

Best practices for deploying HP-UX Secure Resource Partitions (SRP) for SAP

Technical white paper

Table of contents

Executive summary.....	2
Deploying SRP.....	3
SRP basic overview	3
SRP short comparison	4
Other considerations	5
Known limitations.....	8
Deployment best practices	11
Best practices for configuring SRP for SAP during an installation	11
Best practices for configuring SRP for the SAP production system	13
Best practices for configuring saposcol for several SAP systems	15
Best practices for configuring SRP for upgrading an SAP system	17
Best practices for configuring SRP for de-installing an SAP system	18
Example configurations	19
How to enable login to the INIT compartment.....	19
SRP rule file <i>SAP_RUN.h</i> for a running system.....	19
SRP rule file <i>SAP_INST.h</i>	20
SRP rule file <i>DEFAULT_SAPOSCOL.h</i> for non-saphostctrl use.....	21
SRP rule file <i>DEFAULT_SAPHOST.h</i> for <i>saphostctrl</i> use	21
Script <i>add_new_system</i>	21
Summary	23
Appendix – Useful commands	24
For more information.....	25



Executive summary

Consolidation of several SAP systems on one host might lead to questions like “How are resources distributed?” or “How can the SAP instances be secured from each other?” In addition to offering HP Virtual Machines (HP VM) or HP-UX Virtual Partitions (vPar), HP also offers another answer to this question: HP-UX Secure Resource Partitions (SRP).

With SRP, it is possible to create compartments to isolate processes or restrict access to directories on a host from each other, restrict port usage with the help of IPFilter and assign resources with the help of HP Process Resource Manager (PRM).

SRP offers lightweight partitioning. It is deployed easily, has a low performance overhead and it leaves a small footprint on the system. Only one operating system (OS) has to be maintained.

SRP uses components of the operating system such as PRM, IPFilter and HP-UX Security Containment and provides a simple configuration template. No additional support or license costs are required when using SRP in the HP-UX operating system.

This document gives an overview of how to use SRP for SAP and how to adapt a standard SAP system to make it run within an SRP compartment. Compartment rule files for installing an SAP system, running it and blocking other SAP systems are provided here. For easier configuration, a script is offered in this document that simplifies adding the SAP specific rules for an SRP compartment.

Target audience: SAP consultants responsible for SAP consolidation or SAP technicians responsible to implement new OS features.

This white paper describes the tests performed in October 2009.

Deploying SRP

SRP basic overview

HP-UX Secure Resource Partitions (SRP) version 2 enables you to create and manage SRP compartments that provide isolated execution environments for applications. Each SRP compartment can have:

- A compartment home directory tree, which is isolated from other compartments
- A dedicated IP interface
- Isolated inter-process communication (IPC)
- A compartment-specific login environment
- Dedicated CPU and memory resources
- Per-compartment initialization and shutdown capabilities that function as would 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. SRP can be used easily within HP-UX Virtual Partitions (vPar) or HP VM.

The configuration data for an SRP compartment encompasses data for multiple HP-UX subsystems and features, including HP-UX Security Containment and HP PRM.

SRP identifies these data using tags, or special text identifiers. These references enable 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; deleting an SRP compartment removes all data configured for the compartment.

SRP does not deliver different OS environments for each compartment; only one OS environment is available per host. Having one available environment can be an advantage for OS administration but could have another impact when different OS environments are required for different applications. If different environments are required, vPar or HP VM could be used.

The performance overhead of SRP is very low because this solution comes with a very small footprint on the OS. The software lifecycle is also very short; SRP compartments can be created and deleted very easily.

SRP short comparison

SRP and vPars

These two technologies are complementary. vPar is used when more than one OS image is needed. SRP can be used on top of vPar.

