the `swinstall` command to install the new SRP package. The following command displays the status of all the SRPs on the system:

```
# srp -status
```

HP-UX SRP version A.02.02 delivers new default values in the following configuration files:

```
/etc/rc.config.d/srpconf
/etc/opt/hpsrp/cmpt/apache.srp_incl
```



```
/etc/opt/hpsrp/cmpt/base.srp_incl  
/etc/opt/hpsrp/templates/srpdefaults.cst  
/etc/opt/hpsrp/cmpt/oracledb.srp_incl  
/etc/opt/hpsrp/cmpt/sshd.srp_incl  
/opt/hpsrp/bin/util/secsh_setup  
/opt/hpsrp/bin/util/srp_backup  
/opt/hpsrp/bin/util/srp_restore  
/opt/hpsrp/bin/util/apache_setup
```

If you are upgrading from a previous version of SRP and have already modified one of these files, the modified version will be used. For any of these files not modified from the original default values, the installation process will replace the configuration file with the new default version.

2 Setting Up an SRP

This chapter describes how to use `srp_setup` to set up the SRP environment. This chapter addresses the following topics:

- 2.1 The `srp_sys` Utility
- 2.2 Using `srp_sys -setup` to Set or modify system properties
- 2.3 Example: `srp_sys -setup`
- 2.4 Using `srp_sys -list` to Display System Properties
- 2.5 Example: `srp_sys -list`

2.1 The `srp_sys` Utility

The `/opt/hpsrp/bin/srp_sys` utility is used to set and view system-wide configuration properties that affect SRP. It is required to run `srp_sys -setup` before using the SRP Manager GUI or `srp` utility.

The `srp_sys -setup` utility has the following syntax:

```
srp_sys -setup
srp_sys -list [-v[erbose]]
srp_sys -help
```

where:

- `setup` configures and enables system wide configuration properties used by SRP
- `list` lists the configuration status of system wide configuration properties used by SRP
- `help` displays usage information for `srp list`.
- `verbose` displays detailed information for the list operation

2.2 Using `srp_sys -setup` to Set or modify system properties

The `srp_sys -setup` command ensures that the system is in an appropriate state for successful configuration of SRP compartments. The `srp_sys` utility checks the status of the subsystems that can be configured by SRP. If a subsystem is not enabled, `srp_sys` prompts if you want to enable the service. It also prompts for subsystem startup data, such as configuration directories and `autostart` parameters. Once executed `srp_sys -setup` modifies SRP default template with these subsystem startup data. `srp_sys -setup` also prompts you for the SRP services you want to enable. The services you enable also become the default services for the templates (SRP will not apply a service if the service is not valid for a given template).

HP requires that you run `srp_sys -setup` after you install SRP, but you can run it anytime that you want to change the default parameters for SRP.

You can use `srp_sys -setup` to enable the following features:

- **Security Containment compartments** (required for the SRP product). When the Security Containment compartments feature is initially enabled, it creates the `INIT` and `ifaces` compartments. For more information about the `INIT` and `ifaces` compartments, see [1.3.2 Coexistence with the INIT Compartment](#).
- **Compartment Login**. Enabling this feature configures the system to control user based authentication (including login) on a per SRP basis by enabling the `CMPT_LOGIN` flag in `/etc/cmpt/cmpt.conf` and verifying that `/etc/pam.conf` includes the required `pam_hpsec` module.

IMPORTANT: By default, once compartment login is enabled, only the root user (user name of `root`) is allowed to login to the INIT compartment. To allow additional users to login to the INIT compartment, you will need to assign any additional users to the RBAC role of `SRPlogin-init`.

To enable additional users for INIT compartment login:
`>roleadm assign <user_name> SRPlogin-init`

To enable additional groups for INIT compartment login:
`>roleadm assign "<group_name>" SRPlogin-init`

- **Strong ES Model** (required for the SRP product when using networking). Enables symmetric routing on the system which causes connection based protocols such as TCP to use the same interface for both inbound and outbound. Note that enabling the strong ES model makes the system unable to function as an IP router. For more information about the strong ES model, see *1.3.4 IP Routers and Strong End System (ES) Model*.
- **ip_ire_cmpt_route_lookup_policy/ ip6_ire_cmpt_route_lookup_policy** (required for the SRP product when using networking). Controls the route lookup logic in the compartment enabled environment. Set this feature to 0 to enable the strong security model which requires a strict route lookup logic; set this feature to 1 to disable the strong security model.
- **cmpt_allow_local** Allows SRPs on the same server to communicate via the network without requiring additional security configuration. Sets the default rule for inter-compartment loopback communications that are addressed to local network interfaces or IP addresses. The default rule only applies if there is no explicit compartment network rule matching the communication attempt.
- **Limited Scope Secure Shell Daemon.** Used to prevent the secure shell daemon in the INIT compartment from listening on SRP specific IP addresses. You can specify the IP addresses to be used, with the default being the system default IP address. (For more information about address collisions, see *1.3.2.2 Address Collisions with INADDR_ANY and IN6ADDR_ANY Sockets in the INIT Compartment*.)

2.3 Example: `srp_sys -setup`

In this example, the user presses **RETURN** and accepts the default values for each prompt.

```
# /opt/hpsrp/bin/srp_sys -setup
#####
#
# Setup SRP default template
#
#####

Loading SRP default template ... [ OK ]
```

The default services do not include IPFilter or IPSec. You can add them to the set of default services in the following dialog.

Enable SRP configuration for the following services:

```
admin (compartment administrator) [y] RETURN
```

```

init (compartment startup and shutdown scripts) [y] RETURN
login (compartment login via pam_security) [y] RETURN
network (IP address and network interface management) [y] RETURN
prm (Process Resource Management) [y] RETURN
ipfilter (ipfilter host firewall rules) [n] RETURN
ipsec (ipsec secure transport rules) [n] RETURN
provision (run customizable provision script) [y] RETURN

```

Selected SRP service(s) are: cmpt,admin,init,login,network,prm,provision
Would you like to save the changes? [y] RETURN

Saving SRP default template ... [OK]

```

#####
#
# Compartment Setup
#
#####

```

Checking the Compartment module ... [Enabled]

```

#####
#
# cmpt Login configuration
#
#####

```

Checking Compartment Login Configuration File... [OK]

Checking cmpt login feature ... [Enabled]

Note: By default, once compartment login is enabled, only the root user (user name of "root") will be allowed to login to the INIT compartment. To enable additional users, execute the following command:

```
roleadm assign <user_name> SRPlogin-init
```

Adding RBAC role (SRPlogin-init) for INIT cmpt login ... [OK]

Any service monitored by pam_hpsec account management module is enabled with compartment login enabled.

The current PAM configuration file (/etc/pam.conf) is the same as the system default PAM configuration file (/usr/newconfig/etc/pam.conf). You can keep it for compartment login purpose.

```

#####
#
# PRM Setup
#
#####

```

Checking PRM installation ... [OK]

Enter PRM configuration file [/etc/prmconf] RETURN

Saving SRP default template ... [OK]

Checking PRM Memory record for OTHERS group ... [OK]

Checking PRM starts at boot-up ... [Enabled]

```

#####
#
# network configuration
#
#####

Checking network strong ES model ... [ Enabled ]
Checking compartment IPv4 routing policy ... [ Enabled ]

Checking compartment IPv6 routing policy ... [ Enabled ]

Checking kernel tunable cmpt_allow_local ... [ Enabled ]

#####
#
# sshd configuration
#
#####

Checking sshd configuration ... [ OK ]
Enter sshd configuration file: [/opt/ssh/etc/sshd_config]

Saving SRP default template ... [ OK ]

Detected Init Compartment Secure Shell daemon listening on all IP
addresses.
Will conflict with any SRP Secure Shell daemons.
Would you like to restrict the Init compartment's sshd IP addresses? [y]
RETURN

Enter IP addresses, separated by comma ',': [15.146.224.214] RETURN
sshd will then listen on these interfaces:
ListenAddress 15.146.224.214

Would you like to save the changes to /opt/ssh/etc/sshd_config? [y]
RETURN
Changes saved to /opt/ssh/etc/sshd_config

Would you like to restart Init Compartment Secure Shell daemon? [y]
RETURN

Restarting sshd ... [ OK ]

#####
#
# IPFilter Setup
#
#####

Checking the IPFilter module ... [ Enabled ]

#####
#
# IPsec configuration
#
#####

Checking IPsec installation... [ OK ]

Checking IPsec status... [ Enabled ]

```

Would you like to set/change IPsec password? [n] **RETURN**

Checking IPsec starts at boot-up... [Enabled]

```
#####  
#  
# SRP setup completed.  
#  
#####
```

2.4 Using `srp_sys -list` to Display System Properties

You can use the `srp_sys -list` command to review the current settings for system-wide configuration options affecting SRP. For each of the subsystems configured with `srp-sys - setup`, the `srp_sys -list` command displays the current configuration status:

- **OK** The subsystem is installed, enabled and configuration has been validated.
- **Unverified** The subsystem is installed, enabled, but the configuration has been customized in a way that prevents validation
- **Invalid Config** The subsystem is installed, enabled but has failed the configuration validation check, or is configured with an invalid option for the SRP environment
- **Not Enabled** The subsystem is installed, but has not been enabled
- **Not Installed** The subsystem software has not been installed

You can use the `-verbose` option to obtain more detailed information

2.5 Example: `srp_sys -list`

```
% bin/srp_sys -l
```

```
Default Service List ... [ OK ]  
Security Containment Compartments... [ OK ]  
Compartment Login Configuration File... [Unverified]  
PRM Configuration... [ OK ]  
PRM Memory record status: Enabled [ OK ]  
Network strong ES model ... [ OK ]  
IPFilter module ... [ OK ]  
IPsec installation... [ OK ]  
IPsec autostart status: Enabled [ OK ]  
SSHD listener in INIT... [ OK ]
```

3 Executing the su Command in the Target SRP

The `srp_su` command executes the `su(1)` command in the specified SRP. You must execute the `srp_su` command from within the `INIT` compartment. System administrators can use this command to login or execute a command within an SRP.

This chapter addresses the following topics:

- 3.1 Using the `srp_su` Command
- 3.2 Allowing Additional Users to Use the `srp_su` Command
- 3.3 Example: Using the `srp_su` Command to Login to the Target SRP

3.1 Using the `srp_su` Command

The `srp_su` command has the following syntax:

```
srp_su srp_name [su_arguments]
```

Where:

`srp_name`: Name of the target SRP compartment.
`su_arguments`: Arguments to be passed to the `su(1)` command in the target SRP. Any `su` arguments may be used.

Only users with the `hpux.security.srp_su` authorization are allowed to use the `srp_su` command. By default, only the `root` user has this authorization for all SRPs on the system.

3.2 Allowing Additional Users to Use the `srp_su` Command

To allow additional users to use the `srp_su` command, you must create new RBAC roles, and assign the additional users to the role, as follows:

1. Create one new `hpux.security.srp_su` authorization per system:
authadm add hpux.security.srp_su
2. Create a new role per SRP:
roleadm add newRole
3. Assign the `hpux.security.srp_su` authorization to one role per SRP:
authadm assign newRole hpux.security.srp_su "srp_name"
4. Assign a role to each user:
roleadm assign user_name newRole

NOTE: Repeat step 4 for each additional user.

3.3 Example: Using the `srp_su` Command to Login to the Target SRP

In this example, the `root` user establishes a session as `root` in the target SRP. The `root` user logs in `mySRP` SRP from the `INIT` compartment:

```
# /opt/hpsrp/bin/srp_su mySRP
```

The `root` user logs in from the `INIT` compartment to `mySRP` SRP as user `admin1` with a new login session in the `mySRP`:

```
# /opt/hpsrp/bin/srp_su mySRP - admin1
```

User `admin1` logs in from the `INIT` compartment to `mySRP` SRP as `admin2` with a new login session, where `admin2` is configured for compartment login.

Create a new RBAC rule to allow user `admin1` to use the `srp_su` command as follows:

1. Create a new `hpux.security.srp_su` authorization.
authadm add hpux.security.srp_su
2. Create a new `SRPsu-mySRP` role:
roleadm add SRPsu-mySRP
3. Assign the `hpux.security.srp_su` authorization to the `SRPsu-mySRP` role for `mySRP`:
authadm assign SRPsu-mySRP hpux.security.srp_su "mySRP"
4. Assign user `admin1` to the `SRPsu-mySRP` role:
roleadm assign admin1 SRPsu-mySRP

Verify that the role was assigned properly by using the `srp_su` command to create a session in `mySRP`, as follows:

```
# /opt/hpsrp/bin/srp_su mySRP - admin2
```

The correct `admin2` user password will allow `admin2` to login to the `mySRP` SRP.

4 Reporting Process Status for an SRP Compartment

To report process status for an SRP compartment, use the `srp_ps` utility, located in the `/opt/hpsrp/bin` directory. The `srp_ps` utility invokes the `ps` command with the provided `ps_arguments` and filters the `ps` output for the desired SRP compartment:

```
srp_ps [srp_name] [ps_arguments]
```

When you run the `srp_ps` utility from the `INIT` compartment and supply an `srp_name`, `srp_ps` prints process status for the specified compartment. If you do not supply a `srp_name`, the `srp_ps` utility prints process status for the system compartments (`INIT` and `KERNEL`). When you run `srp_ps` within an SRP compartment, only the process status for the current SRP compartment is reported.

Example: To report the process status for an SRP compartment, login to the `INIT` compartment and execute the following command:

```
# srp_ps mySRP -eaf
```

Example: To report process status for the system compartments, login to the `INIT` compartment and execute the following command:

```
# srp_ps -ef
```

5 Using SRP Manager

The SRP Manager is integrated in the System Management Homepage (SMH) and provides a graphical user interface (GUI) to configure and manage HP-UX SRPs. With SRP Manager, you can perform the following tasks:

- Monitor SRP status and activity on your system
- Create a new SRP
- Start or stop an SRP
- Export and import an SRP
- Modify an SRP
- Delete an SRP

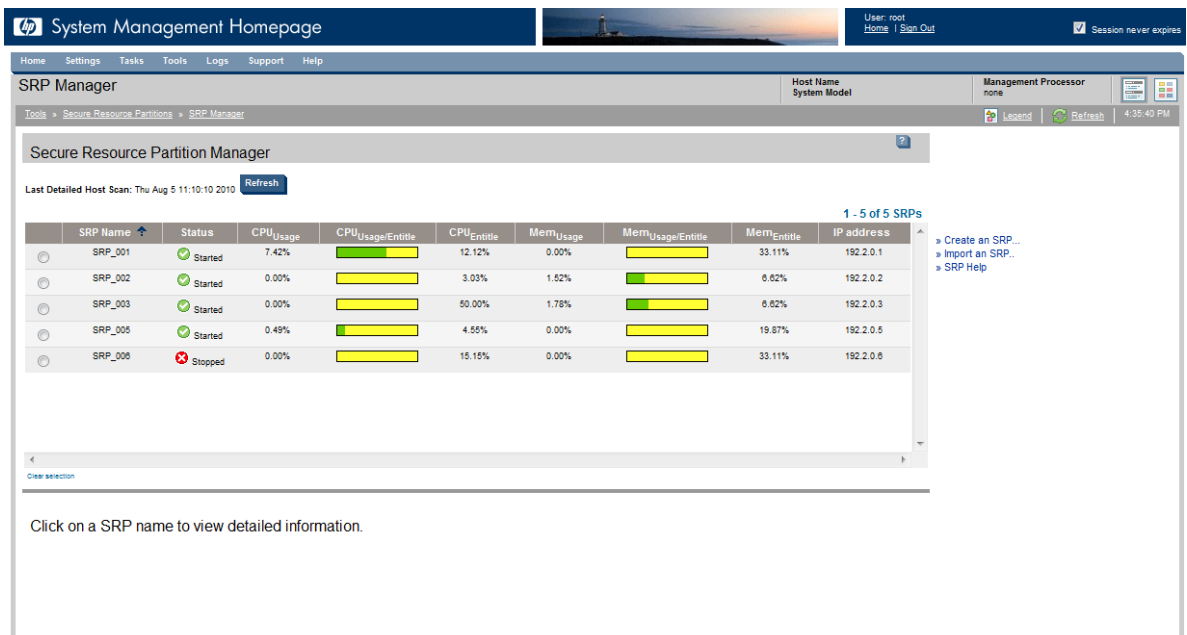
NOTE: HP requires that you run `srp_sys -setup` before using SRP Manager. For more information about `srp_sys -setup`, see [2 Setting Up an SRP](#).

5.1 Configuring and Managing SRPs with SRP Manager

This section provides an overview to introduce the basic operation of the SRP Manager. More information is included with the Help subsystem included with the SRP Manager.

The SRP Manager home page provides a view of all SRPs on the system including current state and resource utilization for each SRP. You can select an individual SRP for more detailed status or management activities. The following figure shows the SRP Manager home page.

Figure 5.1 SRP Manager



The screenshot displays the SRP Manager interface within the System Management Homepage. The main content area shows a table titled "Secure Resource Partition Manager" with the following data:

SRP Name	Status	CPUUsage	CPUUsage/Entitle	CPUEntitle	MemUsage	MemUsage/Entitle	MemEntitle	IP address
SRP_001	Started	7.42%	<div style="width: 7.42%;"></div>	12.12%	0.00%	<div style="width: 0.00%;"></div>	33.11%	192.2.0.1
SRP_002	Started	0.00%	<div style="width: 0.00%;"></div>	3.03%	1.52%	<div style="width: 1.52%;"></div>	6.62%	192.2.0.2
SRP_003	Started	0.00%	<div style="width: 0.00%;"></div>	50.00%	1.78%	<div style="width: 1.78%;"></div>	6.62%	192.2.0.3
SRP_005	Started	0.49%	<div style="width: 0.49%;"></div>	4.55%	0.00%	<div style="width: 0.00%;"></div>	19.87%	192.2.0.5
SRP_008	Stopped	0.00%	<div style="width: 0.00%;"></div>	15.15%	0.00%	<div style="width: 0.00%;"></div>	33.11%	192.2.0.6

From the SRP Manager help files, select the **SRP Listing and Status** help menu item for more information on the SRP home page.

When you create an SRP using SRP Manager, you can define a new SRP for your system and provision the SRP specific directory tree. You can also specify optional services (Network, PRM, IPFilter, IPSec, and templates (SSHD, Apache, Tomcat, Custom, and Oracle) to apply to the SRP.

To create an SRP, from the SRP Manager home page, click **Create an SRP** and the following screen appears:

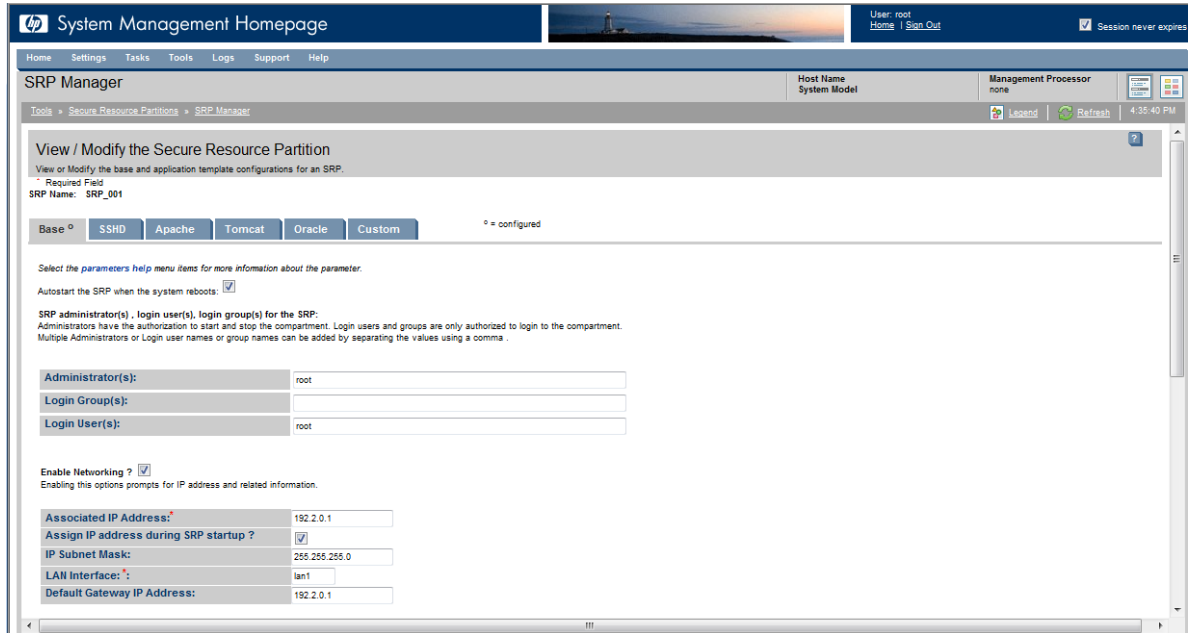
Figure 5.2 SRP Manager – Creating an SRP

The screenshot shows the SRP Manager web interface. The main heading is "Create a Secure Resource Partition". Below this, there is a "Required Field" section with an "SRP Name:" input field. A note indicates to select the parameters help menu items for more information. There is a checkbox for "Autostart the SRP when the system reboots:" which is checked. The "SRP administrator(s), login user(s), login group(s) for the SRP:" section includes fields for "Administrator(s):" (root), "Login Group(s):" (empty), and "Login User(s):" (root). The "Enable Networking?" checkbox is checked, with a note that enabling this prompts for IP address and related information. A section titled "Fill the form to add network service:" includes fields for "Associated IP Address:" (empty), "Assign IP address during SRP startup?" (checked), "IP Subnet Mask:" (empty), "LAN Interface:" (lan1), and "Default Gateway IP Address:" (empty). At the bottom, there is a checkbox for "Enable PRM Service Information?" which is unchecked. The interface includes a navigation menu at the top with "Home", "Settings", "Tasks", "Tools", "Logs", "Support", and "Help". The user is logged in as "root" and the session never expires.