Examples where SRP fits better than vPars (if only one of the technologies will be used):

- Use SRP for smaller systems or systems not supported by vPars
- Use SRP when resource utilization is a priority. While both vPars and SRP have low performance overhead, SRP shares the resources more dynamically; if a workload needs more resources, then they can be acquired more quickly and efficiently from another SRP compartment.
- Use SRP when efficient management of sub-core workloads is desired
- Use SRP to create or delete an SRP compartment very quickly
- Use SRP to save on additional license or support costs because they are included with HP-UX

Examples where vPars fits better than SRP (if only one is used):

- Use vPars when strong out-of-the-box isolation is required
- Use vPars when large workload and/or dedicated I/O resources are required
- Use vPars when independent OS environments are required
- Use vPars to handle almost all applications immediately as certain adaptations or modifications of the application might be required when using SRP

SRP and HP VM

While several SRPs could be deployed within an HP VM, typically either one or the other (SRP or HP VM) would be chosen.

Examples where SRP fits better than HP VM:

- Use SRP for very low performance overhead
- Use SRP for more granular resource sharing, including OS table space
- Use SRP to manage sub-core workloads more efficiently
- Use SRP to create or delete an SRP compartment very quickly
- Use SRP to save on additional license or support costs because they are included with HP-UX
- Use SRP for ease of use in implementation with existing data center management practices because additional management for the host OS and new data center practices are needed with HP VM

Examples where VM fits better than SRP:

- Use VM when stronger out-of-the-box isolation is desired
- Use VM when independent OS environments are required
- Use VM when a fully virtualized environment is required
- Use VM to run almost all applications immediately as modifications may be required in applications running in SRP

Table 1. Naming convention

SAP ID (SID)	SAP system identifier
DBSID	Database identifier
SAP INSTANCE Number (NR)	SAP system instance number, consisting of a two digit number from 00 to 98
SAP INSTANCE	SAP system instance, e.g. DVEBMGS<NR> for ABAP or SCS<NR> for JAVA™ instances
host_name/vhost	Name of the host on which SAP will be installed. May be the same as the hostname that resolves to the SRP compartment.
SRP compartment	SRP consists of compartments and additional options like IPFilter or PRM. For easy readability, referring to compartments in this document also includes all other features available for SRP

Other considerations

There are two approaches to configure SRP for SAP. Either, the SAP system already exists or the system does not yet exist and will be installed. This document describes the best practices for both approaches.

The installation model described in this white paper assumes compartment-specific binaries. This statement means that an SAP system and Oracle® database are installed for each SRP compartment. The installation can be done from the INIT compartment or from a new SRP compartment. The exception to this rule is the use of the executable, saposcol. This executable may exist only once per host and must to be run as a shared binary.

This document will only refer to standard SAP installations with Central Instance and Oracle 10g as the database on one host based on NetWeaver products starting with NetWeaver 04. SRP configurations for NetWeaver 7.0 ABAP or JAVA may be done in the same way. Other SAP installation options like non-default installation directory, distributed installations or dialog instance installation were not tested, the SAP_*.h compartment rule files can be used as a reference, but may need adaptation.

Directory considerations

The standard SAP values for the SAP directories are assumed. If different paths are used during the SAP installation, the rule files in this document will have to be adapted. The Oracle installation is done within an SRP compartment using the Oracle RUNINSTALLER for SAP. For each SRP compartment, a separate Oracle database will be installed, using the standard SAP directories for Oracle: /oracle/<DBSID>.

Commonly used SAP directories, like the transport directory, are made unique by the SAP ID in the examples used here; however, other differentiators may be used. In this case, the example rule files will have to be adapted.

To use SRP for SAP, several manual modifications have to be done for the SAP system; they are listed in the section, “Best practices for configuring SRP for the SAP production system”, in this white paper. With these configurations, an SAP system will run within an SRP compartment. With the compartment startup and shutdown option, it is also possible to start the SAP startup framework at compartment startup.

Hostname considerations

With SRP, it is possible to use the physical hostname for all SRP compartments or define a hostname for each SRP compartment. It is mandatory for each compartment to have a unique IP Address. It is dependent upon the planned SAP landscape as to whether or not a virtual host is used. If each SRP compartment has its own hostname, the SAP system should be installed with the virtual hostname option offered for sapinst. Changing the hostname after the SAP installation is very complicated and is not supported for the SAP J2EE engine part.

Note

Make sure during the planning phase that no SID, DBSID or instance number is used more than once on the host.