From the SRP Manager help files, select the **Creating an SRP** help menu item for more information on creating an SRP.

Once an SRP is created, you can add or modify its configurations and service templates. To view or modify an SRP, from the SRP Manager home page, click **View/Modify** and the following screen appears:

Figure 5.3 SRP Manager – Viewing or Modifying an SRP



From the SRP Manager help files, select the **Viewing or Modifying an SRP** help menu item for more information on viewing or modifying an SRP.

6 Getting Started with SRP

This chapter shows the commands used to manage the lifecycle of a sample SRP compartment. This chapter addresses the following topics:

- *6.1 Sample SRP Lifecycle*
- *Step 1: Setting Up SRP*
- *Step 2: Displaying Input Parameters for the base Template*
- *Step 3: Creating an SRP Compartment*
- *Step 4: Listing the Configuration Data*
- *Step 5: Adding the sshd Template*
- *Step 6: Listing the Configuration Data for the sshd Template*
- *Step 7: Starting the SRP Compartment*
- *Step 8: Listing SRP status information*
- *Step 9: Replacing SRP Configuration Data*
- *Step 10: Stopping the SRP Compartment*
- *Step 11: Deleting the SRP Compartment*

6.1 Sample SRP Lifecycle

The following user session shows the SRP commands used to set up the SRP environment and then create, administer, and delete an example SRP compartment. Each command is numbered and described in the sections that follow.

```
# srp_setup #1 Set up SRP
# srp -help -template base #2 Show input parameters for the
base template
# srp -add mySRP #3 Create a base SRP compartment
# srp -list mySRP -v #4 List the configuration data
# srp -add mySRP -t sshd #5 Add the sshd template
# srp -list mySRP -v -t sshd #6 List the configuration data for
sshd
# srp -start mySRP #7 Start the SRP compartment
# srp -status mySRP #8 Get status of the SRP
# srp -replace mySRP -s prm #9 Replace the PRM configuration
values
# srp -stop mySRP #10 Stop the SRP compartment
# srp -delete mySRP -batch #11 Delete the SRP compartment
```

6.1.2 Run Environment for the SRP Session

By default, you must have superuser capability to run the `srp` utility. In addition, you must have the authorization to modify the system and subsystem configuration files managed by `srp`. You must run the `srp` utility from the `INIT` compartment. The `INIT` compartment is a permanent, default compartment defined by the Security Containment product. (If the Security Containment product is not already enabled, the `srp_setup` script enables it, which creates the `INIT` compartment.) By default, processes running in the `INIT` compartment have no compartment based restrictions on accessing system files.

For more information about using the `INIT` compartment, see *1.3.2.1 Using the INIT Compartment*.

All SRP utilities are located in the directory `/opt/hpsrp/bin`.

Step 1: Setting Up SRP

In this example, the product has just been installed. The `root` user runs `srp_sys -setup` to enable the subsystems managed by SRP.

HP requires that you run `srp_sys -setup` before using SRP Manager or the `srp` utility, but you can run it anytime that you want to change the default parameters for SRP or verify the status of the subsystems configured by SRP.

Now that you have setup your SRP, you can configure and manage your SRP using SRP Manager or the `srp` utility.

NOTE: HP recommends that you use the SRP Manager. See *5 Using SRP Manager* for more information on SRP Manager. If you want to use the `srp` utility, continue to Step 2.

Step 2: Displaying Input Parameters for the `base` Template

Before creating a base SRP compartment, the user enters the `srp -help -template base` command to view the input parameters for the `base` template. The `srp` utility displays usage and general syntax information, then displays the input parameters for the default services used with the `base` template.

```
# /opt/hpsrp/bin/srp -help -template base
```

```
srp -a[dd]|-d[elete]|-r[eplace] <srp_name>
    [-t[emplate] <template>] [-s[ervice] <service>,...] [-i[d]
<instance>]
    [-b[atch]] [-v[erbose]][variable=<value>,...]
    :
    :
```

Variables: (used with `-add`, `-replace`, will vary by `-template`. Variables without defaults must be supplied when executing with `-b`)

```
Template: base      Service: default
```

Name	Description
-----	-----
admin_user	Comma separated list of existing Unix user names to be granted the RBAC administrator role for the SRP. Default: root
login_group	Comma separated list of existing Unix group names allowed for login within the SRP. Default: adm
login_user	Comma separated list of existing Unix user names allowed for login within the SRP. Default: none
	:
	:

Step 3: Creating an SRP Compartment

To create a SRP compartment, enter the following command:

```
srp -add srp_name
```

Where `srp_name` is the name of the SRP compartment you want to create. In this example, the user specifies `mySRP` for the compartment name. In the `srp` dialog, the user specifies the IP address

192.0.2.1 for the compartment address and lan1 for the network interface. The user accepts the default values for all other variables.

The command output and user input for this example are as follows:

```
# /opt/hpsrp/bin/srp -a mySRP
```

Enter the requested values when prompted, then press return.
Enter "?" for help at prompt. Press control-c to exit.

```
Services to add: [cmpt,admin,init,prm,network,login] RETURN
List of Unix user names for compartment administrator: [root] RETURN
List of Unix group names for compartment login: [adm] RETURN
List of Unix user names for compartment login: [] RETURN
PRM group name to associate with this SRP compartment: [mySRP] RETURN
PRM group type (FSS, PSET): [FSS] RETURN
PRM FSS group CPU shares: [10] RETURN
PRM FSS group CPU cap (press return for no cap): [] RETURN
PRM group memory shares: [10] RETURN
PRM group memory cap (press return for no cap): [] RETURN
PRM group physical memory (press return for no dedicated memory): []
RETURN
IP address: [] 192.0.2.1
Assign IP address at SRP start time: [yes] RETURN
IP subnet mask (press return to accept default): [] RETURN
Network interface name: lan1
Gateway server IP address:[ 192.0.2.1] RETURN
Autostart SRP at system boot? [yes] RETURN
The following template variables have been set to the values shown:
```

```
iface          = lan1
ip_address     = 192.0.2.1
```

```
Press return or enter "yes" to make the selected modifications with these
values. Do you wish to continue? [yes] RETURN
add compartment rules succeeded
creating directory /var/hpsrp/mySRP ...
add compartment directory succeeded
add RBAC admin role for compartment succeeded
add RBAC compartment login role succeeded
add startup directories succeeded
add prm rules succeeded
add compartment network service rules succeeded
add ipaddress 192.0.2.1 succeeded
add compartment service succeeded
```

Step 4: Listing the Configuration Data

To list the data configured for the SRP compartment, enter the following command:

```
srp -list srp_name -v
```

The following is the abbreviated output for this example. For the complete output, see *A.1 Sample Base Configuration*.

```
# /opt/hpsrp/bin/srp -list mySRP -v
```

```
Compartment: mySRP  Template: base Service: cmpt
```

```

Compartment Configuration (/etc/cmpt/mySRP.rules):
@tag-start compartment="mySRP" template="base" service="cmpt" id="1" ;
#include "/opt/hpsrp/etc/cmpt/base.srp_incl"

// lock out access to the other compartment's root directory
perm nsearch          /var/hpsrp

// open access to compartment root
perm all              /var/hpsrp/mySRP

// to DNS
grant bidir          udp peer port 53 init
                    :
                    :

```

Step 5: Adding the sshd Template

After you have created a base SRP compartment, you can configure the compartment to host specific services using the `-t template_name` option. For example, to configure the compartment to host an HP-UX `sshd` daemon, enter the following command:

```
srp -add srp_name -t sshd
```

The `srp` utility prompts the user with a list of services valid for the template. In this example, the user specifies the `cmpt` and `provision` services and accepts the default values for all variables. The command output and user input for this example are as follows:

```
# /opt/hpsrp/bin/srp -a mySRP -t sshd
```

Enter the requested values when prompted, then press return.
Enter "?" for help at prompt. Press control-c to exit.

```

Services to add: [cmpt,provision] RETURN
sshd data path: [/var/hpsrp/mySRP/opt/ssh] RETURN
sshd executable path: [/opt/ssh] RETURN
Copy SSH config data from path: [/opt/ssh/newconfig] RETURN
sshd port number: [22] RETURN

```

```

Press return or enter "yes" to make the selected modifications with these
values. Do you wish to continue? [yes]
add compartment rules succeeded
add provision service succeeded

```

Step 6: Listing the Configuration Data for the sshd Template

To list the data configured for the `sshd` template, enter the following command:

```
srp -list srp_name -v -t sshd
```

The `srp` utility lists the compartment rules and added for the `sshd` template. To view all the configuration data for the compartment, omit the `-t sshd` argument.

The output for this example is as follows:

```
# /opt/hpsrp/bin/srp -l mySRP -v -t sshd
```

```
Compartment: mySRP  Template: sshd Service: cmpt
```



```

-----
Compartment Configuration (/etc/cmpt/mySRP.rules):
@tag-start compartment="mySRP" template="sshd" service="cmpt" id="1" ;
//
// allow access to the shared sshd files
//
perm nsearch      /opt
perm nsearch      /opt/ssh

perm nsearch,read /opt/ssh

perm nsearch      /var
perm nsearch      /var/hpsrp
perm nsearch      /var/hpsrp/mySRP
perm nsearch      /var/hpsrp/mySRP/opt
perm nsearch      /var/hpsrp/mySRP/opt/ssh

perm all /var/hpsrp/mySRP/opt/ssh

//
// add shared rules from the include file at
"/opt/hpsrp/etc/cmpt/sshd.srp_incl"

//
#include "/opt/hpsrp/etc/cmpt/sshd.srp_incl"

```

```

Compartment: mySRP  Template: sshd  Service: provision
-----

```

```

SSHD Configuration File:
    /var/hpsrp/mySRP/opt/ssh/sshd_config
SSHD Port:
    22
SSHD Key Files:
    /var/hpsrp/mySRP/opt/ssh/ssh_host_rsa_key
    /var/hpsrp/mySRP/opt/ssh/ssh_host_rsa_key.pub
    /var/hpsrp/mySRP/opt/ssh/ssh_host_dsa_key
    /var/hpsrp/mySRP/opt/ssh/ssh_host_dsa_key.pub
SSHD Pid File:
    /var/hpsrp/mySRP/opt/ssh/sshd.pid
SSHD Startup/Shutdown Script:
    /var/hpsrp/mySRP/sbin/init.d/secsh
SSHD Provision Script:
    /opt/hpsrp/bin/util/sec_sh

```

Step 7: Starting the SRP Compartment

To start an SRP compartment, enter the following command:

```
srp -start srp_name
```

The `srp` utility starts the SRP compartment by setting the SRP state to Started and executing the initialization scripts in the `/var/hpsrp/srp_name/sbin/init.d` subdirectories.

```
# /opt/hpsrp/bin/srp -start mySRP
```

```

*****
Compartment mySRP startup in progress
Mon Dec  7 13:58:18 IST 2009
*****
Configure LAN interfaces..... [ OK ]
Mounting file systems in /var/hpsrp/mySRP/etc/fstab..... [ OK ]
Starting HP-UX Secure Shell..... [ OK ]

```

Step 8: Listing SRP status information

To display the status of an SRP, the `srp` command can be called with the `status` option. The following command displays the status of the SRP `mySRP`:

```
srp -status mySRP
```

If the SRP name is not specified, the status is displayed for all SRPs configured on the system.

Example: Status of the SRP `srptest` using the `srp -status srptest` command:

```

% srp -status srptest

SRP Status:

----- Status for SRP:mySRP -----
      Status:STARTED

      IP: 192.0.2.x   Interface:lan0:1 (UP)

      MEM Entitle:5.38%   MEM Max:(none)   Usage:0.14%
      CPU Entitle:2.78%   CPU Max:(none)   Usage:0.00%

```

Step 9: Replacing SRP Configuration Data

To replace configuration data, use the `srp -replace` command with the template name and service name as follows:

```
srp -replace srp_name [-t template_name] [-s service_name]
```

If you do not specify the `-t` option, SRP uses the base template. If you do not specify the `-s` option, SRP prompts you for information to replace data for all services valid for the template.

In this example, the user wants to replace the PRM data and increase the number of CPU shares to 20. The PRM data is configured with the base template. The base template is the default template for the replace operation, so the user does not have to specify the template name.

The command output and user input for this example are as follows:

```
# /opt/hpsrp/bin/srp -r mySRP -s prm
```

```
Enter the requested values when prompted, then press return.
Enter "?" for help at prompt. Press control-c to exit.
```

```

PRM group name to associate with this SRP compartment: [mySRP] RETURN
PRM group type (FSS, PSET): [FSS] RETURN
PRM FSS group CPU shares: [10] 20
PRM FSS group CPU cap (press return for no cap): [] RETURN
PRM group memory shares: [10] RETURN
PRM group physical memory (press return for no dedicated memory): []
RETURN

```

The following template variables have been set to the values shown:

```
prn_cpu_shares = 20
```

Press return or enter "yes" to make the selected modifications with these values. Do you wish to continue? [yes]
replace prn rules succeeded

Step 10: Stopping the SRP Compartment

To stop an SRP compartment, enter the following command:

```
srp -stop srp_name
```

```
stops the SRP compartment by executing the shutdown scripts in the
/var/hpsrp/srp_name/sbin/init.d subdirectories and setting the SRP state
to "stopped".# /opt/hpsrp/bin/srp -stop mySRP
*****
Compartment mySRP shutdown in progress
Mon Dec 7 13:58:40 IST 2009
*****
Stopping HP-UX Secure Shell..... [ OK ]
Killing user processes..... [ OK ]
Unmounting file systems in /var/hpsrp/mySRP/etc/fstab..... [ OK ]
Unconfigure LAN interfaces..... [ OK ]
```

Step 11: Deleting the SRP Compartment

Because this session was used to show examples of `srp` operations instead of an actual deployment, the user deletes the SRP compartment. The `srp -delete` command deletes all configuration data for the SRP compartment.

```
# /opt/hpsrp/bin/srp -d mySRP
```

Do you wish to delete the compartment "mySRP"? (yes/no) : [no] yes

```
Processing SRP template sshd ...
delete compartment rules succeeded
delete provision service succeeded

Processing SRP template base ...
delete compartment rules succeeded
delete compartment directory succeeded
delete RBAC admin role for compartment succeeded
delete RBAC compartment login role succeeded
delete startup directories succeeded
delete prn rules succeeded
delete compartment network service rules succeeded
delete ipaddress succeeded
deleting directory /var/hpsrp/mySRP ...
delete compartment service succeeded
#
```

7 Using the SRP Environment

Once you have created an SRP, and started it with the `srp -start` command, the SRP is now available for user sessions and execution of programs. This chapter discusses the following topics:

- 7.1 Establishing a User Session in the SRP
- 7.2 Managing SRP Startup and Shutdown Actions
- 7.3 Deploying Applications in an SRP Environment

7.1 Establishing a User Session in the SRP

HP recommends the following two methods to establish a user session in an SRP:

- **srp_su:** Use this command from the `INIT` compartment to establish a user session within the specified SRP compartment. Note that by default, this command is restricted to the `root` user. See 3 *Executing the su Command in the Target SRP* for instructions on how to use the `srp_su` command.
- **Secure Shell (SSH):** If you have applied the SSH template to the SRP, you can now connect across the network to the SRP via Secure Shell. Note that since all SRP's share a common host name, you should specify the SRP's specific IP address to ensure that you connect to the desired SRP. See 13 *Using the sshd Template* for instructions on how to apply the Secure Shell template.

NOTE: Login will be restricted to a set of Unix users and groups specified with the login service in the base template. If you did not apply the login service to this SRP, then login is not restricted.

7.2 Managing SRP Startup and Shutdown Actions

You can automate the activities that you want to be performed when the SRP is started or stopped as follows:

- **Init service:** Each SRP has a startup and shutdown directory structure that replicates the system startup and shutdown structure (`rc1.d` through `rc4.d` in `/var/hpsrp/srp_name/sbin/`). These scripts will be executed within the SRP when the SRP is started or stopped. Note that run levels 1 and 2 are used to manage the SRP startup and shutdown. HP recommends that you use only run levels 3 and 4 (`rc3.d`, and `rc4.d`) for deploying your start scripts. HP recommends that you follow the practices recommended on the `init(1M)` man page for creating and deploying your start scripts.
- **Hard kill:** By default, SRP provisions a shutdown script to kill all active processes in the SRP as one of the final shutdown activities within the SRP when the SRP is stopped (`/var/hpsrp/srp_name/sbin/init.d/srp_killall`). To disable this feature, remove the soft link file for the script:

```
# rm /var/hpsrp/srp_name/sbin/rc.1d/k640srp_killall
```

NOTE: If you disable the hard kill feature, the SRP can enter the `Stopped` state with active processes still executing in the SRP environment.

- **SRP setup:** The SRP setup and shutdown script (`/var/hpsrp/srp_name/.setup/setup`) is executed in the `INIT` compartment before and after startup and shutdown of the SRP (before the `init` service on `start` and after on `stop`). You can modify the script to perform system management activities you want to be

performed when the SRP is started or stopped, such as notifying management or auditing systems, or mounting the SRP home directory (`/var/hpsrp/srp_name`).

NOTE: If you are using shared storage to mount the SRP home directory to facilitate cloning of an SRP, consider using the SRP setup script to automatically mount and unmount the SRP home directory. In addition to ensuring that the home directory is available when the SRP is started, it will also help prevent the accidental deletion of the home directory when deleting one of the SRP clones.

- **fstab:** Each SRP has a file system mount table configuration file that replicates the system file system mount table described in `fstab(4M)`. This file (`/var/hp/hpsrp/srp_name/etc/fstab`) may be accessed and edited from within the SRP. The start script (`/var/hpsrp/srp_name/sbin/init.d/srp_mount`) performs the mounting and unmounting of file systems when the SRP is started and stopped.

7.3 Deploying Applications in an SRP Environment

HP recommends that you execute most applications from within an SRP environment. The `INIT` compartment should be reserved for the execution of system management activities. Note that third party system management utilities should also be executed in the `INIT` compartment. If you chose to deploy a system management utility within an SRP, you may find it necessary to customize the SRP compartment access rules to provide any increased capabilities necessary for the utility to perform its services.

This section describes the following topics:

- *7.3.1 Single Instance Applications*
- *7.3.2 Multi-Instance Applications*
- *7.3.3 Deploying Applications with the Application Templates*
- *7.3.4 Ensuring access to application files located outside the SRP home directory*
- *7.3.5 Best Practices for Application Deployment with SRP*

7.3.1 Single Instance Applications

While nearly all applications can be executed within an SRP environment, some applications do not support multiple instances of the same application executing on the same system concurrently. For these applications, HP recommends that you do not run multiple instances of these applications, even when the instances are located in separate SRPs, HP recommends that you install the application under the `/var/hpsrp/srp_name/` directory if user-specified installation location is supported by the vendor.

7.3.2 Multi-Instance Applications

There are two models to deploy multi-instance applications within an SRP environment:

- **Shared executable:** In this model, the application is installed once per system in a shared location. The application may require a set of files to be replicated and configured per instance. HP recommends that you locate per instance files under the `/var/hpsrp/<srp_name/` directory when supported by the vendor.
- **Per-SRP installation:** For applications that support multiple installations per system, you can install the application for each SRP in which it will be executed. HP recommends that you install the application under the `/var/hpsrp/srp_name/` directory.