This document assumes that several SAP systems, isolated from each other, will be running on one host. Each SAP system will run in a separate SRP compartment with dedicated OS user login rights. The directories are not only isolated from each other, but the running processes cannot be shared between the compartments, with the only exception being the executable, saaposcol. Details about saaposcol can be found in the section of this white paper, "Known limitations."

In the section, "Example configurations," default directory permissions are listed for installing an SAP system, running the SAP system and blocking other SAP systems. These activities are primarily based on directory control.

IPFilter consideration

With SRP, it is also possible to use the IPFilter option for each SRP compartment. The tests conducted for this white paper used activated IPFilter, but since the ports required by SAP are very customer and use-case specific, no rule file for ports is provided in this document. If you want to use IPFilter, refer to the SAP documentation *TCP/IP Ports Used by SAP Applications* for the list of ports SAP requires.

Resource consideration

By default, SRP will configure all SRP compartments with the same share of total memory and CPU. The share enforcement will only take place when system CPU or memory limits are hit. Refer to the *HP-UX Secure Resource Partitions (SRP) Administrator's Guide HP-UX 11i v3* for more information on how to customize resource shares for an SRP compartment.

Being logged in as a root user in HP-UX, it is always possible to log in as any other user without providing a password. Even with SRP compartments this rule is valid as the same */etc/passwd* file is used and the root user ID (0) is the same for all root users on the system. Therefore, it is not possible to distinguish between the different root accounts for the different SRP compartments when calling "su".

For example, you may switch from the root user in SRP compartment SAPSRP1 to the <sid>adm of SRP compartment SAPSRP2 without providing a password. Even if the user appears to be logged in, the rules of compartment SAPSRP1 still apply. The user environment cannot be accessed and the directories and processes of SAPSRP2 cannot be accessed. This issue is not an SRP-related security issue, but, rather, it is common to UNIX® systems.

Handle the login privileges for any root user in the usual security-sensitive way; that is, only selected personnel should know the root password.