7.3.3 Deploying Applications with the Application Templates

SRP includes special templates for deploying key applications that use shared executables. The `ssh`, `apache`, and `tomcat` templates, fully deploy these applications within the SRP using the shared executable model. The `oracledb` template configures the SRP for Oracle usage; however you must first install the Oracle database product on the system in the desired location. Optionally, you may also use the custom template to deploy an Oracle database for your SRP. If you are installing an Oracle database under the `/var/opt/hpsrp/srp_name` directory, the `oracledb` template is not required.

7.3.4 Ensuring access to application files located outside the SRP home directory

If the application files are not all located under `/var/hpsrp/srp_name/`, you must ensure that the compartment rules definition for the SRP includes sufficient capability to allow execution. For executable files, `READ` capability is generally sufficient, while configuration and data files will typically require `READ` and `WRITE` capability. See *11 Using the custom Template* for information on using the custom template to define application specific compartment access rules for your SRP. Note that in addition to any installed files, the application may also create files and directories during execution time. See *19 Verifying and Troubleshooting SRP* for instructions on using `Discover Mode` if you are unable to determine the access rules required by the application.

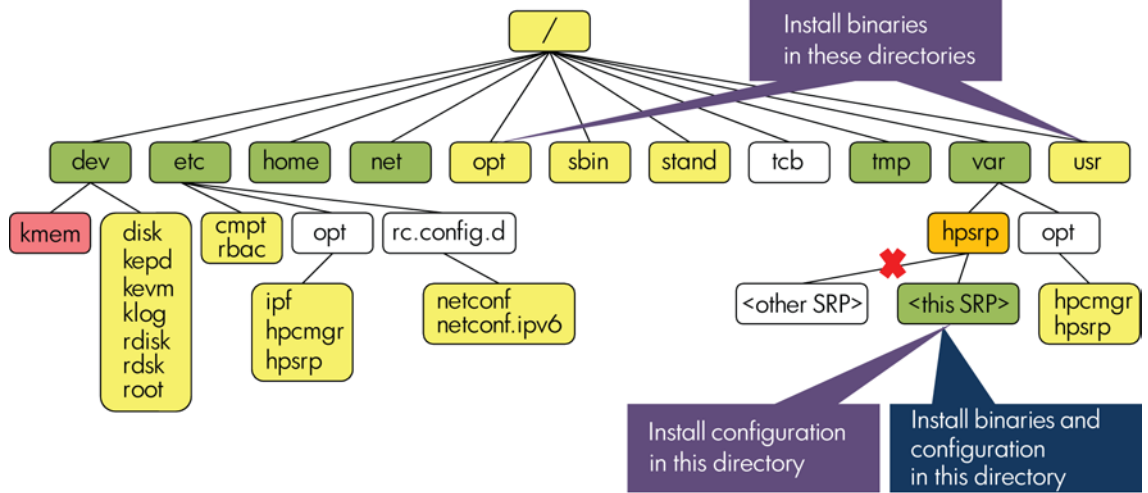
7.3.5 Best Practices for Application Deployment with SRP

Follow these best practices when deploying application with SRP:

- **Deploy as much of the application as possible under the SRP home directory.**
This minimizes the need to customize compartment access rules. When the application is installed entirely under the SRP home directory, customization of the SRPs compartment rules is usually not necessary. Life cycle management, including cloning and migration of the SRP will also be simplified as the application files will be managed as part of the SRP.
- **Deploy files shared by multiple SRPs under the standard Unix directories for hosting shared application files (for example, `/opt/`, `/usr/`).**
By default, SRPs are configured for the `READ` capability for these directories, and will not need additional compartment rules configuration.
- **If you have applied IPFilter for the SRP, ensure that any additional ports used by the application are allowed.**
When the `ipfilter` service is enabled for the SRP, by default the inbound network connections to the SRP are blocked. You must configure the `ipfilter` service to allow inbound connections to any network ports that the application will listen on.
- **Use the custom template to apply additional capabilities to the SRP for the application.**
This will allow you to manage system configuration changes for the SRP on a per SRP basis. Use a recognizable identifier, such as the application name for the `instance_id` parameter when deploying the custom template. When deploying multiple applications within an SRP, consider applying the custom template (if needed) once per application.

Figure 7.1 illustrates the installation rules and file locations.

Figure 7.1 Application Installation Map



Model	
Multi -install model	
Shared binary model	
Key	
all (no restrictions)	
read (read only)	
nread (read only - not inherited)	
none (access blocked)	
Inherited (rules from parent directory)	

8 Using the `base` Template

The `base` template manages SRP compartment data that is not application-specific. This chapter describes how to use the `base` template to create an SRP compartment. You can also use the `base` template to add additional base services to a compartment or to delete or modify the base services for a compartment.

This chapter addresses the following topics:

- [8.1 Creating a SRP Compartment](#)
- [8.2 Replacing or Deleting Base SRP Data](#)

8.1 Creating a SRP Compartment

You can use the `base` template to create an SRP compartment. After you create an SRP compartment, you can use an application template to add application-specific configuration data to the SRP compartment, such as compartment file access rules for application-specific directories and IPFilter rules for application-specific port numbers.

To create an SRP compartment, enter the following `srp -add` command. Specifying the base template (`-t base`) is optional; the base template is the default template for the add operation. The `srp -add` command has the following syntax:

```
srp -a[dd] srp_name [-t base] [-s service[,service]....]
```

Where:

srp_name

Specifies the name of the SRP compartment to create.

service

Specifies the name of the service to configure. If you do not specify the `-s` option, `srp` prompts you for a list of services to configure with a list of default services. The factory-configured default services are as follows (listed in the order that `srp` prompts for input):

- `cmpt` - see *The cmpt Service*
- `admin` - see *The admin Service*
- `prm` - see *The prm Service*
- `network` - see *The network Service*
- `init` - *The init Service*

You can modify the set of default services using the `srp_setup` utility as described in [2 Setting Up an SRP](#).

The following services are also valid with the `base` template:

- `login` - see *The login Service*
- `ipfilter` - see *The ipfilter Service*
- `ipsec` - see *The ipsec Service*

The input data for these services and the data configured are described in the sections that follow. If SRP uses input data for multiple services, the `srp` utility prompts you for the data once and reuses the value.

8.1.1 The `cmpt` Service

The `cmpt` Service configures an HP-UX Security Containment compartment, which forms the core of the SRP compartment. You must use the `cmpt` service when you create an SRP compartment; you cannot create an SRP compartment without the `cmpt` service.

8.1.1.1 Input Data

The `cmpt` service uses the compartment name specified in the `srp` command for the Security Containment compartment name.

8.1.1.2 Configuration Data

The `cmpt` service creates a home directory for the compartment using the following format:
`/var/hpsrp/srp_name`

The `cmpt` service creates a Security Containment compartment if one does not already exist with the same name. The rules for this compartment are stored in the file `/etc/cmpt/srp_name.rules`. This file, like all rule files created using the SRP `base` template, includes a reference to the `/opt/hpsrp/etc/cmpt/base.srp_incl` file.

When combined with the contents of the `base.srp_incl` file, the rule set properties includes the following:

- Access to the home directory for the compartment.
- Read-only access to system binary files, including kernel files (`/usr`, `/opt`, `/sbin`, and `/stand`).
- Full access to other commonly used system directories and files. This enables you to access the directories and files needed for most OS and networking operations. You might want to modify the file access rules to remove or limit access according to your environment.
- IPC access to the Security Containment `INIT` compartment. The `INIT` compartment is a special compartment defined by the Security Containment product. By default, most operating system processes (processes started by the `init` process) run in the `INIT` compartment. Allowing IPC access to the `INIT` compartment enables the SRP compartment to communicate with most local OS processes, including client network processes that communicate with remote systems.
- Network access for DNS request and reply packets through the network interfaces in the Security Containment `INIT` compartment. This enables DNS client routines running in the SRP compartment to send and receive packets to and from a DNS server on the local system.

A.1 Sample Base Configuration shows an example compartment rules file created by `srp` for an SRP compartment.

Compartment Home Directory

The `cmpt` service creates a home directory for the compartment (`/var/hpsrp/srp_name`) with the following subdirectories that are intended to be compartment-specific versions of the system subdirectories below the root directory:

- `etc`
- `home`
- `net`
- `opt`
- `sbin`
- `tmp`
- `usr`

- `var`

For example, SRP creates a `/var/hpsrp/srp_name/sbin` directory with `init.d`, `rc0.d`, `rc1.d`, `rc2.d`, `rc3.d`, and `rc4.d` subdirectories for use by initialization scripts, as described in *14.1 SRP Startup and Shutdown Processing*.

8.1.2 The `admin` Service

The `admin` service associates HP-UX users with an RBAC role that has authorization to administer the compartment. By default, this authorization enables the administrator to start and stop the compartment.

8.1.2.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

Unix usernames for compartment administrator	HP-UX user names separated by “,” for the SRP administrator. These user names must already exist in the HP-UX user database (<code>/etc/password</code>). Variable Name: <code>admin_user</code> . Default: <code>root</code>
--	---

8.1.2.2 Configuration Data

The `admin` service uses RBAC to add information about the administrator in the RBAC configuration directory, `/etc/rbac`.

The `admin` service performs the following tasks:

- Creates a role with the name `SRPadmin-srp_name` for the compartment. SRP uses the `roleadm add` command to perform this task.
- Creates an authorization with the name `hpux.SRPadmin-srp_name` with the object set to the compartment. SRP uses the `authadm add` command to perform this task.
- Assigns the authorization `hpux.SRPadmin.srp_name` to the role `SRPadmin-srp_name`. SRP uses the `authadm assign` command to perform this task.
- Associate the specified HP-UX user name to the role `SRPadmin-srp_name`. The user name must already exist in the HP-UX user database. SRP uses the `roleadm assign` command to perform this task.
- Assigns `hpux.SRPadmin-srp_name` the authorization to execute the SRP master startup script `/opt/hpsrp/bin/srp_rc` in the compartment. This enables the administrator to start up and shut down the compartment. SRP uses the `cmdprivadm add` command to perform this task.

Configuring an administrative user does not grant that user login access to the compartment. A user does not have to be logged in to an SRP compartment to start or stop the compartment, or to modify the configuration data.

To specify the users authorized to log in to the compartment, use the SRP `login` service or the `authadm` command.

8.1.3 The `prm` Service

The `prm` Service creates a new PRM group for an SRP compartment. SRP does not allow you to add an SRP compartment to an existing PRM group. To add an SRP compartment to an existing PRM

group, use the procedure is described in *HP Process Resource Manager User's Guide*, "Assigning secure compartments to PRM groups."

8.1.3.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

PRM Group Name	Name for the PRM group. Variable Name: <code>prm_group_name</code> . Default: The SRP compartment name.
PRM Group Type (FSS or PSET)	Specifies the type of PRM group. A <i>Fair Share Scheduler (FSS)</i> PRM group uses a CPU entitlement mechanism that is specified in shares. An FSS group uses the Fair Share Scheduler (FSS) in the HP-UX kernel within the system's default processor set. A <i>Processor Set (PSET)</i> PRM group uses a CPU entitlement mechanism that is specified by assigning a subset of the system's cores to the PRM group. (A core is the actual data-processing engine within a processor. A single processor might have multiple cores. Processes in a PSET have equal access to CPU cycles on their assigned cores through the HP-UX standard scheduler. Variable Name: <code>prm_group_type</code> . Default: FSS.
PRM FSS group CPU shares	(Valid for FSS groups only) The number of CPU shares allocated for this group. PRM determines the actual amount of CPU allocated for this group by calculating the number of shares allocated for this group divided by the total CPU shares allocated. Variable Name: <code>prm_cpu_shares</code> . Default: 10.
PRM FSS group CPU cap	(Valid for FSS groups only) The maximum percentage of CPU available for this group. Variable Name: <code>prm_cpu_max</code> . Default: No cap.
PRM FSS group cores	(Valid for PSET groups only) The number of core processors allocated for this group. Variable Name: <code>prm_cores</code> . Default: 1.
PRM group memory shares	The number of memory shares to allocate to this compartment from the available system's memory for user processes. PRM determines the actual amount of memory allocated for this group by calculating the number of shares allocated for this group divided by the total memory shares allocated. Variable Name: <code>prm_mem_shares</code> . Default: 10.
PRM group memory cap	Specifies a max (upper bound) for memory consumption of system's memory for user processes. The value is expressed as a percent, which is an integer value, ranging from the percentage determined by the group's number of memory shares to 100. Variable Name: <code>prm_mem_max</code> Default: No cap
PRM group	The amount of physical memory allocated to this group for shared memory. This

physical memory value is specified in megabytes.
Variable Name: `prm_phys_mem`.
Default: 0 (no dedicated physical shared memory).

8.1.3.2 Configuration Data

By default, SRP creates a new PRM group using the SRP compartment name as the PRM group name. By default, the PRM group information is stored in the `/etc/prmconf` file. You can change the filename by running the `srp_setup` utility, as described in [2 Setting Up an SRP](#).

8.1.4 The network Service

The network service configures an IP interface for the SRP compartment.

You do not have to use a dedicated network interface card for the compartment; you can create a logical IP interface on a network interface card.

8.1.4.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in [15.1 Creating an SRP Compartment or Adding Data to an SRP](#).

IP address	Specifies the IP address for the compartment. You must specify an IP address not currently configured on the system. Variable Name: <code>ip_address</code> . Valid Input: An IPv4 address in dotted-decimal notation or an IPv6 address in colon-hexadecimal notation. Default: None.
Network interface name	Specifies the primary or secondary network interface name for the IP interface. A primary network interface name has no IP index number or an IP index number set to 0 (such as <code>lan2</code> or <code>lan2:0</code>). A secondary network name has a non-zero IP index number (such as <code>lan2:1</code>). Secondary interfaces share the same physical network interface as the corresponding primary interface. If you specify a primary network interface that is already configured for IP, SRP configures a secondary interface for you. Variable Name: <code>iface</code> . Default: None.
IP subnet mask	(Valid for IPv4 addresses only) Specifies the subnet mask for the interface, in dotted-decimal notation. Variable Name: <code>ip_mask</code> . Default: The network mask for the address class, as specified in the RFC 791 IETF specification.
Default gateway IP Address	IP address to use for routing network packets for the configured compartment IP address. The value is an IPv4 address in dotted-decimal notation or an IPv6 address in colon-hexadecimal notation. Variable Name: <code>gw_ip_address</code> Default: <code>ip_address</code> (the IP address configured for this compartment)
Assign IP address at SRP start time?	Configure the IP address in network configuration file (<code>/etc/rc.config.d/netconf</code> or <code>netconf-ipv6</code>). Set this value to <code>no</code> if the IP address assignment is managed by another subsystem, such as Serviceguard. Variable Name: <code>assign_ip</code>

Default: (yes)

8.1.4.2 Configuration Data

SRP configures IP interface information for the HP-UX Transport subsystem, the initialization and shutdown service, and for the compartment, as described in the sections that follow.

HP-UX Transport

If you specify an IP address that is not already configured for the system, SRP also configures the IP interface information for the HP-UX Transport subsystem as follows:

IPv4 Address

If you specify an IPv4 address, SRP adds configuration data to the `/etc/rc.config.d/netconf` file.

If you specify a primary interface name, SRP assigns the address to the primary interface if the primary interface does not already have an IP address configured. If the primary interface already has an IP address configured, SRP creates a new secondary (logical) interface using the next available IP index number for that primary interface. If you specify a secondary interface name and the corresponding primary interface is not already configured, SRP displays an error message.

SRP adds the interface to the `/etc/rc.config.d/netconf` file with the `INTERFACE_STATE` set to `down`. The interface is brought up when the SRP is started as described in *Network Initialization and Shutdown Service*.

Route Information

SRP provides an option to add or modify the default gateway routing table entry for the SRP IP address. The compartment IP address is always used as the source IP address.

If no target default gateway IP address is provided, the SRP IP address is used, with a hop (route) count set to 0. If a target default gateway IP address is provided, the hop (route) count is set to 1. The SRP adds the routing configuration data to `/etc/rc.config.d/netconf` (for IPv4) and `/etc/rc.config.d/netconf-v6` (for IPv6).

IPv6 Address

If you specify an IPv6 address, SRP adds configuration data to the `/etc/rc.config.d/netconf-ipv6` file.

If you specify an address that is not a link local address and a primary interface that does not already have an IP address configured, SRP configures the primary interface with the appropriate link local address, then it configures a secondary interface with the specified IPv6 address.

SRP adds the interface to the `/etc/rc.config.d/netconf-ipv6` file with the `INTERFACE_STATE` set to `down`. The interface is brought up when the SRP is started, as described in *Network Initialization and Shutdown Service*.

Network Initialization and Shutdown Service

SRP creates the file `/var/hpsrp/srp_name/sbin/init.d/srp_net` to bring the IP interface up and down.

This script also adds or deletes the default gateway route for the compartment interface. This script is executed when the SRP is started and stopped.

The `/var/hpsrp/srp_name/sbin/init.d/srp_net` file is linked to `/var/hpsrp/srp_name/sbin/init.d/rc2.d/S340srp_net` and `/var/hpsrp/srp_name/sbin/init.d/rc1.d/K660srp_net`.

For more information about SRP initialization and shutdown scripts, see *14 Starting and Stopping SRP Compartments*.

Security Containment Compartment

SRP adds a network interface rule for the IP address to the compartment rule file (`/etc/cmpt/srp_name.rules`). This allows the SRP access to its IP address.

8.1.5 The init Service

The `init` service creates startup and shutdown scripts for the compartment, and an SRP-specific `/var/hpsrp/<srp name>/sbin/init.d` directory structure that replicates the `/sbin/init.d` directory structure. SRP also configures the `autostart` feature for the SRP so that the system startup and shutdown scripts automatically execute the SRP startup and shutdown scripts.

8.1.5.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

<code>Autostart SRP</code>	Specifies if you want the SRP to be started at system boot time.
<code>at system boot</code>	Variable Name: <code>autostart</code> .
	Default: <code>yes</code> .

8.1.5.2 Configuration Data

SRP configures the following data:

- SRP adds the following entries to the `/etc/rc.config.d/srpconf` file to enable the `autostart` feature for the compartment:

```
SRP_NAME[n]="srp_name"  
START_SRP[n]=1
```

Where `n` is a unique index number and `srp_name` is the name of the SRP compartment.

- SRP creates `\ SRP-specific init` subdirectories below the `/var/hpsrp/srp_name/sbin` that contain startup and shutdown scripts. For more information about the directory structure, files, and how they are executed at system startup and shutdown time, see *14 Starting and Stopping SRP Compartments*.

8.1.6 The login Service

The `login` service enables you to specify the set of HP-UX users and HP-UX user groups whose members are authorized to log in to the SRP compartment. If you do not configure the `login` service and you are using the default RBAC system configuration, only the `root` user is authorized to log in to the compartment.

You can use the `login` service to grant non-root users the authorization to log in to the compartment.

8.1.6.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

Unix groups for
compartment login Name of the HP-UX user groups separated by “,” whose members are authorized to log in to the SRP compartment. These groups must already exist in a HP-UX groups database (such as `/etc/group`).
Variable Name: `login_group`.
Default: `adm`.

Unix users for
compartment login Name of the HP-UX users separated by “,” authorized to log in to the SRP compartment. These users must already exist in an HP-UX users database (such as `/etc/password`).
Variable Name: `login_user`.
Default: `None`.

8.1.6.2 Configuration Data

The `login` service controls login access to the compartment using the Security Containment compartment login feature. It uses RBAC authorizations to allow specified Unix users and group members to pass PAM authentication in the module `pam_hpsec`, which controls PAM-enabled authentication services (used by `login`, `ftp`, and other user session services) occurring within the SRP compartment.

The `login` service performs the following tasks:

- Creates the role `SRPlogin-srp_name`. SRP uses the `roleadm add` command to perform this task.
- Assigns the specified user or group ID to the `SRPlogin-srp_name` role. SRP uses the `roleadm assign` command to perform this task.
- Assigns the `SRPlogin-srp_name` role login authorization (the authorization `hpux.security.compartment.login`) for the compartment. SRP uses the `authadm` command to perform this task.

8.1.7 The ipfilter Service

The `ipfilter` service configures HP-UX IPFilter for the compartment. The base SRP IPFilter configuration allows the following packets to pass:

- All outbound packets from the compartment IP address
- Inbound TCP, UDP, and ICMP responses to packets sent from the compartment IP address.
- All inbound ICMP packets to the compartment IP address.

All other inbound packets are blocked.

You can also configure IPFilter to allow inbound and outbound IPsec packets to pass.

8.1.7.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

Add IPFilter rules for
IPsec? Specifies whether or not you want to add IPFilter rules to allow IPsec packets to pass.
Variable Name: `ipf_for_ipsec`.

Valid Input: yes or no.

Default: no.

8.1.7.2 Configuration Data

If the compartment address is an IPv4 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf.conf` file. If the compartment address is an IPv6 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf6.conf` file.

SRP adds the following IPFilter rules for the compartment, where `cmpt_address` is the compartment IP address:

- Rules that allow all TCP, UDP, and ICMP outbound packets from the compartment address. These rules specify the `keep` state keywords to allow inbound replies for these packets:

```
pass out quick proto tcp from cmpt_address to any keep state
pass out quick proto udp from cmpt_address to any keep state
pass out quick proto icmp from cmpt_address to any keep state
```

If the compartment address is an IPv6 address, the last rule is `pass out quick proto icmpv6 from cmpt_address to any keep state`.

- A rule that allows inbound ICMP packets from any address to the compartment IP address:

```
pass in quick proto icmp from any to cmpt_address
```

If the compartment address is an IPv6 address, the rule is `pass in quick proto icmpv6 from any to cmpt_address`.

- A rule that blocks all inbound packets to the compartment IP address:

```
block in quick from any to cmpt_address
```

Rule Order and Selection

By default, IPFilter selects a rule for a packet by reading the rules in a configuration file from top to bottom and selects the last rule that matches a packet. The `quick` keyword changes this behavior and causes IPFilter to immediately apply the rule to a packet if it matches the filter (instead of continuing to evaluate rules for the packet). When using the `quick` keyword, rules are generally ordered from most specific to least specific.

SRP specifies the `quick` keyword in the IPFilter rules it configures. SRP inserts these rules at the top of the IPFilter configuration file in the order shown.

8.1.7.3 IPFilter Rules

If you specify that you want to add IPFilter rules for IPsec, SRP also adds IPFilter rules that allow IPsec Encapsulating Security Payload (ESP; protocol 50) and Authentication Header (AH; protocol 51) packets and IPsec control packets (Internet Key Exchange, or IKE; UDP port 500) to pass. These rules are inserted above the more general IPFilter rules for the compartment. For more information, see *Using IPsec with IPFilter*.

8.1.8 The ipsec Service

The `ipsec` service configures HP-UX IPsec to encrypt and authenticate IP packets between the compartment IP address and a remote IP address.

8.1.8.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

<code>IPsec peer IP address</code>	The destination, or remote IP address for the IPsec policies. Variable Name: <code>ipsec_peer_addr</code> . Valid Input: An IPv4 address in dotted-decimal notation or an IPv6 address in colon-hexadecimal notation. Default: None.
<code>IPsec transform</code>	The transform for the IPsec host policy. This must be compatible with the transform configured on the peer system. Variable Name: <code>ipsec_transform</code> . Valid Input: ESP_AES128_HMAC_SHA1 ESP_AES128_HMAC_MD5 ESP_3DES_HMAC_SHA1 ESP_3DES_HMAC_MD5 ESP_NULL_HMAC_SHA1 ESP_NULL_HMAC_MD5 Default: <code>ESP_AES128_HMAC_SHA1</code>
<code>IPsec preshared key</code>	The preshared key used to authenticate the identity of the IPsec peer. This must match the value configured on the peer system. Parameter Name: <code>ipsec_psk</code> . Valid value: A text string, containing 1 - 128 ASCII characters (whitespaces are not allowed). Default: None.

8.1.8.2 Configuration Data

SRP adds IPsec configuration data using the `ipsec_config` utility. IPsec adds the data to the IPsec database, `/var/adm/ipsec/config.db`. To view the contents of the IPsec database, use the `ipsec_config` or the `ipsec_report` utility. To modify the contents of the IPsec database, you must use the `ipsec_config` utility.

SRP adds the following IPsec configuration data:

- A host IPsec policy

The host policy specifies encryption and authentication using the specified transform between the specified remote IP address and the local (compartment) address. The default HP-UX IPsec values are used for all other parameters.
- An Internet Key Exchange (IKE) policy

The IKE policy specifies parameters used to establish an IKE security association with the specified remote IP address. The authentication method is PSK (preshared key). The default HP-UX IPsec values are used for all other parameters.
- An authentication record

The authentication record contains the specified remote IP address and preshared key value. The default HP-UX IPSec values are used for all other parameters.

HP-UX IPSec Default Parameter Values

For IPSec parameters that SRP does not prompt for, SRP uses the IPSec default values in the configuration records. The IPSec default values are read from the default IPSec profile file, `/var/adm/ipsec/.ipsec_profile`. You can view this text file to determine the default IPSec parameters and determine what values need to be configured on the peer system. Some of the main parameters and the default values set in the factory-installed profile file are as follows:

- IKE exchange type: Main Mode
- IKE hash algorithm: MD5
- IKE encryption algorithm: 3DES
- IKE Diffie-Hellman group: 2

Policy Selection and Priority

When IPSec selects policies, it selects the first policy that matches the search criteria. Because of this selection algorithm, IPSec policies are typically ordered from most specific to least specific. SRP adds the policies using the IPSec automatic priority increment mechanism, where IPSec determines the priority for a new policy by adding n to the current highest priority for that policy category, where n is the automatic priority increment value. When a policy is added with this mechanism, it becomes the last policy evaluated before the default policy in the category; you might have to modify the priority value for your policies.

Using IPSec with IPFilter

HP-UX IPFilter is located below HP-UX IPSec in the networking stack. HP-UX IPFilter processes inbound IP packets before HP-UX IPSec and processes outbound packets after HP-UX IPSec.

To use IPSec with IPFilter, you must configure IPFilter to pass the following packets:

- IP packets with protocol 50 (IPsec Encapsulating Security Payload protocol, ESP)
- IP packets with protocol 51 (IPsec Authentication Header protocol, AH)
- UDP packets with port 500 (IPsec Internet Key Exchange protocol, IKE)

If HP-UX IPSec secures a packet (the packet has an AH or ESP header), HP-UX IPFilter cannot filter the packet based on upper layer information, such as TCP port numbers and connection states, and ICMP message types. The only upper-layer protocol information that HP-UX IPFilter processes is the IP protocol number. IPSec packets do not match any IPFilter rules based on the TCP, UDP, or ICMP protocol type or based on field values for these protocols (such as port numbers).

8.1.9 Completing the Configuration

After you configure a `base` compartment, you can apply an application template to add application-specific configuration data. For more information, see *8 Using the base Template*, *9 Using the apache Template*, *10 Using the tomcat Template*, and *11 Using the custom Template*.

8.2 Replacing or Deleting Base SRP Data

Use the following command to replace `base` template data from an SRP compartment:

```
srp -r[ep]lace srp_name -t base [-s service[,service]...]
```

The `srp -replace` command deletes the specified data, then prompts you for replacement data. For example, the following command deletes all PRM data for the base template, then prompts you for replacement data:

```
srp -replace mySRP -t base -s prm
```

Use the following command to delete base template data from an SRP compartment:

```
srp -d[ele]te] srp_name -t base [-s service[,service]...]
```

CAUTION: If you do not specify the `-template` and/or `-service` arguments, `srp` deletes all templates and/or services for the compartment. For example, the command `srp -delete mySRP` deletes the entire `mySRP` SRP compartment.

For more information, see *15.2 Deleting Configuration Data* and *15.3 Replacing Configuration Data*.

9 Using the apache Template

This chapter describes how to use the `apache` template to configure and provision an HP-UX Apache-based Web Server in an SRP compartment. You can also use the `apache` template to delete or modify the `apache` template data for a compartment.

This chapter addresses the following topics:

- 9.1 Adding the apache Template to an SRP Compartment
- 9.2 Replacing or Deleting Apache SRP Data

9.1 Adding the apache Template to an SRP Compartment

To use the `apache` template, you must create a base SRP compartment first, then use the `srp -add` command to add the `apache` template to the compartment.

For example:

```
# srp -add mySRP # create a base SRP compartment
# srp -add mySRP -template apache
```

The syntax for adding the `apache` template to an SRP compartment is as follows:

```
srp -a[dd] srp_name -t[emplate] apache [-s[ervice] service[,service]....]
```

Where:

srp_name

Specifies the name of an existing SRP compartment.

Specifies the name of the service to configure. The following services are valid with the `apache` template:

- `cmpt`
- `ipfilter`
- `provision`

service

If you do not specify any services in the command line, `srp` prompts you for the services you want to apply and displays a list of the default services that are valid with the `apache` template. If you are using the factory-configured default services, the only valid default service is `cmpt,provision`.

The input data for these services and the data configured are described in the sections that follow. If SRP uses input data for multiple services, the `srp` utility prompts you for the data once and reuses the value.

9.1.1 The `cmpt` Service

The `cmpt` service for the `apache` template configures Security Containment file system rules to allow the compartment to access the specified Apache directories.

9.1.1.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in 15.1 *Creating an SRP Compartment or Adding Data to an SRP*.

Apache Web
Server Suite
Version

The HP-UX Webserver Suite version of Apache to be used to configure the
template
Variable Name: `wss_version`.

Default: 3.0.

Apache data path The root directory for Apache data. The `cmpt` service adds rules to allow the compartment all access to this directory. Users and processes in the SRP compartment can read, write, traverse (`nsearch`), and delete (`ulink`) the contents of these directories.
Variable Name: `data_path`.
Default: `/var/hpsrp/srp_name/opt/hpws22/apache`.

Apache executable path The root directory for Apache executables. The `cmpt` service adds rules to allow the compartment read access to this directory.
Variable Name: `exec_path`.
Default: `/opt/hpws22/apache`.

9.1.1.2 Configuration Data

SRP adds entries to the SRP compartment rules file (`/etc/cmpt/srp_name.rules`) that authorize access to the `exec_path` and `data_path` directories. SRP also adds an `include` statement to add the rules from the `/opt/hpsrp/etc/cmpt/apache.srp_incl` file. As delivered by HP, this file is empty. You can edit this file to contain compartment rules to be applied when configuring the `cmpt` service with the `apache` template.

9.1.2 The ipfilter Service

The `ipfilter` service for the `apache` template adds rules to allow inbound requests to the specified ports used by the Apache server to pass. You can also specify additional inbound destination TCP port numbers for IPFilter pass rules.

9.1.2.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in [15.1 Creating an SRP Compartment or Adding Data to an SRP](#).

Apache HTTP port number Specifies the TCP port number on which the compartment Apache server will receive HTTP requests.
Variable Name: `http_port`.
Valid Input: A TCP port number in the range 1- 65535.
Default: 80, the IANA registered port number for HTTP.

Apache HTTPS port number Specifies the TCP port number on which the compartment Apache server will receive HTTPS (SSL) requests.
Variable Name: `https_port`.
Valid Input: A TCP port number in the range 1- 65535.
Default: 443, the IANA registered port number for HTTPS.

Tomcat AJP port number Specifies the TCP port number on which the compartment Apache server will send requests to a Tomcat server.
Variable Name: `ajp_port`.
Valid Input: A TCP port number in the range 1- 65535.
Default: 8009.

IPFilter Port Numbers Specifies the local TCP port numbers for IPFilter rules that allow inbound packets.

Variable Name: `ipf_tcp_ports`.

Valid Input: One or more TCP port numbers each in the range 1- 65535, separated by commas.

Default: 80, 443. These are the IANA registered port numbers for HTTP and HTTPS (SSL).

9.1.2.2 Configuration Data

If the compartment address is an IPv4 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf.conf` file. If the compartment address is an IPv6 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf6.conf` file.

SRP configures rules that allow inbound packets from any remote IP address to the compartment IP address with the specified destination TCP port numbers. SRP inserts these rules at the top of the IPFilter rules file and uses the `quick` keyword. The IPFilter configuration file already contains rules from the `base` template to allow all outbound TCP, UDP, and ICMP packets from the compartment IP address, as described in *Configuration Data*.

9.1.3 The `provision` Service

The `provision` service executes the script `/opt/hpsrp/bin/util/apache_setup` to provision (deploy) an apache service in the SRP compartment.

9.1.3.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

Apache Web Server Suite Version	The HP-UX Webserver Suite version of Apache to be used to configure the template Variable Name: <code>wss_version</code> . Default: 3.0.
Copy Apache data from path	The directory from which you want to copy Apache data. The <code>provision</code> service creates a copy of this subtree and its contents and installs it in the specified <code>data_path</code> for use by the SRP compartment. The input for this variable is typically the <code>newconfig</code> subdirectory under the Apache product directory. Variable Name: <code>data_src</code> . Default: <code>/opt/hpws22/apache/newconfig</code> .
Apache data path	The target directory for the copied Apache data. Variable Name: <code>data_path</code> . Default: <code>/var/hpsrp/srp_name/opt/hpws22/apache</code> .
Apache user name	Specifies the Unix user name under which the Apache processes in this compartment will run. Variable Name: <code>user</code> . Default: <code>www</code> .
Apache HTTP port number	Specifies the TCP port number on which the compartment Apache server will receive HTTP requests. Variable Name: <code>http_port</code> . Valid Input: A TCP port number in the range 1- 65535. Default: 80, the IANA registered port number for HTTP.

Apache HTTPS port number	Specifies the TCP port number on which the compartment Apache server will receive HTTPS (SSL) requests. Variable Name: <code>https_port</code> . Valid Input: A TCP port number in the range 1- 65535. Default: 443, the IANA registered port number for HTTPS.
Tomcat AJP port number	Specifies the TCP port number on which the compartment Apache server will send request to Tomcat server. Variable Name: <code>ajp_port</code> . Valid Input: A TCP port number in the range 1- 65535. Default: 8009.
Start Apache at SRP start time	Specifies if you want to SRP to add a script to the SRP <code>init</code> directory structure to start and stop Apache. The script is automatically executed when the SRP is started or stopped. Variable Name: <code>start_apache</code> . Valid Input: <code>yes</code> or <code>no</code> . Default: <code>yes</code> .
Start Apache in ssl mode	Specifies if you want to start Apache in ssl mode. Variable Name: <code>startssl_apache</code> Valid Input: <code>yes</code> or <code>no</code> Default: <code>no</code>

9.1.3.2 Configuration Data

By default, the tasks executed by the `/opt/hpsrp/bin/util/apache_setup` script include:

- Creating `bin`, `cgi-bin`, `conf`, `htdocs`, and `logs` subdirectories below the compartment Apache home directory.
- Creating a compartment-specific `http.conf` file with compartment-specific configuration data, such as setting data paths to the appropriate directories below the compartment Apache home directory and setting the IP address to the compartment IP address. Enables the `mod_ajp` module for apache tomcat integration.
- Creating compartment-specific initialization scripts and startup file to start Apache with the compartment-specific `http.conf` file when the compartment startup script is executed. The setup script:
 - Modifies the compartment-specific `apachectl` file in the `bin` subdirectory below the `data_path` (by default, the path is `/var/hpsrp/srp_name/opt/hpws22/apache/bin/apachectl`) to use the compartment-specific Apache data path directory as the `ServerRoot`.
 - Creates the compartment-specific startup configuration file, `/var/hpsrp/srp_name/etc/rc.config.d/hpws22_apacheconf`, which specifies the compartment-specific Apache home directory.
 - Adds the startup and shutdown script `hpws22_apache` to the compartment-specific `init.d` directory, `/var/hpsrp/srp_name/sbin/init.d`. This file is linked to the `/var/hpsrp/srp_name/sbin/rc3.d/S823hpws22_apache` and `/var/hpsrp/srp_name/sbin/rc3.d/K177hpws22_apache` files.

9.1.3.3. Completing the Configuration

After you apply the `apache` `cmt` service and the default `apache` provisioning script, you can start the SRP compartment, and have a fully-functional HP-UX Apache-based Web Server in the compartment. You can further customize the Web Server as needed by editing the compartment-specific Apache configuration files

(`/var/hpsrp/srp_name/etc/rc.config.d/hpws22_apacheconf` and the compartment-specific `apachectl` file, located in the `bin` subdirectory below the `data_path`).

9.2 Replacing or Deleting Apache SRP Data

Use the following command to replace `apache` template data from an SRP compartment:

```
srp -r[ep]lace srp_name -t apache [-s service[,service]....]
```

The `srp -replace` command deletes the specified data, then prompts you for replacement data. For example, the following command deletes all the IPFilter data for the `apache` template, then prompts you for replacement data:

```
srp -replace mySRP -t apache -s ipfilter
```

Use the following command to delete `apache` template data from an SRP compartment:

```
srp -d[ele]te srp_name -t apache [-s service[,service]....]
```

CAUTION: If you do not specify the `-template` and/or `-service` arguments, `srp` deletes all templates and/or services for the compartment. For example, the command `srp -delete mySRP` deletes the entire `mySRP` SRP compartment.

For more information, see [15.2 Deleting Configuration Data](#) and [15.3 Replacing Configuration Data](#).

10 Using the `tomcat` Template

This chapter describes how to use the `tomcat` template to add configuration data for hosting an HP-UX Tomcat servlet engine in an SRP compartment. You can also use the `tomcat` template to delete or modify the `tomcat` template data for a compartment.

This chapter addresses the following topics:

- 10.1 Adding the `tomcat` Template to an SRP Compartment
- 10.2 Replacing or Deleting Tomcat SRP Data

10.1 Adding the `tomcat` Template to an SRP Compartment

To use the `tomcat` template, you must create a base SRP compartment first, then use the **`srp -add`** command to add the `tomcat` template to the compartment.

For example:

```
# srp -add mySRP # create a base SRP compartment
# srp -add mySRP -template tomcat
```

The syntax for adding the `tomcat` template to an SRP compartment is as follows:

```
srp -a[dd] srp_name -t[emplate] tomcat [-s[ervice] service[,service]...]
```

Where:

`srp_name`

Specifies the name of an existing SRP compartment.

Specifies the name of the service to configure. The following services are valid with the `tomcat` template:

- `cmpt`
- `ipfilter`
- `provision`

`service`

If you do not specify any services in the command line, `srp` prompts you for the services you want to apply and displays a list of the default services that are valid with the `tomcat` template. If you are using the factory-configured default services, the only valid default service is `cmpt`, `provision`.

The input data for these services and the data configured are described in the sections that follow. If SRP uses input data for multiple services, the `srp` utility prompts you for the data once and reuses the value.

10.1.1 The `cmpt` Service

The `cmpt` service for the `tomcat` template configures Security Containment file system rules to allow the compartment to access the specified Tomcat directories.

10.1.1.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in 15.1 *Creating an SRP Compartment or Adding Data to an SRP*.

Tomcat Web Server Suite Version The HP-UX Webserver Suite version of Tomcat Servlet Engine to be used to configure the template

Variable Name: `wss_version`.

Default: `3.0`.

`Tomcat data path` The root directory for Tomcat data. The `cmpt` service adds rules to allow the compartment all access to this directory. Users and processes in the SRP compartment can read, write, traverse (`nsearch`), and delete (`unlink`) the contents of these directories.

Variable Name: `data_path`.

Default: `/var/hpsrp/srp_name/opt/hpws22/tomcat`.

`Tomcat executable path` The root directory for Tomcat executables. The `cmpt` service adds rules to allow the compartment read access to this directory.

Variable Name: `exec_path`.

Default: `/opt/hpws22/tomcat`.

`Java Home Path` The java home path

Variable Name: `java_path`

Default: `/opt/java1.5`

10.1.1.2 Configuration Data

SRP adds entries to the SRP compartment rules file (`/etc/cmpt/srp_name.rules`) that authorize access to the `exec_path`, `data_path`, and `java_path` directories. SRP also adds an include statement to add the rules from the `/opt/hpsrp/etc/cmpt/tomcat.srp_incl` file. As delivered by HP, this file is empty. You can edit this file to contain compartment rules to be applied when configuring the `cmpt` service with the `tomcat` template.

10.1.2 The ipfilter Service

The `ipfilter` service for the `tomcat` template adds rules to allow inbound requests to the specified ports used by the Tomcat server to pass. You can also specify additional inbound destination TCP port numbers for IPFilter pass rules.

10.1.2.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in [15.1 Creating an SRP Compartment or Adding Data to an SRP](#).

`Tomcat Control port` Specifies the TCP port number on which the compartment Tomcat server will receive request from an Apache webserver.

Variable Name: `control_port`.

Valid Input: A TCP port number in the range 1- 65535.

Default: `8005`

`Tomcat HTTP port number` Specifies the TCP port number on which the compartment Tomcat server will receive HTTP requests for servlets.

Variable Name: `http_port`.

Valid Input: A TCP port number in the range 1- 65535.

Default: `8081`.

`Tomcat AJP port number` Specifies the TCP port number on which the compartment Tomcat server will receive request from an Apache webserver.

Variable Name: `ajp_port`.

Valid Input: A TCP port number in the range 1- 65535.
Default: 8009.