Figure 1. Simplified SAP directory isolation in two compartments

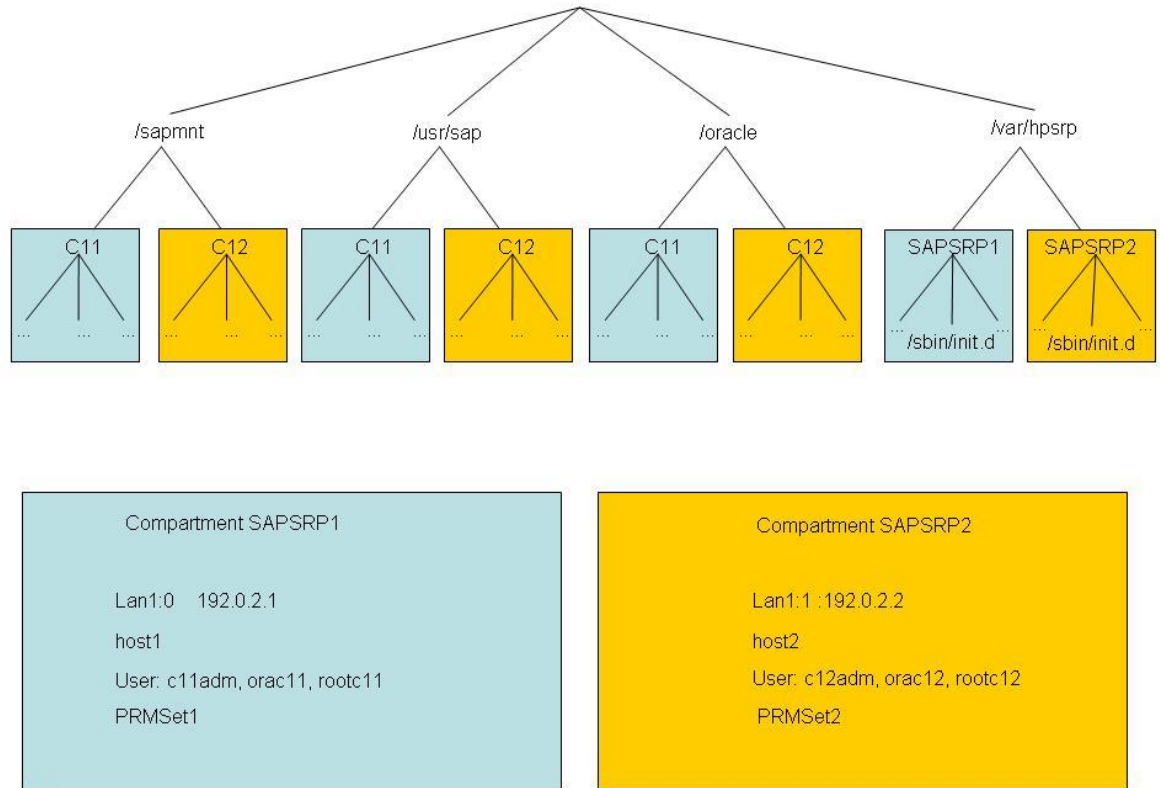


Figure 1 shows how the simplified SAP default directories of two SAP systems are assigned to two SRP compartments. No user assigned to SRP compartment SAPSRP1 can log in to SRP compartment SAPSRP2 or read or modify the content of its files or directories.

With different IPs and optional virtual hostnames connected to the SRP compartment, the two systems can be accessed individually. No process of compartment SAPSRP2 can be accessed from compartment SAPSRP1 or vice versa.

Known limitations

Saposcol

SAP delivers its own data collector, saposcol, to provide OS information (e.g. for CCMS). The data collector may run only once on a system. It addresses a shared memory segment with a predefined shared memory key that cannot be changed. If saposcol is started by an SAP instance in one compartment, other SAP instances cannot connect to saposcol due to the SRP compartment rules. They also cannot start their own saposcol because the shared memory segment is already occupied.

To avoid this problem, saposcol can be started in one of two ways: 1) in the INIT compartment as root user without further restrictions, or 2) in a separate compartment created only for saposcol. Which solution is used depends on the desired level of security.

If saposcol will be run in a secure environment, it is recommended to create a separate SRP compartment dedicated for saposcol with full access to saposcol and allowing IPC communication but deny execution access for all other compartments. With this approach, data collected by saposcol can be read by all SRP compartments but the process cannot be stopped by any of them. The known issue with this approach is that the data shown by saposcol is valid not only for the SRP compartment requesting the information but for the whole system as well. For example, the CPU utilization for the whole system is displayed, not just the CPU utilization for the compartment requesting the information.

Display problem in transaction ST06

The Collector status display in SAP transaction ST06 also has limitations. The information is not read from the shared memory, but from the output of calling saposcol. Calling saposcol is prohibited in the default examples so that saposcol cannot be stopped accidentally by any SAP system.

Figure 2. Display issue in ST06

```
Version          :COLL 20.95 711 - V2.7.1 2008-08-11 HP-UX IA64
Collecting Since :Fri Oct 23 14:56:11 2009
Operating System :HP-UX hsi020 B.11.31 U ia64 2635313719
Interval         :10

> saposcol -s

*****
Collector Versions :
  running : COLL 20.95 711 - V2.7.1 2008-08-11 HP-UX IA64
  dialog  : COLL 20.95 711 - V2.7.1 2008-08-11 HP-UX IA64
Shared Memory     : attached
Number of records : 18238
Active Flag       : active (01)
Operating System  : HP-UX hsi020 B.11.31 U ia64 2635313719
Collector PID     : 26420 (6734)
Collector         : not running (process ID not found)
Start time coll.  : Fri Oct 23 14:56:11 2009

Current Time      : Thu Oct 29 15:51:38 2009

Last write access : Thu Oct 29 15:51:12 2009

Last Read Access  : Thu Oct 29 15:51:38 2009

Collection Interval : 60 sec (next delay).
Collection Interval : 61 sec (last ).
Status              : free
Collect Details     : required
Refresh             : required

Header Extention Structure
Number of x-header   Records : 1
Number of Communication Records : 60
Number of free Com.  Records : 60
Resulting offset to 1.data rec. : 61

Trace level         : 2

Collector in IDLE - mode ? : YES
  become idle after 300 sec without read access.
  Length of Idle Interval : 60 sec
  Length of norm.Interval : 10 sec
*****
```

If the status overview in ST06 is mandatory in the user environment, then consider enabling the execution rights for saposcol in the respective SRP compartment rule file.

Note

Please note that in this case, saposcol can be stopped for the whole system from just one SAP system.

Even if the execution access is allowed, the status display is not correct. In figure 2, the line item Collector in the status overview shows “not running (process ID not found).” The reason why is because SAP gets the process ID (pid) of saposcol out of the shared memory and checks on the OS to see if the pid still exists. Since saposcol is running in a separate compartment, the pid cannot be detected and this error message appears.

To avoid the error message, enter *send signal <SAPOSCOL-compartment>* in the SRP rule file of the respective SRP compartment for each compartment in order to communicate with saposcol, then activate the new rules with the command *setrules*.

The configuration examples in this document describe the restricted approach without signaling to the saposcol compartment, accepting the display issue in ST06.

Currently saposcol is not capable of displaying the data per compartment if PRM is used. The data shown is valid for the complete host and all systems.

Figure 3. Overview of saposcol compartment

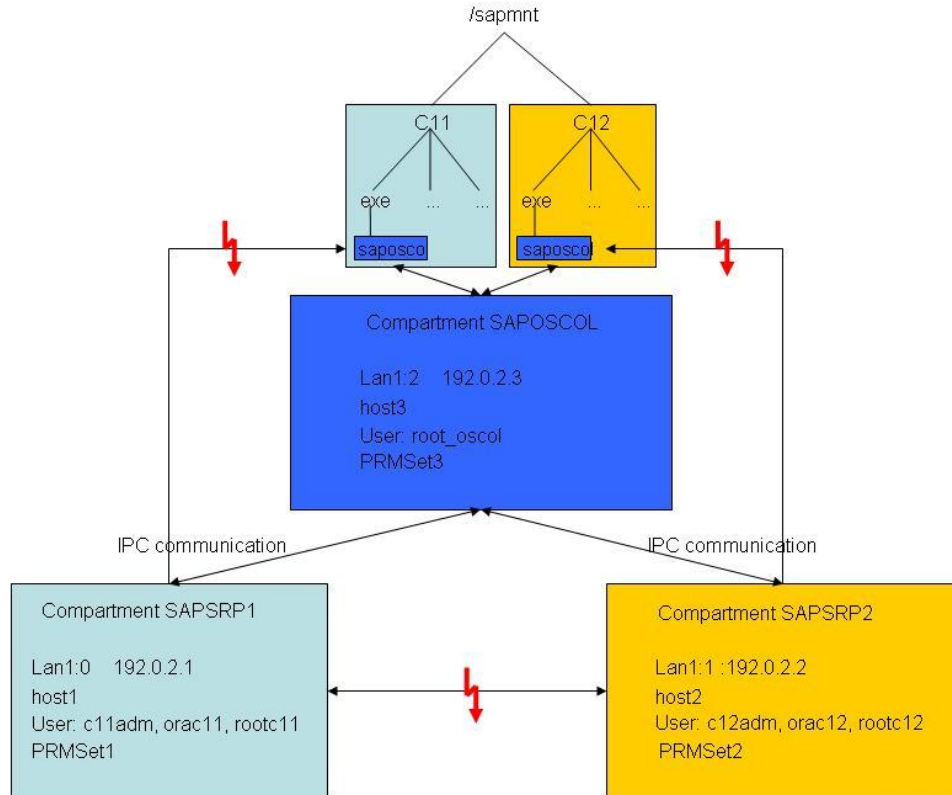


Figure 3 shows the setup of a separate SAPOSCOL compartment. The two SRP compartments do not have execution privileges for their saposcol executables. Saposcol can only be started by a root user in the SAPOSCOL compartment. Data collected by saposcol can be read by the SRP compartments via IPC communication.