IPFilter Port Numbers Specifies the local TCP port numbers for IPFilter rules that allow inbound packets.
Variable Name: `ipf_tcp_ports`.
Valid Input: One or more TCP port numbers each in the range 1- 65535, separated by commas.
Default: 8085,8081,8009

*https port is disabled by default in tomcat.

10.1.2.2 Configuration Data

If the compartment address is an IPv4 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf.conf` file. If the compartment address is an IPv6 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf6.conf` file.

SRP configures rules that allow inbound packets from any remote IP address to the compartment IP address with the specified destination TCP port numbers.

SRP inserts these rules at the top of the IPFilter rules file and uses the `quick` keyword.

The IPFilter configuration file already contains rules from the `base` template to allow all outbound TCP, UDP, and ICMP packets from the compartment IP address, as described in *Configuration Data*.

10.1.3 The provision Service

The `provision` service executes the script `/opt/hpsrp/bin/util/tomcat_setup` to provision (deploy) a `tomcat` service in the SRP compartment.

10.1.3.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*

Tomcat Web Server Suite Version The HP-UX Webserver Suite version of Tomcat Servlet Engine to be used to configure the template
Variable Name: `wss_version`.
Default: 3.0.

Tomcat executable path The root directory for Tomcat executables.
Variable Name: `exec_path`.
Default: `/opt/hpws22/tomcat`.

Copy Tomcat data from path The directory from which you want to copy Tomcat data. The `provision` service creates a copy of this subtree and its contents and installs it in the specified `data_path` for use by the SRP compartment. The input for this variable is typically the `newconfig` subdirectory under the Tomcat product directory.
Variable Name: `data_src`.
Default: `/opt/hpws22/tomcat/newconfig`.

Tomcat data path The target directory for the copied Tomcat data.
Variable Name: `data_path`.

Default: `/var/hpsrp/srp_name/opt/hpws/tomcat.`

Java Home Path	The java home path Variable Name: <code>java_path</code> Default: <code>/opt/java1.5</code>
Tomcat user name	Specifies the Unix user name under which the Tomcat processes in this compartment will run. Variable Name: <code>user.</code> Default: <code>www.</code>
Tomcat HTTP port number	Specifies the TCP port number on which the compartment Tomcat server will receive HTTP requests. Variable Name: <code>http_port.</code> Valid Input: A TCP port number in the range 1- 65535. Default: <code>8081.</code>
Tomcat AJP port number	Specifies the TCP port number on which the compartment Tomcat server will receive request from apache webserver. Variable Name: <code>ajp_port.</code> Valid Input: A TCP port number in the range 1- 65535. Default: <code>8009.</code>
Tomcat Control port	Specifies the TCP port number on which the compartment Tomcat server will receive request from apache webserver. Variable Name: <code>control_port.</code> Valid Input: A TCP port number in the range 1- 65535. Default: <code>8005</code>
Start Tomcat at SRP start time	Specifies if you want to SRP to add a script to the SRP <code>init</code> directory structure to start and stop Tomcat. The script is automatically executed when the SRP is started or stopped. Variable Name: <code>start_tomcat.</code> Valid Input: <code>yes</code> or <code>no.</code> Default: <code>yes.</code>

10.1.3.2 Configuration Data

By default, the tasks executed by the `/opt/hpsrp/bin/util/tomcat_setup` script include:

- Creating `conf`, `logs`, `temp`, `webapps` and `work` subdirectories below the compartment Tomcat home directory.
- Creating a compartment-specific `server.xml` file with compartment-specific configuration data, such as setting tomcat http, ajp ports.
- Creating compartment-specific initialization scripts and startup file to start Tomcat with the compartment-specific configuration files when the compartment startup script is executed. The setup script:
 - Modifies the initialization scripts to start/stop tomcat application as the tomcat user. Also, exported variables that define tomcat's `CATALINA_HOME`, `CATALINA_BASE` and `JAVA_HOME` directory.

- o Creates the compartment-specific startup configuration file, `/var/hpsrp/srp_name/etc/rc.config.d/hpws22_tomcatconf`, which specifies the compartment-specific tomcat home directory.
- o Adds the startup and shutdown script `hpws22_tomcat` to the compartment-specific `init.d` directory, `/var/hpsrp/srp_name/sbin/init.d`. This file is linked to the `/var/hpsrp/srp_name/sbin/rc3.d/S823hpws22_tomcat` and `/var/hpsrp/srp_name/sbin/rc3.d/K177hpws22_tomcat` files. The

Completing the Configuration

After you apply the tomcat `cmpt` service and the default tomcat provisioning script, you can start the SRP compartment, and have a fully-functional HP-UX Tomcat-based servlet Engine in the compartment. You can further customize the servlet engine as needed by editing the compartment-specific Tomcat configuration files

(`/var/hpsrp/srp_name/etc/rc.config.d/hpws22_tomcatconf` and the compartment-specific `server.xml`, located in the `conf` subdirectory below the `data_path`).

10.2 Replacing or Deleting Tomcat SRP Data

Use the following command to replace tomcat template data from an SRP compartment:

```
srp -r[eplace] srp_name -t tomcat [-s service[,service]...]
```

The `srp -replace` command deletes the specified data, then prompts you for replacement data. For example, the following command deletes all the IPFilter data for the tomcat template, then prompts you for replacement data:

```
srp -replace mySRP -t tomcat -s ipfilter
```

Use the following command to delete tomcat template data from an SRP compartment:

```
srp -d[eleete] srp_name -t tomcat [-s service[,service]...]
```

CAUTION: If you do not specify the `-template` and/or `-service` arguments, `srp` deletes all templates and/or services for the compartment. For example, the command `srp -delete mySRP` deletes the entire `mySRP` SRP compartment.

For more information, see *15.2 Deleting Configuration Data* and *15.3 Replacing Configuration Data*.

11 Using the custom Template

The `custom` template enables you to specify additional Security Containment file access rules and IPFilter rules for an SRP compartment. You can use the `custom` template to accommodate additional applications in a SRP compartment, or to add compartment or IPFilter rules to increase security controls for an SRP compartment.

You can also use the `custom` template to delete or modify the `custom` template data for an SRP compartment.

This chapter addresses the following topics:

- 11.1 Adding the custom Template to an SRP Compartment
- 11.2 Replacing or Deleting Custom SRP Data

11.1 Adding the `custom` Template to an SRP Compartment

To use the `custom` template, you must create an SRP compartment first, then use the **`srp -add`** command to add the `custom` template to the compartment.

For example:

```
# srp -add mySRP # create a base SRP compartment
# srp -add mySRP -template custom -id myID
```

The syntax for adding the `custom` template to an SRP compartment is as follows:

```
srp -a[dd] srp_name -t[emplate] custom -i[d] instance [-s[ervice]
service[,service]...]
```

Where:

srp_name Specifies the name of an existing SRP compartment.

instance Unique string identifier used to identify an instance of an application of the custom template (the custom template can be added multiple times to the same SRP compartment).
Valid Input: A text string with alphanumeric, dash (-), or underscore (_) characters.
The maximum length is 20 characters.
Default: None.

service Specifies the name of the service to configure. The following services are valid with the `custom` template:

- `cmpt`
- `ipfilter`
- `provision`

If you do not specify any services in the command line, `srp` prompts you for the services you want to apply and displays a list of the default services that are valid with the `custom` template. If you are using the factory-configured default services, the only valid default service is `cmpt`.

The input data for these services and the data configured are described in the sections that follow. If SRP uses input data for multiple services, the `srp` utility prompts you for the data once and reuses the value.

11.1.1 The `cmpt` Service

The `cmpt` service for the `custom` template applies additional compartment rules to your compartment. You can specify a rules file to include and/or specify file system paths to configure for different access types.

11.1.1.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

Compartment rule files	Specifies compartment rule files to include in the compartment rules file for this SRP compartment. To specify multiple files, use commas to separate file names. Variable Name: <code>cmpt_rule_file</code> . Default: None.
Read access paths	Specifies directories to configure with read access (<code>nsearch</code> and <code>read</code>) in the compartment rules file for this SRP compartment. To specify multiple directories, use commas to separate directory names. Variable Name: <code>read_access</code> . Default: None.
All access paths	Specifies directories to configure with <code>all</code> access in the compartment rules file for this SRP compartment. To specify multiple directories, use commas to separate directory names. Variable Name: <code>all_access</code> . Default: None.
no access paths	Specifies directories to configure with <code>none</code> access in the compartment rules file for this SRP compartment. This will disallow access to the specified directories unless an additional access rule has been specified for this path from this SRP. To specify multiple directories, use commas to separate directory names. Variable Name: <code>no_access</code> . Default: None.

11.1.1.2 Configuration Data

SRP adds entries to the rules file for the SRP compartment to authorize access according to the descriptions in the previous sections. SRP also adds an `include` statement to add the rules from the files specified by `cmpt_rule_file`.

11.1.2 The `ipfilter` Service

The `ipfilter` service for the `custom` template enables you to allow inbound packets to specific TCP or UDP port numbers.

11.1.2.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

IPFilter TCP port numbers	Specifies the local TCP port numbers for IPFilter rules that allow inbound packets.
---------------------------	---

Variable Name: `ipf_tcp_ports`.

Valid Input: One or more TCP port numbers each in the range 1- 65535, separated by commas.

Default: None.

IPFilter UDP port numbers

Specifies the local UDP port numbers for IPFilter rules that allow inbound packets.

Variable Name: `ipf_udp_ports`.

Valid Input: One or more UDP port numbers each in the range 1- 65535, separated by commas.

Default: None.

11.1.2.2 Configuration Data

If the compartment address is an IPv4 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf.conf` file. If the compartment address is an IPv6 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf6.conf` file.

SRP configures rules that allow inbound packets from any remote IP address to the compartment IP address with the specified destination TCP or UDP port numbers.

SRP inserts these rules at the top of the IPFilter rules file and uses the `quick` keyword.

The IPFilter configuration file already contains rules from the `base` template to allow all outbound TCP, UDP, and ICMP packets from the compartment IP address, as described in [11.1.3.2 Configuration Data](#).

11.1.3 The provision Service

The `provision` service executes the customizable script `/opt/hpsrp/bin/util/custom_setup` to provision (deploy) an additional application in the SRP compartment.

11.1.3.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in [15.1 Creating an SRP Compartment or Adding Data to an SRP](#).

Provision Script name

The provision script path to use to configure additional set of applications .

Variable Name: `script`

Default: `/opt/hpsrp/bin/util/custom_setup`.

11.1.3.2 Configuration Data

By default, the `/opt/hpsrp/bin/util/custom_setup` script:

- Prints the script name used by the instance during the verbose listing.
- Prints the arguments to the script in the verbose mode for any operation. These arguments include:
 - **verbose**: Is set to 1 if verbose is enabled
 - **compartment**: SRP compartment name
 - **script**: current script name
 - **srp_id**: current instance id
 - **service**: selected services

- **srp**root: SRP root directory of the compartment
- Allows users to write their own functionality for each of the operations like add/delete/replace.

11.2 Replacing or Deleting Custom SRP Data

Use the following command to replace custom template data from an SRP compartment:

```
srp -r[eplace] srp_name -t custom [-s service[,service]...] id instance
```

The `srp -replace` command deletes the specified data, then prompts you for replacement data.

For example, the following command deletes all the IPFilter data for the custom template added with the id 2008-05-09, then prompts you for replacement data:

```
srp -replace mySRP -t custom -s ipfilter id 2008-05-09
```

Use the following command to delete custom template data from an SRP compartment:

```
srp -d[ele]te] srp_name -t custom [-s service[,service]...] id instance
```

CAUTION: If you do not specify the `-template` and/or `-service` arguments, `srp` deletes all templates and/or services for the compartment. For example, the command `srp -delete mySRP` deletes the entire `mySRP` SRP compartment.

For more information, see [15.2 Deleting Configuration Data](#) and [15.3 Replacing Configuration Data](#).

12 Using the oracledb Template

This chapter describes how to use the `oracledb` template to configure an SRP compartment to share a single set of Oracle executables with other SRPs. You do not need to use this template if you are installing a separate instance of the Oracle executables in the SRP.

You can also use the `oracledb` template to delete or modify the `oracledb` template data for a compartment.

12.1 Adding the oracledb Template to an SRP Compartment

To use the `oracledb` template, you must create a base SRP compartment first, then add the `oracledb` template to the compartment. For example:

```
srp -add mySRP # create a base SRP compartment
srp -add mySRP -template oracledb
```

The syntax for adding the `oracledb` template to an SRP compartment is as follows:

```
srp -a[dd] srp_name -t[emplate] oracledb [-s[ervice]
service[,service]...
```

Where:

srp_name Specifies the name of an existing SRP compartment.

service Specifies the name of the service to configure. The following services are valid with the `oracledb` template:

- `cmpt`
- `ipfilter`

If you do not specify any services in the command line, `srp` prompts you for the services you want to apply and displays a list of the default services that are valid with the `oracledb` template. If you are using the factory-configured default services, the only valid default service is `cmpt,provision`.

The input data for these services and the data configured are described in the sections that follow. If SRP uses input data for multiple services, the `srp` utility prompts you for the data once and reuses the value.

12.1.1 The `cmpt` Service

The `cmpt` service for the `oracledb` template configures Security Containment file system rules to allow the compartment to access the specified Oracle directories.

12.1.1.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

`Oracle executable path` The root directory for Oracle executables. The `cmpt` service adds rules to allow the compartment read access to this directory. Because this parameter is configured per compartment, you can select different versions of the Oracle Database server product on the system.

Variable Name: `exec_path`.
Default: `/opt/var/hpsrp/srp_name/opt/u01/home/oracle`.

Oracle DB data path The root directory for Oracle data. The `cmpt` service adds rules to allow the compartment all access to this directory. Users and processes in the SRP compartment can read, write, traverse (`nsearch`), and delete (`unlink`) the contents of these directories. In most cases, you would set up the Oracle configuration and schema under this path, and set the value of the `ORACLE_HOME` environment variable to this path.
Variable Name: `data_path`.
Default: `/var/hpsrp/srp_name/opt/u01/home/oracle`.

12.1.1.2 Configuration Data

SRP adds entries to the SRP compartment rules file (`/etc/cmpt/srp_name.rules`) that authorize access to the `exec_path` and `data_path` directories. SRP also adds an `include` statement to add the rules from the `/opt/hpsrp/etc/cmpt/oracledb.srp_incl` file. As delivered by HP, this file is empty. You can edit this file to contain compartment rules to be applied when configuring the `cmpt` service with the `oracledb` template.

12.1.2 The ipfilter Service

The `ipfilter` service for the `oracledb` template adds rules to allow inbound requests to the specified ports used by the Oracle database server to pass. You can also specify additional inbound destination TCP port numbers for IPFilter pass rules.

12.1.2.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in [15.1 Creating an SRP Compartment or Adding Data to an SRP](#)

IPFilter Port Numbers Specifies the local TCP port numbers for IPFilter rules that allow inbound packets.
Variable Name: `ipf_ports`.
Valid Input: One or more TCP port numbers each in the range 1- 65535, separated by commas.
Default: 1521. This is the default port number for the Oracle Net Listener process (commonly referred to as the listener).

12.1.2.2 Configuration Data

If the compartment address is an IPv4 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf.conf` file. If the compartment address is an IPv6 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf6.conf` file.

SRP configures rules that allow inbound packets from any remote IP address to the compartment IP address with the specified destination TCP port numbers.

SRP inserts these rules at the top of the IPFilter rules file and uses the `quick` keyword.

The IPFilter configuration file already contains rules from the `base` template to allow all outbound TCP, UDP, and ICMP packets from the compartment IP address, as described in [Configuration Data](#).

12.1.3 The provision Service

The `provision` service executes the script provided to provision (deploy) an admin, login, network service in the SRP compartment.

12.1.3.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in [15.1 Creating an SRP Compartment or Adding Data to an SRP](#).

Oracle executable path The root directory for Oracle executables.
Variable Name: `exec_path`.
Default: `/opt/var/hpsrp/srp_name/opt/u01/home/oracle`.

Oracle DB data path The root directory for Oracle data.
Variable Name: `data_path`.
Default: `/var/hpsrp/srp_name/opt/u01/home/oracle`.

12.1.3.2 Completing the Configuration

After applying the `cmpt` service and optionally the `ipfilter` service for the `oracledb` template, you can deploy an Oracle Database Server in the compartment. You may need to make a copy or link from the Oracle product installation directory to the `exec_path` configured for the `cmpt` service. You might also need to set up the Oracle configuration and schema under the `data_path` configured for the `cmpt` service.

Also, `oracledb` template provides provision service optionally. You must write your provision script if you need one for any of your customized operations; such as to change the permissions or copy some files.

You can also copy or create a startup and shutdown script for the Oracle processes, and install or link it to files in a `/var/hpsrp/srp_name/sbin/rcN.d` directory.

12.2 Replacing or Deleting Oracle SRP Data

Use the following command to replace `oracledb` template data from an SRP compartment:
`srp -r[ep]lace srp_name -t oracledb [-s service[,service]...]`

The `srp -replace` command deletes the specified data, then prompts you for replacement data. For example, the following command deletes all the IPFilter data for the `oracledb` template, then prompts you for replacement data:

```
srp -replace mySRP -t oracledb -s ipfilter
```

Use the following command to delete `oracledb` template data from an SRP compartment:
`srp -d[ele]te srp_name -t oracledb [-s service[,service]...]`

CAUTION: If you do not specify the `-template` and/or `-service` arguments, `srp` deletes all templates and/or services for the compartment. For example, the command `srp -delete mySRP` deletes the entire `mySRP` SRP compartment.

For more information, see [15.2 Deleting Configuration Data](#) and [15.3 Replacing Configuration Data](#).

13 Using the sshd Template

This chapter describes how to use the `sshd` template to configure and provision an HP-UX Secure Shell daemon (`sshd`) in an SRP compartment. You can also use the `sshd` template to delete or modify the `sshd` template data for a compartment.

This chapter addresses the following topics:

- 13.1 Adding the `sshd` Template to an SRP Compartment
- 13.2 Replacing or Deleting SSHD SRP Data

13.1 Adding the `sshd` Template to an SRP Compartment

To use the `sshd` template, you must create a SRP compartment first, then add the `sshd` template to the compartment. For example:

```
srp -add mySRP # create a SRP compartment
srp -add mySRP -template sshd
```

The syntax for adding the `sshd` template to an SRP compartment is as follows:

```
srp -a[dd] srp_name -t[emplate] sshd [-s[ervice] service[,service]....]
```

Where:

srp_name Specifies the name of an existing SRP compartment.

service Specifies the name of the service to configure. The following services are valid with the `sshd` template:

- `cmpt`
- `ipfilter`
- `provision`

If you do not specify any services in the command line, `srp` prompts you for the services you want to apply and displays a list of the default services that are valid with the `sshd` template. If you are using the factory-configured default services, the only valid default service is `cmpt,provision`.

The input data for these services and the data configured are described in the sections that follow. If SRP uses input data for multiple services, the `srp` utility prompts you for the data once and reuses the value.

13.1.1 The `cmpt` Service

The `cmpt` service for the `sshd` template configures Security Containment file system rules to allow the compartment to access the specified Secure Shell directories.

13.1.1.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in 15.1 *Creating an SRP Compartment or Adding Data to an SRP*.

`sshd data path` Specifies the compartment-specific target directory for `sshd` configuration and key files.

Variable Name: `data_path`.
Default: `/var/hpsrp/srp_name/opt/ssh`.

`sshd executable path` The location of the executables for the HP-UX Secure Shell product.
Variable Name: `exec_path`.
Default: `/opt/ssh`.

13.1.1.2 Configuration Data

SRP adds entries to the rules file for the SRP compartment to authorize read access to `exec_path` and all access to `data_path`. SRP also adds an `include` statement to add the rules from the `/opt/hpsrp/etc/cmpt/sshd.srp_incl` file. As delivered by HP, this file is empty. You can edit this file to contain compartment rules to be applied when configuring the `cmpt` service with the `sshd` template.

13.1.2 The ipfilter Service

The `ipfilter` service for the `sshd` template adds rules to allow inbound requests from any IP address to the compartment `sshd` daemon to pass. You can also specify additional inbound destination TCP port numbers for IPFilter pass rules.

13.1.2.1 Input Data

SRP prompts for the following data. You can also specify a variable name and value in the command line, as described in *15.1 Creating an SRP Compartment or Adding Data to an SRP*.

`sshd port number` Specifies the TCP port number on which the compartment `sshd` will receive connection requests.
Variable Name: `sshd_port`.
Valid Input: A TCP port number in the range 1- 65535.
Default: 22, the IANA registered port number for SSH login.