Deployment best practices

Operating system choice

The following OS requirements are needed for using SRP with SAP:

- HP-UX 11i version 3 (11.31)
- Minimum OS patch requirements for SAP according to SAP note 837670
- HP-UX 11i kernel requirements for SAP according to SAP Note: 172747

The following product versions were used for the reference architecture:

- HP-UX Secure Resource Partitions and Configuration Manager: HP-UX-SRPA.02.00.001
- HP-UX Secure Shell: SecureShell A.05.20.006,
- If IPFilter will be used:
 - IPFilter A.11.31.16
 - OpenSSL Secure Network Communications Protocol : A.00.09.08k.003

Best practices for configuring SRP for SAP during an installation

SRP with SAP configurations

If a high separation level is required during an installation, the SRP compartment should be created before starting the SAP installation. The access to other already existing SAP systems in other SRP compartments on the host will then be limited.

For the installation of an SAP system, more access rights to directories like */oracle* have to be provided. This has to be changed after the SAP installation.

Note

After the installation of SAP within an SRP compartment, follow the instructions in the section of this white paper, "Best practices for configuring SRP for the SAP production system."

To configure a completely new SAP system with SRP usage, follow the next steps:

1. Create the file system and check the OS patch level and kernel tunables as described in the respective SAP installation guide and Note: 837670 and Note: 172747.
The directory structure required by SAP should be created before creating the individual SRP compartments and starting the installation. Refer to the white paper section, "SRP rule file SAP_INST.h" for which directories must exist. If they do not exist, *sapinst* will fail, because the parent directories like */oracle* or */sapmnt* do not have write permission per definition of the SRP compartment rule files.
2. Create the SAP users and groups for the SAP system as described in the SAP OS dependency guide. Required users are the *<sid>adm*, the *ora<sid>* user and a root user with the respective OS groups.
3. Create a new OS login group used for the login to the SRP compartment. Add the created users required for SAP to this group.
4. Define an IP Address for the SRP compartment to be created. Decide if a hostname, other than the physical hostname, will be used for the SRP compartment; if so, use this hostname

later for the SAP installation as the virtual hostname (vhost). If each SAP system will run logically on a different host, the use of virtual hostnames is recommended.

5. Configure on the host in the INIT compartment as root user with the command *srp_setup*. For details about how to use *srp_setup*, refer to the *SRP Administrator's Guide*. At a minimum, the following services must be activated:
 - *cmpt*
 - *admin*
 - *init*
 - *login*
 - *network*
 - *provision*
6. After initially calling *srp_setup*, login problems with another user than root to the INIT compartment might occur. See the section, "How to enable login to the INIT compartment" in this white paper for a solution to the problem.
7. Create the SRP compartment and configure the base template with the optional hostname, IP-Address and login group with the command "*srp -a <compartment>*" as described in the *SRP Administrator's Guide*. Include the *sshd* template.
8. To include the SAP specific rule files, copy the script from the section of this white paper, "Script *add_new_system*", into a new file on your system. Copy the file *SAP_INST.h* to */etc/cmpt/include*. Grant execution permission to the script *add_new_system* and call it as root user in the INIT compartment:
add_new_system <SRP compartment> <SID> <DBSID> <Systemnr> INST
The allow rule file for the compartment and a general Block rule file for other compartments will be created and added to the compartment as a custom include file.
9. If the *IPFilter* service will be used, refer to the SAP document, *TCP/IP Ports Used by SAP Applications* to see which ports have to be enabled.
10. For SAP products based on 6.40 and following:
Log in to your newly created SRP compartment with the root user assigned to this compartment using *ssh*. Start the *sapinst*. If a virtual hostname is used for the SRP compartment, use the *sapinst* option "*SAPINST_USE_HOSTNAME= <hostname>*".
11. Follow the *sapinst* routine as described in the respective SAP installation document. If another Oracle listener already exists on the host (e.g. in another SRP compartment), make sure to change the default entry of the port of Oracle listener during the parameter input phase; it cannot be equal to any other port used on the system.

12. Only for Oracle installation:
If the directory `/oracle/oralInventory` already exists on the host created by another Oracle installation in another SRP compartment, the next Oracle installation will have issues that can be avoided by utilizing either of the next steps:
 - Rename the existing directory but be very careful not to delete the directory as it will be needed for future Oracle updates or upgrades. If an Oracle upgrade is planned for later on, rename the directory back to "oralInventory".
 - Start the Oracle `RUNINSTALLER` and Oracle patch `runInstaller` with the option `-invPtrLoc` to define the new Oracle Inventory directory. For example:

```
RUNINSTALLER -invPtrLoc
/var/hpsrp/<compartment>/ora_ux10/db_1/oralnst.loc
```
13. After the SAP installation completes, configure the system as described in the section of this white paper, "SAP modifications for SRP."

Best practices for configuring SRP for the SAP production system

SAP modifications for SRP

To use the SAP system to its fullest potential, modify the SAP system as follows:

1. Rename the Transport Directory.
Use a unique name on the host for the new transport directory if several SAP systems exist on the host. For example, the SAP ID could be used: `/usr/sap/trans_<SID>`.

Adapt the instance profile parameters `DIR_TRANS` to `/usr/sap/trans_<SID>` and `DIR_EPS_ROOT` to `/usr/sap/trans_<SID>/EPS` and restart the system. This restart can be done later when the SRP compartment is restarted.

Rename the transport directory path in transaction STMS to the new name.

2. For Systems ≥ 7.00 only, adapt startup framework as follows:
 - Rename the `sapSERVICE` file.
If several SAP systems are installed on your host, rename the `sapSERVICE` file to something unique. A unique hostname could be used to identify the different `sapSERVICE` files. In this document, the use of the SAP ID is assumed.

```
mv /usr/sap/sapSERVICE /usr/sap/sapSERVICE_<SID>
```
 - If there are already several systems on the host with entries in the `sapSERVICE` file, split the file and create a new file for each system with only the respective entries for this system.
 - To enable the automatic startup of the `sapSTARTSRV` service after a compartment start, the `sapINIT` script has to be relocated.
 - Move the file `/sbin/init.d/sapINIT` to the `init.d` directory of your compartment:

```
mv /sbin/init.d/sapINIT /var/hpsrp/<compartment>/sbin/init.d
```

 In the `sapINIT` file edit the value for the parameter `PATH`. Change the value from `"/sbin"` to `"/var/hpsrp/<compartment>/sbin"`
 If the `sapSERVICE` file was renamed, also change the value of `SAPSERVICE_PATH` to the new `sapSERVICE` name.
 - Create a logical link to enable the automatic start of the `sapSTARTSRV` process:

```
ln -s /var/hpsrp/<Compartment>/sbin/init.d/sapINIT
/var/hpsrp/<Compartment>/sbin/rc<n>.d/SAPINIT
```

3. For Oracle installations only:

- Rename the oracle-client directory to something unique on the system, e.g. `/oracle/client_<SID>`.
The compartment rules have to be adapted for this action. Most important to note is that while you are modifying the instantclient, full permission must be given to Oracle or else the task will have to be executed as root user in the INIT compartment.
- If Oracle 10g is used, go to the instantclient directory and adapt the logical link of the instantclient:

```
cd /oracle/client_<DBSID>/10x_64
rm -r instantclient
ln -s /oracle/client_<DBSID>/10x_64/instantclient_10204 instantclient
```
- The environment of the <sid>adm user has to be adapted to the new directory. The respective entries can be found in the home directory of the user in the files `".dbenv.sh"`, `".dbenv_<host>.sh"`, `".dbenv.csh"` and `".dbenv_<host>.csh"`.
- J2EE specific
If a J2EE engine is used, the RDBMS driver location has to be adapted.
- Go to the configtool directory, which is typically `/usr/sap/<SID>/<INSTANCE><NR>/j2ee/configtool`.
With a text editor change the value of the parameter `rdms.driverLocation` in the file `configtool.properties` to the new client directory, e.g. :
`/oracle/client_<DBSID>/10x_64/instantclient/odbc14.jar`
 - Start `configtool.sh` and change the value of the parameter `rdms.driverLocation` to the new value under
`cluster-data -> instance _ID<nnn> -> server_ID<nnn> -> managers -> Configuration Manager`
and
`cluster-data -> instance _ID<nnn> -> dispatcher_ID<nnn> -> managers -> Configuration Manager`
 - Go to directory `/usr/sap/<SID>/<INSTANCE><NR>/j2ee/cluster/bootstrap` and edit file `bootstrap.properties` manually. Change the value of parameter `rdms.driverLocation` to the new client directory.
 - Change to the directory `/usr/sap/<SID>/<INSTANCE><NR>/SDM/programs/config` and edit file `sdmrepository.sdc`. Search for the old jdbc entry and change it to the new one.
 - Change to the directory `/usr/sap/<SID>/<INSTANCE><NR>/j2ee/deploying` and edit file `rdm.properties`. Change the value of the parameter `rdms.driverLocation` to the new value
 - Change to the directory `/usr/sap/<SID>/<INSTANCE><NR>/j2ee` and edit file `templateconfiguration.properties`. Change the value of the parameter `RDBMS_DRIVER_LOCATION` to the new value.
 - In the SAP profile directory edit the Instance profile (e.g. `/sapmnt/<SID>/profile/<INSTANCE_PROFILE>`) and change the value of the parameter `j2ee/dbdriver` to the new value.
 - A restart of the J2EE is necessary to make the changes effective. This restart can be done later when the SRP compartment is restarted.