`IPFilter port numbers` Specifies the local TCP port numbers for IPFilter rules that allow inbound packets.
Variable Name: `ipf_tcp_ports`.
Valid Input: One or more TCP port numbers each in the range 1- 65535, separated by commas.
Default: 22. This is the IANA registered port number for SSH remote login.

13.1.2.2 Configuration Data

If the compartment address is an IPv4 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf.conf` file. If the compartment address is an IPv6 address, SRP adds IPFilter rules to the `/etc/opt/ipf/ipf6.conf` file.

SRP configures rules that allow inbound packets from any remote IP address to the compartment IP address with the specified destination TCP port numbers.

SRP inserts these rules at the top of the IPFilter rules file and uses the `quick` keyword.

The IPFilter configuration file already contains rules from the `base` template to allow all outbound TCP, UDP, and ICMP packets from the compartment IP address, as described in *Configuration Data*.

13.1.3 The provision Service

The provision service executes the script `/opt/hpsrp/bin/util/secsh_setup` to provision (deploy) an `sshd` service in the SRP compartment.

13.1.3.1 Input Data

SRP prompts for the following data:

<code>sshd data path</code>	Specifies the compartment-specific target directory for <code>sshd</code> configuration and key files. Variable Name: <code>data_path</code> . Default: <code>/var/hpsrp/srp_name/opt/ssh</code> .
<code>sshd executable path</code>	The location of the executables for the HP-UX Secure Shell product. Variable Name: <code>exec_path</code> . Default: <code>/opt/ssh</code> .
<code>Copy SSH config data from</code>	Specifies the directory from which you want to copy SSH configuration data. In most cases, this should be the <code>newconfig</code> directory shipped with the HP-UX Secure Shell product. Variable Name: <code>data_src</code> . Default: <code>/opt/ssh/newconfig</code> .
<code>sshd port number</code>	Specifies the TCP port number on which the compartment <code>sshd</code> will receive connection requests. Variable Name: <code>sshd_port</code> . Valid Input: A TCP port number in the range 1- 65535. Default: 22, the IANA registered port number for SSH login.
<code>sshd Provision script</code>	Specifies the provision script to be used to configure <code>sshd</code> server in the compartment. Variable Name: <code>script_name</code> Default: <code>/opt/hpsrp/bin/util/secsh_setup</code>

13.1.3.2 Configuration Data

By default, the `/opt/hpsrp/bin/util/secsh_setup` script:

- Uses the SSH `ssh-keygen` utility to generate an RSA key pair to use for the `sshd` host key pair. These keys are stored in the compartment-specific `sshd` data path directory (`/var/hpsrp/srp_name/opt/ssh`) with the following names:

`ssh_host_rsa_key` (RSA private key)
`ssh_host_rsa_key.pub` (RSA public key)

- Creates a compartment-specific copy of the `sshd` configuration file by copying the `sshd_config` file from the specified `data_src` directory to the `data_path` directory and modifying it with compartment-specific data, including setting the `HostKey` parameter to `/var/hpsrp/srp_name/opt/ssh/ssh_host_rsa_key`.

- Creates compartment-specific initialization scripts and startup file to start the `sshd` with the compartment-specific `sshd_config` file when the compartment startup script is executed. The setup script:
 - Creates the compartment-specific startup configuration file, `/var/hpsrp/srp_name/etc/rc.config.d/sshd`, which specifies the compartment-specific **sshd** configuration file as a startup argument for `sshd`.
 - Adds the startup and shutdown script `secsh` to the compartment-specific `init.d` directory, `/var/hpsrp/srp_name/sbin/init.d`. This file is linked to the `/var/hpsrp/srp_name/sbin/rc2.d/S393secsh` and `/var/hpsrp/srp_name/sbin/rc1.d/K393sech` files.

13.1.3.3 Completing the Configuration

Tasks you might need to perform to complete the configuration include the following:

- Editing the compartment `sshd_config` file (the default location is `/var/hpsrp/srp_name/opt/ssh/sshd_config`).
- If a client has the `StrictHostKeyChecking` directive set to `yes`, you must add the host public key file (`ssh_host_dsa_key.pub` or `ssh_host_rsa_key.pub`) to the client configuration, as described in the HP-UX Secure Shell documentation.

13.2 Replacing or Deleting SSHD SRP Data

Use the following command to replace `sshd` template data from an SRP compartment:

```
srp -r[ep]lace srp_name -t sshd [-s service[,service]...]
```

The `srp -replace` command deletes the specified data, then prompts you for replacement data.

For example, the following command deletes all the `IPFilter` data for the `sshd` template, then prompts you for replacement data:

```
srp -replace mySRP -t sshd -s ipfilter
```

Use the following command to delete `sshd` template data from an SRP compartment:

```
srp -d[ele]te srp_name -t sshd [-s service[,service]...]
```

CAUTION: If you do not specify the `-template` and/or `-service` arguments, `srp` deletes all templates and/or services for the compartment. For example, the command `srp -delete mySRP` deletes the entire `mySRP` SRP compartment.

For more information, see [15.2 Deleting Configuration Data](#) and [15.3 Replacing Configuration Data](#).

14 Starting and Stopping SRP Compartments

This chapter describes how to start and stop SRP compartments. For complete syntax information, see `srp(1M)`.

This chapter addresses the following topics:

- *14.1 SRP Startup and Shutdown Processing*
- *14.2 Starting an SRP Compartment*
- *14.3 Stopping an SRP Compartment*

14.1 SRP Startup and Shutdown Processing

By default, all SRP compartments are automatically started at system startup time and are automatically stopped at system shutdown time. Each SRP home directory contains an `sbin` subdirectory (`/var/hpsrp/srp_name/sbin`) that contains a subtree similar to the system `/sbin` subtree. Each `sbin` directory contains an `init.d` directory and an `rcN.d` subdirectory tree with the subdirectories `rc0.d`, `rc1.d`, `rc2.d`, `rc3.d`, and `rc4.d`.

The SRP product also includes the `/opt/hpsrp/bin/util/srp_rc` script. This is a sequencer script that is executed at system startup and shutdown time and is similar to the `/sbin/rc` utility, (see `rc(1M)`), with the following differences:

- The `/opt/hpsrp/bin/util/srp_rc` script first restarts itself in the target compartment by executing the following command:

```
privrun -c srp_name /opt/hpsrp/bin/util/srp_rc
```

This causes all the processes and scripts executed by `srp_rc` to run in the target SRP compartment.

- The `srp_rc` script executes the files in the `/var/hpsrp/srp_name/sbin/init.d/rcN.d` directories (instead of the files in the `/sbin/init.d/rcN.d` directories).

The `/opt/hpsrp/bin/util/srp_rc` script executes the scripts in the `rcN.d` directories in alphabetical order. When the system starts up, SRP traverses these directories in ascending order and executes scripts in these directories. When the system shuts down, SRP traverses these directories in descending order and executes scripts in these directories.

NOTE: The numeric portion of the subdirectory name does not correspond to the system run level at which the scripts are executed. The system run level at which the scripts are executed is determined by the run level at which the SRP master script, `/sbin/init.d/srp`, runs (run level 3 when the system starts up and at run level 2 when the system shuts down).

At system startup or shutdown time, the SRP scripts are executed as follows:

- The `/sbin/init.d/srp` file is linked to the following files:

```
/sbin/rc3.d/S999srp  
/sbin/rc2.d/K001srp
```

- The `/sbin/rc3.d/S999srp` file is the last or one of the last startup scripts executed when the transitions from run level 2 to run level 3 (typically at system startup). The `/sbin/rc2.d/K001srp` file is the first or one of the first shutdown scripts executed when the system transitions from run level 3 to run level 2 (typically at system shutdown). The SRP initialization and shutdown scripts are processed as follows:
- The `/sbin/init.d/srp` script reads the `/etc/rc.config.d/srpconf` file to determine which SRP compartments have the autostart feature enabled.
- The `/sbin/init.d/srp` script executes the `/opt/hpsrp/bin/util/srp_rc` script for each compartment that has the autostart feature enabled.
- The `/opt/hpsrp/bin/util/srp_rc` script executes the scripts in the `rcN.d` directories in alphabetical order. When the system starts up, SRP traverses these directories in ascending order and executes scripts in these directories. When the system shuts down, SRP traverses these directories in descending order and executes scripts in these directories.

The files in the `/var/hpsrp/srp_name/sbin/rcN.d` directories are also executed when the `srp -start` or the `srp -stop` command is executed for the compartment.

14.2 Starting an SRP Compartment

Use the following command to start an SRP compartment:

```
srp -sta[rt] srp_name
```

Where:

srp_name Specifies the name of an SRP compartment.

The `srp -start` command starts an SRP compartment by executing the `/opt/hpsrp/bin/srp_rc` script with the compartment name and `start` argument:

- The `srp_rc` script runs a customizable prestart script, `/var/hpsrp/srp_name.setup/setup`. The script is run in the INIT compartment, with the following command line:
`/var/hpsrp/srp_name.setup/setup start srp_name /var/hpsrp/srp_name`
- The `srp_rc` script runs each of the start scripts in the run-level directories of the SRP, `/var/hpsrp/srp_name/sbin/init.d/rcN.d`. The start scripts are files with names that start with an S. Each script is run in the compartment of the SRP that has the `start` parameter.

14.3 Stopping an SRP Compartment

Use the following command to stop an SRP compartment:

```
srp -sto[p] srp_name
```

Where:

srp_name Specifies the name of an SRP compartment.

The `srp -stop` command shuts down an SRP compartment by executing the `/sbin/init.d/srp_rc` script with the compartment name and the `stop` argument:

- The `srp_rc` script executes the shutdown scripts in the `/var/hpsrp/srp_name/rcN.d` directories. The shutdown scripts are files with names that start with a K. The scripts are run in the compartment of the SRP that has the `start` parameter.

- The `srp_rc` script runs a customizable post-stop script, `/var/hpsrp/srp_name.setup/setup`. The script is run in the INIT compartment, with the following command line:
`/var/hpsrp/srp_name.setup/setup start srp_name /var/hpsrp/srp_name`

15 Managing SRP Data

This chapter describes how to add, update, delete, list, and manage SRP data. For complete syntax information, see *srp(1m)*. This chapter addresses the following topics:

- 15.1 Creating an SRP Compartment or Adding Data to an SRP
- 15.2 Deleting Configuration Data
- 15.3 Replacing Configuration Data
- 15.4 Displaying Help Text and Input Parameters
- 15.5 Listing Configuration Information About SRP Compartments
- 15.6 Displaying status of SRP Compartments
- 15.7 Using *srp* in Batch Mode

15.1 Creating an SRP Compartment or Adding Data to an SRP

The *srp -add* command creates an SRP base compartment, or adds service data for a template to a compartment. To add service data for an application template (any template other than the base template), the SRP compartment must already exist. If you do not specify a template, *srp* uses the base template. If you do not specify services, *srp* prompts you with a list of default services valid for the template.

Use the following command to add a template or service to an SRP compartment:

```
srp -a[dd] srp_name [-t template[,template]....] [-s service[,service]....] [i[d] instance] [variable=value]...
```

Where:

<i>srp_name</i>	Specifies the SRP compartment name. If you are using the <i>base</i> template and this compartment does not already exist, <i>srp</i> creates the compartment for you. If you are using an application template, this must be the name of an existing SRP compartment.
<i>template</i>	Specifies the template names. Valid Input: <i>base</i> , <i>apache</i> , <i>tomcat</i> , <i>custom</i> , <i>oracledb</i> , <i>sshd</i> . Default: <i>base</i> .
<i>service</i>	Specifies the names of the services to configure. If you do not specify any services in the command line, <i>srp</i> prompts you for the services you want to apply and displays a list of the default services that are valid with the template being processed. Table 15.1 lists the services valid for each template.

Table 15.1 Valid Services

Template	Valid Services
----------	----------------

base	admin
	cmpt
	init
	ipfilter
	ipsec
	login
	network
	prm

```

apache    cmpt
          ipfilter
          provision

custom    cmpt
          ipfilter
          provision

oracledb  cmpt
          ipfilter

sshd      cmpt
          ipfilter
          provision

```

Default: The default service set is configured via `srp_sys -setup`.

instance Unique string identifier used to identify an instance of a template usage for templates that can be applied multiple times. This is valid for the custom template only and is ignored for all other templates.
Valid Input: A text string with alphanumeric, dash (-), or underscore (_) characters. The maximum length is 20 characters.
Default: None.

variable=value A valid variable for the specified service when used with the specified template and the value for the variable. To list the valid variable names for a service used with a given template, you can use the `srp -h -v template template_name -service service_name` command. For example, the `network` service for the `base` template supports the variable `ip_address`. The user can set the value for this variable in the command line as follows:

```
srp -add mySRP ip_address=192.0.2.1 iface=lan1:1
```

The `srp` utility will skip the prompt for the `ip_address` variable in the interactive dialog and use the specified value.
This feature is often used with the `-batch` option, as described in *Using srp in Batch Mode*.

15.2 Deleting Configuration Data

Use the following command to delete template or service data from an SRP compartment:

```
srp -d[delete] srp_name [-t template[,template]...] [-s service[,service]...] [i[d] instance]
```

Where:

srp_name Specifies the name of an existing SRP compartment.

template Specifies the template names.
Valid Input: `base`, `apache`, `tomcat`, `custom`, `oracledb`, `sshd`.
Default: All templates configured for the SRP compartment.

<i>service</i>	Specifies the names of the services to delete. Default: All services configured for the template.
<i>instance</i>	Unique string identifier used to identify an instance of a template usage for templates that can be applied multiple times. This is valid for the custom template only and is ignored for all other templates. Valid Input: A text string with alphanumeric, dash (-), or underscore (_) characters. The maximum length is 20 characters. Default: None.

CAUTION: If you do not specify the `-template` and/or `-service` arguments, `srp` deletes all templates and/or services for the compartment. For example, the command `srp -delete mySRP` deletes the entire `mySRP` SRP compartment.

15.3 Replacing Configuration Data

Use the following command to replace template or service data from an SRP compartment:

```
srp -r[ep]lace srp_name [-t template[,template]...] [-s
service[,service]...] [i[d] instance]
```

Where:

<i>srp_name</i>	Specifies the name of an existing SRP compartment.
<i>template</i>	Specifies the template names. Valid Input: <code>base</code> , <code>apache</code> , <code>tomcat</code> , <code>custom</code> , <code>oracledb</code> , <code>sshd</code> . Default: All templates configured for the SRP compartment.
<i>service</i>	Specifies the name of the service data to replace. Default: All services configured for the template.
<i>instance</i>	Unique string identifier used to identify an instance of a template usage for templates that can be applied multiple times. This is valid for the custom template only and is ignored for all other templates. Valid Input: A text string with alphanumeric, dash (-), or underscore (_) characters. The maximum length is 20 characters. Default: None.

The `srp -replace` command prompts for new data for the specified services and replaces the entire data set for the service with the replacement data. If you do not supply new values for an option or variable, the `srp` default value will be used (not the currently configured value).

CAUTION: If you do not specify the `-template` and/or `-service` arguments, `srp` deletes all templates and/or services for the compartment. For example, the command `srp -delete mySRP` deletes the entire `mySRP` SRP compartment.

15.4 Displaying Help Text and Input Parameters

Use the following command to display `srp` help text and information about input parameters:

```
srp -h[elp] [-v[erbose]] [-t template[,template]...] [-s
service[,service]...]
```

Where:

<i>verbose</i>	Displays verbose (detailed) help text.
<i>template</i>	Specifies the templates for which you want to display parameters. Valid Input: <code>base</code> , <code>apache</code> , <code>tomcat</code> , <code>custom</code> , <code>oracledb</code> , <code>sshd</code> . Default: <code>base</code> .
<i>service</i>	Specifies the services for which you want to display parameters.

Table 15.1 lists the services valid for each template.

Default: The default services that are valid for the template. The factory configured default services are: `admin`, `cmpt`, `init`, `login`, `network`, and `prm`. You can configure an alternate set of default services via `srp_sys -setup`, as described in *2 Setting Up an SRP*.

15.5 Listing Configuration Information About SRP Compartments

Use the following command to display the configuration summary or details for the SRP compartments configured on the system:

```
srp -l[ist] [srp_name] -v[erbose] [-t template[,template]....] [-s
service[,service]....] [i[d] instance][-x[mloutput]]
```

Where:

<i>srp_name</i>	Specifies the name of an existing SRP compartment. If you do not specify an SRP name, <code>srp</code> displays information for all SRPs configured on the system.
<i>verbose</i>	Verbose mode. Displays detailed configuration data. If not specified, the list operation will display only the names of templates and services configured.
<i>template</i>	Specifies the templates for which you want to list configuration. Valid Input: <code>base</code> , <code>apache</code> , <code>tomcat</code> , <code>custom</code> , <code>oracledb</code> , <code>sshd</code> . Default: All templates currently applied to the SRP.
<i>service</i>	Specifies the services for which you want to display configuration. Default: All services currently applied to the SRP.
<i>instance</i>	Unique string identifier used to identify an instance of a template usage for templates that can be applied multiple times. This is valid for the custom template only and is ignored for all other templates. Default: All instances for the specified service(s) and template(s).
<i>xmloutput</i>	Display output in XML format.

15.6 Displaying status of SRP Compartments

Use the `srp -status` command to display a status summary for SRP compartments:

```
srp -status [[srp_name] [-verbose|-xmloutput[]]]
```

<i>srp_name</i>	Specifies the name of an existing SRP compartment. If you do not specify a compartment name, <code>srp</code> displays information about all compartments configured on the system.
<i>verbose</i>	Verbose mode. Displays detailed status data.
<i>template</i>	Specifies the templates for which you want to display the status data. Valid Input: <code>base</code> , <code>apache</code> , <code>tomcat</code> , <code>custom</code> , <code>oracledb</code> , <code>sshd</code> . Default: All templates currently applied to the SRP.
<i>service</i>	Specifies the services for which you want to display configuration. Default: All services currently applied to the SRP.
<i>xmloutput</i>	Display output in XML format.

15.7 Using `srp` in Batch Mode

The `-batch` or `-b` option runs `srp` in batch mode. Instead of prompting the user for input, `srp` uses the default values for input variables. If there is no input default value for an input variable, you must specify the value in the command line. For example:

```
/opt/hpsrp/bin/srp -add mySRP -batch ip_address=192.0.2.2 iface=lan1:1
```


16 Customizing SRP Data

This chapter describes procedures for customizing SRP data. It addresses the following topics:

- *16.1 Modifying Provision Scripts*
- *16.2 Modifying Compartment Rule Include Files*
- *16.3 Manually Editing SRP Configuration Data*

NOTE: You should run the system administration and performance tools (for example: `glance`, `gpm`, `kprof`, `kgmon`, `ktrace`, and `caliper`) in the `INIT` compartment

16.1 Modifying Provision Scripts

A provision script performs the tasks needed to provision or deploy an application in an SRP compartment. These tasks can include copying data from an application's normal installation directory to the home directory for the SRP compartment. The `srp` utility passes selected `srp` utility arguments and variables to the provision scripts, such as the `srp` operation, the compartment name, compartment IP address, compartment data and execution paths, and other application-specific variables.

You can modify the provision scripts to add tasks needed to deploy an application. The provision scripts provided with SRP are:

- `apache`: `/opt/hpsrp/bin/util/apache_setup`
- `tomcat`: `/opt/hpsrp/bin/util/tomcat_setup`
- `ssh`: `/opt/hpsrp/bin/util/secsh_setup`
- `Custom`: provided as an input variable

16.2 Modifying Compartment Rule Include Files

The `srp` utility uses include files to configure Security Containment compartment rules. There is an include file for each template. If you modify the contents of an include file for a template, all SRP compartments configured with the `cmpt` service for that template will use the modified include file. The include file names have the following format:

```
/opt/hpsrp/etc/cmpt/template_name.srp_incl
```

For example, `/opt/hpsrp/etc/cmpt/apache.srp_incl`.

16.2.1 Securing SRP Compartments with Compartment Rule Include Files

The base template rules file delivered with the product provides a rule set designed to allow maximum application compatibility while providing restricted access to files not needed to be modified or accessed by applications or user sessions. To increase the security of your environment, you can replace this file with a more restrictive rule set tuned to your application requirements and local security policy.

To create an environment with the minimal compartment access rights, you can use a procedure such as the following:

1. Make a copy the default base compartment rules file,
`/opt/hpsrp/etc/cmpt/base.srp_incl`. For example:

```
# cd /opt/hpsrp/etc/
```

```
# cp base.srp_incl myCustom.srp_incl
```

2. Remove the rules in the original (`base.srp_incl`) file. This creates an empty Security Compartment rules file. A compartment that uses only this file for its compartment rule set will have no access any files, system IPC, or network interfaces.

NOTE: Creating an empty Security Compartment rules file for the `base` template files affects all compartments using this file, including those previously created. This practice is recommended in a highly secure environment to ensure that all compartments are specifically configured, and no compartments are continuing to execute with default rules.

3. Determine the minimum set of rules that you need for a compartment and add them to the new file (`myCustom.srp_incl` in this example). For more information on creating a deployment-specific compartment rules set, see *HP-UX System Administrator's Guide: Security Management: HP-UX 11i Version 3*.
4. Use the custom template to associate this new rules file to compartments requiring the specified access. For example:

```
# srp -a mySRP -template custom -id myID
```

When `srp` prompts for `Compartment rule files`, enter the name of the new file (`/opt/hpsrp/etc/myCustom.srp_incl` in this example)

16.3 Manually Editing SRP Configuration Data

SRP marks the data it adds to subsystem configuration files and databases with *tags*, or text-string identifiers. SRP uses these tags when selecting data for SRP replace and delete operations. You can use these tags to identify and manually edit SRP configuration data and still use SRP replace and delete operations to manage this data if you retain the tag information.

A quick way to identify configuration data managed by SRP is by using the following command:
`srp -l srp_name -v`

16.3.1 Tag Formats

The general format for most tags that indicate the start of SRP data is as follows:

```
@tag-start 'compartment="srp_name" template="template_name"  
service="service_name" id="version";
```

Where:

srp_name Specifies the SRP compartment name.

template Specifies the name of the template used to configure the data

service Specifies the service name.

version A string used to identify an instance of a service applied to a compartment. This field is meaningful only with the `custom` template, which allows you to create multiple instances of service data for the same template and compartment. For all other templates, the string is always `1`.

The specific tag format for each subsystem is described in the sections that follow.

16.3.1.4 Security Containment Compartment Tag Format

Data is stored in the `/etc/cmpt/srp_name.rules` file by default. When SRP adds data, it indicates the start of the data with the following tag:

```
//@tag-start 'compartment="srp_name" template="template_name"  
service="cmpt" id="instance";
```

SRP indicates the end of the data with the following tag:

```
//@tag-end;
```

16.3.1.5 RBAC and Compartment Login Tag Format

Data is stored in files under the `/etc/rbac` directory. HP recommends that you use RBAC commands (`roleadm`, `authadm`, `cmdprivadm`) to modify RBAC data.

SRP identifies RBAC data for the `admin` service by using the following values:

- Role name: `SRPadmin-srp_name` for the compartment
- Authorization: `hpux.SRPadmin.srp_name` for the compartment
- Command privilege: `hpux.SRPadmin.srp_name` for the compartment

SRP identifies RBAC data for the `login` service by using the following values:

- Role name: `SRPlogin-srp_name` for the compartment
- Authorization: `hpux.security.compartment.login` for the compartment

16.3.1.6 Network Configuration Tag Format

For IPv4 interfaces, SRP adds the following entry to the `/etc/rc.config.d/netconf` file:

```
IPV4_CMGR_TAG[index]='compartment="srp_name" template="base"  
service="network" id="instance"'
```

Where `index` is the first available index number for interface parameters in the `netconf` file. SRP uses the index number to identify the following interface parameters:

```
INTERFACE_NAME  
IP_ADDRESS  
SUBNET_MASK  
INTERFACE_STATE  
BROADCAST_ADDRESS  
DHCP_ENABLE  
INTERFACE_MODULES
```

SRP uses the address configured for the `IP_ADDRESS` entry to identify the `ROUTE_SOURCE` entry for the compartment, and uses that index number to identify the corresponding route entries.

The data is similar for IPv6 interfaces, with the following differences:

- The data is stored in the `/etc/rc.config.d/netconf-ipv6` file.
- The names of the interface parameters are correct for IPv6 interfaces, such as `IPV6_INTERFACE`, `IPV6_ADDRESS`, `IPV6_INTERFACE_STATE`.
- SRP does not add or manage IPv6 route entries.

16.3.1.7 PRM Tag Format

Data is stored in the `/etc/prmconf` file by default. When SRP adds data, it indicates the start of the data with the following tag:

```
#@tag-start compartment="srp_name" template="base" service="prm"  
id="instance";
```

SRP indicates the end of the data with the following tag:

```
#@tag-end;
```

16.3.1.8 IPFilter Tag Format

Data is stored in the `/etc/opt/ipf/ipf.conf` file for IPv4 addresses and in `/etc/opt/ipf/ipf6.conf` for IPv6 addresses. When SRP adds data, it indicates the start of the data with the following tag:

```
#@tag-start compartment="srp_name" template="template_name"  
service="ipfilter" id="instance";
```

SRP indicates the end of the data with the following tag:

```
#@tag-end;
```

16.3.1.9 IPSec Tag Format

IPSec stores configuration data in the IPSec database, `/var/adm/ipsec/config.db`. To modify the contents of the IPSec database, you must use the `ipsec_config` utility.

The configuration objects (IPSec host policy, IKE policy, and authentication record) each has a name with following format:

```
SRP-srp_name-base-ipsec
```

17 Exporting and Importing SRPs

You can export and import an SRP across systems by using the `srp -export` and `srp -import` commands. These commands allow you to accomplish the following:

- Create a clone of an SRP on a secondary system for high availability or load balancing purposes.
- Migrate an SRP across systems: export and import an SRP, then delete the original SRP.
- Create a copy of an SRP for archival purposes. Similarly, an SRP can be taken offline by exporting and deleting the original SRP.

This chapter discusses the following:

- *17.1 Using the `srp -export` Command*
- *17.2 Using the `srp -import` Command*
- *17.3 Best practices for Exporting and Importing an SRP*

17.1 Using the `srp -export` Command

The `srp -export` command exports the configuration data and optionally, specified directories, for the specified SRP compartment. To export the SRP, the `srp -export` command copies the SRP's data and stores it in the specified exchange file. The exchange file is used by the `srp -import` command. You can only export an SRP that is in a `stopped` state.

The `srp -export` command has the following syntax:

```
srp -export [<srp_name>] [-xfile <exchange_file>]
           [-b[atc]] [variable=<value>...]
```

The options are as follows:

<i>srp_name</i>	Specifies the name of an existing SRP compartment.
<i>exchange_file</i>	Specifies the exchange file name. The file is created if it does not already exist. The default is <code>srp.exchange</code> .
<i>batch</i>	Run <code>srp-export</code> in batch mode. The command will not prompt for arguments or template variable values.
<i>variable=value</i>	The most commonly used variable for the <code>export</code> operation is <code>export_copy_dirs</code> . It is a comma separated list of directory names to be copied to the exchange file. If <code>export_copy_dirs</code> is not specified, no directories are exported, and only the system configuration that defines the SRP is included in the exchange file. This is useful for archiving the SRP definition, and for cloning an SRP across systems where the entire SRP home directory will be mounted via shared storage.

NOTE:

- To export the directories mounted via the `fstab` (`/var/hpsrp/<srp_name>/etc/fstab`) file of the SRP, you must mount the directories prior to executing the `srp -export` command, as follows:
 1. Mount the `fstab` filesystems of the SRP:

```
#export compartment=<srp_name>
```

```
#/var/hpsrp/<srp_name>/sbin/init.d/srp_mount start
```

```
2. Export the SRP:  
#srp -export<srp_name>
```

```
3. Unmount the fstab filesystems of the SRP:  
#export compartment=<srp_name>  
#/var/hpsrp/<srp_name>/sbin/init.d/srp_mount stop
```

- User and group definitions are system properties that are not SRP specific and therefore are not exported with the SRP. The compartment login permissions for users and groups allowed to login to the SRP will be exported, but you must ensure that the required users and groups are configured on the target system, or are accessible via a common name service, such as LDAP-UX.

Example: Export the SRP `mySRP` and all directories under the `mySRP` home directory:
`#srp -export mySRP export_copy_dirs=/var/hpsrp/mySRP`

17.2 Using the `srp -import` Command

The `srp -import` command imports the SRP contained in the specified exchange file. The exchange file contains the previously exported SRP's configuration, and possibly specified directories. The `srp -import` command validates the ability for the target system to accept the exchange file and configures the new SRP. You can only import an SRP that does not exist on the target system.

NOTE:

- An imported SRP will not be automatically started at system boot time. Refer to [8.1.5.1 Input Data](#) for information about how to enable the `autostart` feature.
- Cloning an SRP on the same system is not supported.

You can save the exchange file for archival purposes, or you can import it to another system.

The `srp -import` command has the following syntax:

```
srp -import [-b[atch]] [-preview] [-xfile <exchange_file>]  
           [variable=<value>...]
```

The options are as follows:

<i>batch</i>	Run <code>srp-import</code> in batch mode. The command will not prompt for arguments or template variable values.
<i>preview</i>	Validates if the target system will accept an exchange file for import. If you specify the <code>preview</code> option, only the validation is performed, no import action occurs.
<i>exchange_file</i>	Specifies the existing exchange file name. The <code>srp -export</code> command creates the exchange file. The default is <code>srp.exchange</code> .
<i>variable=value</i>	The most commonly used variables for the <code>import</code> operation are as follows: <ul style="list-style-type: none">• <code>iface</code> The network interface name on which to assign the imported IP address of the SRP.• <code>allow_sw_mismatch</code> Set to <code>yes</code> to force an import operation to succeed if the required products for SRP are not present or if there is

a software version mismatch from the source to the target system.
The default is no.

You can use the following notation to assign a value to a variable:
name=value, name='value', or name="value"

17.3 Best practices for Exporting and Importing an SRP

To simplify the export and import of an SRP across systems, HP recommends to you keep the properties of the SRP to be atomic and that you do not share files and data with other SRP compartments unnecessarily. The following are best practices for using the `srp -export` and `srp -import` commands:

- **Maintain consistent OS versions and patch levels across systems**
While SRP itself has minimal requirements for OS levels and patches to be synchronized across systems, applications utilized within the SRP compartments may have more specific requirements. By maintaining consistency across systems, you will not have to manually track application dependencies when determining a target system for the SRP to import.
- **Consider using shared storage or file systems when creating an SRP that will be cloned**
By using shared storage for the SRP home directory and any file systems mounted within the SRP, you will not need to export and import file sets, and the data between SRPs will remain consistent.
- **Keep files and directories used by the SRP within the SRP home directory**
Locating files within the SRP home directory (`/var/hpsrp/<srpname>`) will simplify exporting or mounting SRP file sets.
- **Avoid cross-compartment security relationships**
If you customize the SRP compartment configuration to include access rules specifying another compartment, the other compartment must exist on the system you import the SRP into.
- **Synchronize or centralize user and group management**
By ensuring that all systems have the same users and groups, SRP login user and groups and file and directory ownership will not have to be modified after importing an SRP on a target system. Consider defining a single group for all users for a given SRP.
- **Use the `-preview` option to identify a suitable target system**
You can use the `srp -import` command with the `-preview` option to validate if a target system will accept an exchange file for import.
- **Adjust device configuration after import**
Configuration that specifies physical paths such as network interface devices and file system mount points require manual configuration changes after import. HP recommends that you adjust these device configurations after completing the import operation.

18 Using Serviceguard with SRP

Serviceguard allows you to create high availability clusters of HP 9000 or HP Integrity Servers. A high availability computer system allows application services to continue in spite of a hardware or software failure. Highly available systems protect users from software failures as well as from failure of a system processing unit (SPU), disk, or local area network (LAN) component. In the event that one component fails, the redundant component takes over. Serviceguard and other high availability subsystems coordinate the transfer between components.

You can use Serviceguard to provide high availability to your SRP deployment. Serviceguard can manage a Serviceguard package executing within an SRP, or manage the SRP itself as a Serviceguard package. You can also use the export and import features of SRP to create a secondary (failover) environment. For more information on copying an SRP environment, see *17 Exporting and Importing SRPs*.

This chapter discusses the following:

- 18.1 Choosing a Model
- 18.2 Creating an SRP to Use with Serviceguard
- 18.3 Adapting Serviceguard Scripts for the Classic Model
- 18.4 Creating Serviceguard Scripts for the SRP Package Model

18.1 Choosing a Model

HP offers two different models when using Serviceguard with SRP: the **classic model** and the **SRP package model**.

In the classic model, the SRP is in the `started` state and Serviceguard has not yet started managing the application inside the SRP. This model is most compatible with the existing Serviceguard packages.

In the SRP package model, the SRP itself is the Serviceguard package. This model takes advantage of the capabilities of SRP by simplifying the Serviceguard scripts and allowing application startup and shutdown to be managed by SRP. Serviceguard starts and stops the SRP; and the SRP initialization and shutdown process starts and stops the applications within the SRP. This model simplifies the Serviceguard packages and requires less maintenance and administration of startup and shutdown activities. With this model, you can choose either Serviceguard or SRP to control the file system mounting and the network interface management.

18.2 Creating an SRP to Use with Serviceguard

If you want to create an SRP that will use Serviceguard, you must first determine how SRP and Serviceguard will interact together. The following steps will give you the information that you need to configure an SRP appropriately:

1. **Select the model**

If you have existing Serviceguard control scripts that you want to leverage, HP recommends that you use the classic model. For a new deployment of a Serviceguard package, HP recommends that you use the SRP package model as it is easier to create.

2. Select which application will have the control

Determine whether SRP or Serviceguard will control the mounting of file systems and management of the network interface, as follows:

- If you selected the classic model in step 1, HP recommends using Serviceguard to control the mounting of file systems and management of the network interface.
- If you selected the SRP package model in Step 1, HP recommends using SRP to control the file system mounting and management of the network interface. If you want to use the Serviceguard network failover capability, then Serviceguard must control the management of the network interface.

3. Create the SRP

Use the `srp -add` command to create the SRP. You will be asked specific questions regarding your choices that you made in step 1 and step 2:

```
Assign IP address at SRP start time? [yes]
Select "yes" to instruct SRP to control network interface management.
Enter "no" to defer control of network management to Serviceguard.
(Note: you may also use assign_ip=yes|no on the srp command line)
```

```
Autostart SRP at system boot? [yes]
Select "yes" for the classic model.
Enter "no" for the SRP package model.
(Note: you may also use autostart=yes|no on the srp command line)
```

18.3 Adapting Serviceguard Scripts for the Classic Model

In an SRP environment, Serviceguard runs in the `INIT` compartment while the managed applications run inside the SRP compartments. If you want Serviceguard to manage or monitor the applications executing within the managed SRP, use the `srp_su` command to let Serviceguard access the managed SRP. You must append the `srp_su` command to the command that requires execution within the SRP compartment.

In the following example, the representative Serviceguard package was modified to control `mySRP`, a package executing in the SRP. The `service_cmd` value is the only value that changed in the script:

Before:

```
service_name      service_ping
service_cmd       "/usr/sbin/ping node_a"
service_restart   unlimited
service_fail_fast_enabled no
service_halt_timeout 300
```

After:

```
service_name      service_ping
service_cmd       "/opt/hpsrp/bin/util/srp_su mySRP root -c '/usr/sbin/ping node_a'"
service_restart   unlimited
service_fail_fast_enabled no
service_halt_timeout 300
```

Either SRP or Serviceguard can manage the network interfaces. If Serviceguard is managing the network interfaces, HP recommends that you configure the default route for any SRP IP address. In the

following example, the representative Serviceguard package was modified to add a default route, `external_script`:

Before:

```
# SG ip address
ip_subnet          192.10.25.0
ip_address         192.10.25.12
```

After:

```
# SG ip address
ip_subnet          192.10.25.0
ip_address         192.10.25.12
# srp_route_script configures the required source based routing entries
for
# the SG managed IP addresses
external_script    /etc/cmcluster/pkg1/srp_route_script
```

See

Appendix B SRP Serviceguard Default Route Script for an example of the `srp_route_script` script.

18.4 Creating Serviceguard Scripts for the SRP Package Model

The SRP package model manages the SRP itself as a package. At start time, the package starts the SRP and relies on the SRP initialization process to mount any application specific file systems and start the applications. Similarly, at stop time, the package stops the SRP relying on SRP to perform any application shutdown and file system unmounting.

Monitoring of the package is accomplished using the same method as described for the classic model. The SRP package model allows you to share the entire SRP home directory between primary and secondary SRPs. If you choose to share a common home directory, then the Serviceguard package should mount and unmount the home directory before starting and stopping the SRP. See *7.2 Managing SRP Startup and Shutdown Actions*.

Either SRP or Serviceguard can manage the network interfaces. If Serviceguard is managing the network interfaces, HP recommends that you instruct Serviceguard to configure the default route for any SRP IP address. See *Appendix B SRP Serviceguard Default Route Script* for an example.

You can find a reference implementation of the SRP package model at:
`/opt/hpsrp/example/serviceguard/srp_as_sg_package/`

To create a functional example of a working SRP based on a Serviceguard package, follow the instructions as specified in the following file:

`/opt/hpsrp/example/serviceguard/srp_as_sg_package/README`

19 Verifying and Troubleshooting SRP

This chapter contains procedures for verifying and troubleshooting SRP. This chapter addresses the following topics:

- 19.1 Verification Procedures
- 19.2 Troubleshooting Procedures
- 19.3 Reporting Problems

NOTE: You can run system administration and performance tools (such as `glance`, `gpm`, `kprof`, `kgmon`, `ktrace`, and `caliper`) in the `INIT` compartment.

19.1 Verification Procedures

This section includes the following procedures to verify the subsystem data configured by SRP:

- 19.1.1 Verifying SRP Subsystems
- 19.1.2 Verifying Security Containment Compartment Data
- 19.1.3 Verifying RBAC Data
- 19.1.4 Verifying PRM Data
- 19.1.5 Verifying Network Data
- 19.1.6 Verifying IPFilter Data
- 19.1.7 Verifying IPSec Data

19.1.1 Verifying SRP Subsystems

You can use the `srp_setup` utility to quickly verify the status of the subsystems with data managed by SRP.

19.1.2 Verifying Security Containment Compartment Data

Use the following procedures to verify Security Containment Compartment configuration data:

- Verify that the compartment rules are loaded into the kernel.

Enter the following command:
`getrules -m srp_name`

- Manually test the file access rules.

Login to the SRP compartment and attempt file access operations that should succeed or fail, such as `cd` and `touch` commands for files not available from the SRP. From the `INIT` compartment, you can create a temporary file in a directory for which the SRP compartment does not have `u`link (delete) access. Login to the SRP compartment and attempt to delete the file.

- Verify that the processes configured for the SRP compartment are running in the compartment.

Use the `ps -ef` command to find the PID for applications in your SRP compartment. For example:

```
# ps -ef | grep sshd
root  968      1  0  Oct 14   ?        0:00 /usr/sbin/sshd
```

Use the `getprocxsec -c pid` command to verify the compartment in which the process is running. For example:

```
# getprocxsec -c 968
cmpt= SRP2
```

- If an application is failing in a compartment and you want to determine if it is failing because of Security Containment rules, you can use the HP-UX audit utility to configure and view audit to see if operations are failing because of permission problems.

One method to reduce the number of unrelated audit entries is to disable auditing for all users, then enable auditing for the user ID used to execute the application. Next, configure auditing for failed attempts for common file and IPC operations. For example:

```
audevent -F -e open -e create -e delete -e ipccreat -e ipcopen \
-e ipcclose -s kill
```

19.1.3 Verifying RBAC Data

Use the following procedures to verify RBAC configuration data:

- Use the `authadm` command to verify the authorization information configured for the compartment:
`authadmlist list object=srp_name`

For the `admin` service, you should see the following entry:

```
SRPadmin-srp_name: (hpux.SRPadmin.srp_name, srp_name)
```

For the `login` service, you should see the following entry:

```
SRPlogin-srp_name: (hpux.security.compartment.login, srp_name)
```

Alternatively, you can enter the following commands to view the authorization information:

```
authadm list operation=hpux.SRPadmin.srp_name
authadm list operation=hpux.security.compartment.login \
object=srp_name
```

- To verify the users and user groups assigned to the roles used by the compartment, enter the following commands:
`roleadm list role=SRPadmin-srp_name`
`roleadm list role=SRPlogin-srp_name`
- To verify command privileges, view the `/etc/rbac/cmd_priv` file. If you configured the `init` service for a compartment, you will see an entry authorizing execution of the `srp_rc` script for an authorization granted to the compartment administrator as follows:

```
/opt/hpsrp/bin/util/srp_rc:dflt:(hpux.SRPadmin.srp_name, *):0/0//:srp_name:dflt:dflt
```

- You can also use the `rbacdbchk` utility to verify the contents of the RBAC database.

19.1.4 Verifying PRM Data

Use the `prmlist` and `prmmmonitor` commands to verify that the PRM configuration is loaded for the group used by the SRP compartment (the default PRM group name is the SRP compartment name).

For example, the `prmlist -g -s` command displays configuration information for PRM groups (`-g`) and the PRM group for each Security Containment compartment (`-s`):

```
# prmlist -g -s
```

```
PRM configured from file: /etc/prmconf  
File last modified: Tue Oct 14 12:57:58 2008
```

PRM Group	PRMID	Entitlement	CPU Max	CPU Attr
EntDir	2	29.17%	80%	
MktDB	65536	12.50%		
MktWeb	3	21.88%	45%	
OTHERS	1	21.88%		
SRP2	4	14.58%	25%	

```
Compartment Default PRM Group
```

EntDir	EntDir
MktDB	MktDB
MktWeb	MktWeb
SRP2	SRP2

The `prmmmonitor` utility displays statistics for each PRM group.

```
# prmmmonitor
```

```
PRM configured from file: /etc/prmconf  
File last modified: Tue Oct 14 12:57:58 2008
```

```
HP-UX habs B.11.31 U ia64 10/14/08
```

```
Tue Oct 14 13:03:11 2008 Sample: 1 second  
CPU scheduler state: Enabled
```

PRM Group	PRMID	CPU Entitle	CPU Max	CPU Used	LCPUs State
OTHERS	1	21.88%		3.06%	
EntDir	2	29.17%	80%	24.10%	
MktWeb	3	21.88%	45%	12.36%	
SRP2	4	14.58%	25%	22.88%	
MktDB	65536	12.50%		12.46%	

```
PRM application manager state: Enabled (polling interval: 30 seconds)
```

19.1.5 Verifying Network Data

Use the `netstat -in` and `netstat -rn` commands to verify the compartment interface and route entries.

The output for the `netstat -in` command lists the IP interfaces configured on the system. An asterisk next to the interface name indicates that the interface is configured, but its state is down. In the following example, the state for `lan1`, `lan1:1` and `lo0` is up, but the state for `lan1:1` is down.

```
# netstat -in
```

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs
Coll							
lan1	1500	10.0.0.0	10.0.0.1	460732	0	279522	0
lan1:1	1500	192.0.2.0	19.0.2.1	32890	0	51537	0
lan1:2*	1500	192.0.2.0	19.0.2.2	0	0	0	0
lo0	32808	127.0.0.0	127.0.0.1	890170	0	890178	0

If an SRP compartment is up and has a dedicated IP interface, the `netstat -rn` command shows a default route entry with the compartment IP address (192.0.2.1) as the gateway. For example:

```
# netstat -rn
```

```

Routing tables
Destination          Gateway              Flags Refs Interface  Pmtu
:
:
default              192.0.2.1          U      0    lan1:1    1500

```

19.1.6 Verifying IPFilter Data

Use the following `ipfstat` command to view the active (loaded) inbound and outbound IPFilter rules:

```
ipfstat -io
```

For example:

```
# ipfstat -io
```

```

pass out quick proto tcp from 192.0.2.1/32 to any keep state
pass out quick proto udp from 192.0.2.1/32 to any keep state
pass out quick proto icmp from 192.0.2.1/32 to any keep state
pass in quick proto icmp from any to 192.0.2.1/32
block in quick from any to 192.0.2.1/32

```

19.1.7 Verifying IPsec Data

Enter the following IPsec commands to verify IPsec data:

- Use the following `ipsec_report` command to view the host rules:
`ipsec_report -host`

The output should include a host policy with the name `SRP-srp_name-base-1`

For example:

```

----- Configured Host Policy Rule -----
Rule Name: SRP-web2-base-1      ID: 7      Priority: 30
Src IP Addr: 192.0.2.1  Prefix: 32  Port number: 0
Dst IP Addr: 10.2.2.2   Prefix: 32  Port number: 0
Network Protocol: All      Action: Dynamic key SA
Number of SA(s) Needed: 1 Pair(s)
Proposal 1: Transform: ESP-AES128-HMAC-SHA1
                Lifetime Seconds: 28800
                Lifetime Kbytes: 0

```

- Use the following `ipsec_report` command to view the IKE rules:
`ipsec_report -ike`

The output should include an IKE policy with the name `SRP-srp_name-base-1`. For example:

```

----- IKE Rule -----
Rule Name: SRP-web2-base-1      Priority: 30      Cookie: 6
Remote IP Address: 10.2.2.2   Prefix: 32
Group Type: 2      Authentication Method: Pre-shared Keys
Authentication Algorithm: HMAC-MD5      Encryption Algorithm: 3DES-
CBC
Number of Quick Modes: 100      Lifetime (seconds): 28800
Action: Secure

```

- Use the following `ipsec_config` command to view the authentication records:
`ipsec_config show auth`

The output should include an IKE policy with the name `SRP-srp_name-base-1`. For example:

```

auth SRP-web2-base-1
-remote 10.2.2.2/32
-preshared myPresharedKey
-exchange MM

```

- You can also use the `ipsec_policy` utility to verify the IPsec host rule selected for a packet from the peer address. In the following example, the SRP compartment address is 192.0.2.1 and the peer address is 10.2.2.2. The `ipsec_policy` command queries IPsec to determine which IPsec and IKE policies are selected for an outbound packet (`-dir out`) with source IP address (`-sa`) 192.0.2.1 and destination IP address (`-da`) 10.2.2.2.

```
# ipsec_policy -sa 192.0.2.1 -da 10.2.2.2 -dir out
```

```

----- Active Host Policy Rule -----
Rule Name: SRP-web2-base-1      ID: 8      Cookie: 3      Priority: 30
Src IP Addr: 192.0.2.1  Prefix: 32  Port number: 0
Dst IP Addr: 10.2.2.2   Prefix: 32  Port number: 0
Network Protocol: All      Direction: outbound
Action: Dynamic key SA      State: SPI(s) Not Established
Number of SA(s) Needed: 1 Pair(s)
Number of SA(s) Created: 0 Pair(s)
Kernel Requests Queued: 0
Proposal 1: Transform: ESP-AES128-HMAC-SHA1
              Lifetime Seconds: 28800
              Lifetime Kbytes: 0

```

```

----- IKE Rule -----
Rule Name: SRP-web2-base-1      Priority: 20      Cookie: 4
Remote IP Address: 10.2.2.2   Prefix: 32
Group Type: 2      Authentication Method: Pre-shared Keys
Authentication Algorithm: HMAC-MD5      Encryption Algorithm: 3DES-CBC
Number of Quick Modes: 100      Lifetime (seconds): 28800
Action: Secure

```

19.2 Troubleshooting Procedures

This section includes the following troubleshooting procedures:

- 19.2.1 Using the Security Containment Compartment Discover Feature
- 19.2.2 Removing or Disabling IPFilter
- 19.2.3 Removing or Disabling IPsec

19.2.1 Using the Security Containment Compartment Discover Feature

In a secure environment, you can use the Security Containment discover feature to remove compartment restrictions and view the rules that are needed to allow access. (If you are not in a secure environment, you can use IPFilter to allow access from only trusted systems before removing compartment restrictions.)

You can use a procedure similar to the following to use the discover feature:

1. Stop the SRP compartment:

```
srp -stop srp_name
```
2. Edit the compartment rules file (`/etc/cmpt/srp_name`), and tag the compartment definition at the beginning of the file with the `discover` keyword. This opens the compartment for all access. For example:

```
discover compartment mySRP {
:
```

:

3. Start the SRP compartment:
`srp -start srp_name`
4. Attempt to access the compartment applications. After you successfully access the applications, enter the following command to generate a machine readable version of the rules used to access the compartment:
`getrules -m srp_name`
5. Compare the output from the `getrules` command with the compartment rules file and make the necessary changes.
6. Stop the SRP compartment, remove the `discover` keyword from the compartment rules file, and then restart the compartment.

19.2.2 Removing or Disabling IPFilter

If you are using IPFilter with SRP, you can see if IPFilter rules are blocking access to the compartment applications. One way to do this is by removing the `ipfilter` service from the compartment by entering the following command:

```
srp -d srp_name [-t template] -s ipfilter
```

If you do not specify the **-t** argument, **srp** removes the IPFilter configuration for the base template. To add the `ipfilter` service back to the compartment after you have completed testing, enter the following command:

```
srp -d srp_name [-t template] -s ipfilter
```

Another method to test if IPFilter rules are blocking access to the compartment applications is by disabling the IPFilter module. Enter the following command:

```
/opt/ipf/bin/ipfilter -d
```

To enable IPFilter after you have completed testing, enter the following command:

```
/opt/ipf/bin/ipfilter -e
```

19.2.3 Removing or Disabling IPsec

If you are using IPsec with SRP, you can see if IPsec policies are blocking access to the compartment applications. One method to determine if IPsec policies are blocking packets is by removing the `ipsec` service from the compartment by entering the following command:

```
srp -d srp_name -s ipsec
```

To add the `ipsec` service back to the compartment after you have completed testing, enter the following command:

```
srp -d srp_name -s ipsec
```

Another method to test if IPsec policies are blocking access to the compartment applications is by stopping the IPsec product. Enter the following command:

```
/usr/sbin/ipsec_admin -stop
```

To restart IPsec after you have completed testing, enter the following command:

```
/usr/sbin/ipsec_admin -start
```


19.3 Reporting Problems

If you are unable to solve a problem with SRP, complete the following steps:

1. Read any published release notes for SRP to see if the problem is known. If it is a known issue, use the prescribed solution.
2. Determine whether the product is still under warranty or whether your company purchased support services for the product. Your operations manager can supply you with the necessary information.
3. Access <http://www.itrc.hp.com/> and search the technical knowledge databases to determine if the problem you are experiencing has already been reported. The type of documentation and resources you have access to depend on your level of entitlement.

NOTE: The ITRC resource forums at <http://www.itrc.hp.com/> offer peer-to-peer support to solve problems and are free to users after registration.

If this is a new problem or if you need additional help, log your problem with the HP Response Center, either on line through the support case manager at <http://www.itrc.hp.com/>, or by calling HP Support. If your warranty has expired or if you do not have a valid support contract for your product, you can still obtain support services for a fee, based on the amount of time and material required to solve your problem.

4. If you are requested to supply any information pertaining to the problem, gather the necessary information and submit it.

Include the following information:

- Output from the following command:

```
srp -l srp_name -v
```
- The contents of the compartment initialization log file,

```
/var/hpsrp/srp_name/etc/rc.log.
```
- A description of the applications hosted in the compartment, including version information.

Appendix A Configuration Example

This appendix includes a sample SRP compartment configuration.

A.1 Sample Base Configuration

This example shows the system configuration created for a sample compartment.

```
# /opt/hpsrp/bin/srp -list mySRP -verbose

Compartment: mySRP  Template: base Service: cmpt
-----

Compartment Configuration (/etc/cmpt/mySRP.rules):
@tag-start compartment="mySRP" template="base" service="cmpt" id="1" ;
#include "/opt/hpsrp/etc/cmpt/base.srp_incl"

// lock out access to the other compartment's root directory
perm nsearch          /var/hpsrp

// open access to compartment root
perm all              /var/hpsrp/mySRP

// to DNS
grant bidir          udp peer port 53 init

Compartment: mySRP  Template: base Service: admin
-----

RBAC Admin Service Configuration:

Role(s):
    SRPadmin-mySRP

Authorization(s):
SRPadmin-mySRP: (hpux.SRPadmin.mySRP, mySRP)

Command privilege(s):
/opt/hpsrp/bin/util/srp_rc:dflt:(hpux.SRPadmin.mySRP,*):0/0//:mySRP:dflt:
dflt:

Compartment: mySRP  Template: base Service: login
-----

RBAC Login Service Configuration:

Role(s):
&adm:SRPlogin-mySRP

Authorization(s):
SRPlogin-mySRP: (hpux.security.compartment.login, mySRP)

Compartment: mySRP  Template: base Service: init
-----
```

SRP init service:

```
//etc/rc.config.d/srpconf: SRP_NAME[1]="mySRP"  
//etc/rc.config.d/srpconf: START_SRP[1]=1
```

Compartment: mySRP Template: base Service: prm

```
PRM Configuration (/etc/prmconf):  
@tag-start compartment="mySRP" template="base" service="prm" id="1" ;  
mySRP:3:10::  
#!PRM_MEM:mySRP:10:::  
#!SCOMP:mySRP:mySRP
```

Compartment: mySRP Template: base Service: network

```
Compartment Configuration (/etc/cmpt/mySRP.rules):  
@tag-start compartment="mySRP" template="base" service="network" id="1" ;  
// owns the IP address  
interface 192.0.2.1
```

Compartment: mySRP Template: base Service: network

```
IP Address Configuration (/etc/rc.config.d/netconf):  
//etc/rc.config.d/netconf: INTERFACE_NAME[2]="lan1:1"  
//etc/rc.config.d/netconf: IP_ADDRESS[2]="192.0.2.1"  
//etc/rc.config.d/netconf: SUBNET_MASK[2]=" "  
//etc/rc.config.d/netconf: INTERFACE_STATE[2]="down"  
//etc/rc.config.d/netconf: BROADCAST_ADDRESS[2]=" "  
//etc/rc.config.d/netconf: DHCP_ENABLE[2]=0  
//etc/rc.config.d/netconf: INTERFACE_MODULES[2]=" "  
//etc/rc.config.d/netconf: ROUTE_DESTINATION[1]=default  
//etc/rc.config.d/netconf: ROUTE_MASK[1]=" "  
//etc/rc.config.d/netconf: ROUTE_GATEWAY[1]="192.0.2.1"  
//etc/rc.config.d/netconf: ROUTE_COUNT[1]=0  
//etc/rc.config.d/netconf: ROUTE_ARGS[1]=" "  
//etc/rc.config.d/netconf: ROUTE_SOURCE[1]="192.0.2.1"
```

The base.srp_incl File

The compartment rules file (/etc/cmpt/mySRP.rules) for this example, like all SRP compartment rule files, includes a reference to the /opt/hpsrp/etc/cmpt/base.srp_incl file. The contents of this file are as follows:

```
/*  
*****  
* Copyright (c) 2008 Hewlett-Packard Development Company L.P.  
*  
* SRP base shared rules include file:  
*  
* This file contains compartments(4) rules that are shared by all  
* SRP compartments that have applied the "base" template.
```

```

*****
*/

/*
*****
* privileges
*****
*/
disallowed privileges none

/*
*****
* ipc/fifo/uxsock to init compartment
*****
*/
access ipc, fifo, uxsock  init

/*
*****
* by default gives users read permission unless otherwise granted
*****
*/
perm read          /

/*
*****
* full access directories for application compartments
*****
*/
perm all           /dev
perm all           /etc
perm all           /home
perm all           /net
perm all           /tmp
perm all           /var

/*
*****
* read-only to system binary libraries, and kernel
*****
*/
perm read          /usr
perm read          /sbin
perm read          /opt
perm read          /stand

/*
*****
* narrow down on /dev:
*****
*/
perm none          /dev/kmem      // kernel memory

perm read          /dev/dsk      // disks
perm read          /dev/rdisk
perm read          /dev/disk
perm read          /dev/rdisk

perm read          /dev/root
perm read          /dev/klog
perm read          /dev/kevm

```

```

perm read                /dev/kepd

/*
*****
* narrow down on /var:
*****
*/
perm none                /var/hpsrp          // SRP compartment root
perm read                /var/opt/hpcmgr
perm read                /var/opt/hpsrp

/*
*****
* narrow down on /etc:
*****
*/
perm read                /etc/opt/hpsrp    // managed by srp
perm read                /etc/opt/hpcmgr
perm read                /etc/opt/ipf
perm read                /etc/cmpt
perm read                /etc/rbac
perm read                /etc/rc.config.d/netconf
perm read                /etc/rc.config.d/netconf-ipv6

/*
*****
* Trusted Mode database
*****
*/
perm nsearch            /tcb
perm nsearch            /tcb/files
perm all                /tcb/files/auth

```

Appendix B SRP Serviceguard Default Route Script

The following script can be used by a Serviceguard package to assign a default route for an IP address associates with an SRP. This script is included with the SRP Serviceguard Reference Implementation and is installed with the SRP product at:

/opt/hpsrp/example/serviceguard/srp_as_sg_package/srp_route_script

```
# Copyright (c) 2009 Hewlett-Packard Development Company L.P.
#
# This script runs the 'route' command to manage source based routing
entry
# for the SRP.
#
# This script should be configured into the package configuration file
# as the first "external_script" parameter entry. It will be executed
# right after Serviceguard IP addresses assignment during package start
time,
# and before removing IP addresses during package halt time.
#
# This script uses the environment variable SRP_SG_MANAGED_IP and
# SRP_SG_GATEWAY. The environment variables must be set in the
# srp_script.incl file in the same directory as this script.
#

#####
# Source utility functions.
#####

if [[ -z $SG_UTILS ]]
then
    . /etc/cmcluster.conf
    SG_UTILS=$SGCONF/scripts/mscripts/utils.sh
fi

if [[ -f ${SG_UTILS} ]]; then
    . ${SG_UTILS}
    if (( $? != 0 ))
    then
        echo "ERROR: Unable to source package utility functions file:
${SG_UTILS}"
        exit 1
    fi
else
    echo "ERROR: Unable to find package utility functions file:
${SG_UTILS}"
    exit 1
fi

#####
#
# Get the environment for this package through utility function
# sg_source_pkg_env().
#
#####

sg_source_pkg_env $*
```

```

#####
#
# Get the SRP environment from
"/etc/cmcluster/hpsrp/<srp>/srp_script.incl"
#
# Environemnt variable example: use a local gateway on the host
# SRP_SG_MANAGED_IP[0]="192.0.0.99"
# SRP_SG_GATEWAY[0]="192.0.0.99"
#
# Environemnt variable example: use a remote gateway
# SRP_SG_MANAGED_IP[1]="10.1.1.99"
# SRP_SG_GATEWAY[1]="10.1.1.1"
#
#####

. `dirname $0`/srp_script.incl

#####
#
# Functions
#
#####

# add routing entry
function srp_route_add
{
    # run 'route' command for each IP address
    rval=0
    index=0
    last_index=${#SRP_SG_MANAGED_IP[@]}
    while [ "$index" -lt "$last_index" ]
    do
        srp_ip="${SRP_SG_MANAGED_IP[$index]}"
        srp_gateway="${SRP_SG_GATEWAY[$index]}";
        if [ -z "$srp_ip" ] # skip empty slot in the array
        then
            let index=$index+1
            let last_index=$last_index+1
            continue
        fi
        if [ "$srp_ip" = "$srp_gateway" ]
        then
            # use local IP as gateway
            msg=$(/usr/sbin/route add default $srp_gateway 0 \
                source $srp_ip 2>&1)
        else
            # use remote gateway
            msg=$(/usr/sbin/route add default $srp_gateway 1 \
                source $srp_ip 2>&1)
        fi
        if (($? != 0)); then
            print "ERROR: $msg" >$2
            rval=1
        fi
        let index=$index+1
    done
    return $rval
}

# delete routing entry

```

```

function srp_route_delete
{
    # run 'route' command for each IP address
    rval=0
    index=0
    last_index=${#SRP_SG_MANAGED_IP[@]}
    while [ "$index" -lt "$last_index" ]
    do
        srp_ip="${SRP_SG_MANAGED_IP[$index]}"
        srp_gateway="${SRP_SG_GATEWAY[$index]}";
        if [ -z "$srp_ip" ] # skip empty slot in the array
        then
            let index=$index+1
            let last_index=$last_index+1
            continue
        fi
        if [ "$srp_ip" = "$srp_gateway" ]
        then
            # use local IP as gateway
            emsg=$(/usr/sbin/route delete default $srp_gateway 0 \
                source $srp_ip 2>&1)
        else
            # use remote gateway
            emsg=$(/usr/sbin/route delete default $srp_gateway 1 \
                source $srp_ip 2>&1)
        fi
        if (($? != 0)); then
            print "ERROR: $emsg" >$2
            rval=1
        fi
        let index=$index+1
    done
    return $rval
}

#####
# main routine
#####

sg_log 5 "SRP routing entry configuration script"

#####
#
# Customer defined external script must be specified with three required
# entry points: start, stop, and validate.
#
# It's not recommended to add additional entry points to the script
# due to potential name space collision with future Serviceguard
# releases.
#
#####

typeset -i exit_val=0

case ${1} in
    start)
        srp_route_add
        exit_val=$?
        ;;

```



```
stop)
    srp_route_delete
    exit_val=$?
    ;;

validate)
    exit_val=0
    ;;

*)
    sg_log 0 "INFO: Unknown operation: $1"
    ;;
esac

exit $exit_val
```

Technology for better business outcomes

© Copyright 2010 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

5900-0911, August 2010



Get connected

www.hp.com/go/getconnected

Current HP drivers, support & security alerts
delivered directly to your desktop