4. Now SRP has to be configured on the host. This is done in the INIT compartment as root user with the command `srp_setup`. For details how to use `srp_setup`, refer to the *SRP Administrator's Guide*.
At a minimum, the following services have to be activated:
 - `cmpt`
 - `admin`
 - `init`
 - `login`
 - `network`
 - `provision`
5. After initially calling `srp_setup`, login problems with another user than root to the INIT compartment might occur. See the section of this white paper, "How to enable login to the INIT compartment" for a solution.
6. Create the SRP compartment and configure the base template with IP-Address and optional hostname. Create an SRP specific UNIX login group and add the SAP users to this group. Use this group as login group when creating the SRP compartment with the command "`srp -a <compartment>`" as described in the *SRP Administrator's Guide*. Include the `sshd` template.
7. If the IPFilter service will be used, refer to the SAP document *TCP/IP Ports Used by SAP Applications* to see which ports have to be enabled.
8. To include the SAP specific rule files, copy the script from section of this white paper, "Script `add_new_system`" into a new file on your system. Also copy the file `SAP_RUN.h` to `/etc/cmpt/include`. Grant execution permission to the script `add_new_system` and call it as root user in the INIT compartment:

```
add_new_system <SRP compartment> <SID> <DBSID> <Systemnr> RUN
```

If the rule file for the installation of an SAP system is already included in the compartment, it will be replaced with a new allow rule file. If the BLOCK rule file for all compartments was not yet extended by the SAP system, this extension will be done as well. The compartment will be updated with the new rule files.
9. Optional: Create separate compartment for `saposcol`
If the usage of `saposcol` will also be secured, the following section "Best practices for configuring `saposcol` for several SAP systems" should be taken into account. As only one `saposcol` may run per OS instance, a separate compartment has to be created to grant all SAP instances access to `saposcol`. For details how to create this compartment, see the section of this white paper, "Best practices for configuring `saposcol` for several SAP systems". If you are using the default rule file created with the script `add_new_system`, the execution permission for `saposcol` is already denied. `Saposcol` has to be started manually once as root user in the `saposcol` compartment.

Best practices for configuring `saposcol` for several SAP systems

To get the highest level of isolation between different SAP systems in SRP compartments, it is recommended to create a special compartment only for `saposcol`. Running `saposcol` requires the login of a root user, the execution permission for `saposcol` and IPC access to the other compartments to gather process information.

For systems not using saphostctrl, follow these guidelines:

1. If `saposcol` is running, login to the SRP compartment as root in which `saposcol` is running and stop the process by calling "`saposcol -k`"
2. Create a new SRP compartment with own IP and LAN interface. The default SRP settings can be used for this. Select an existing root user for the compartment login.
3. Customize the rule file `DEFAULT_SAPOSCOL.h` and include it in the SRP compartment.
4. Login to the SAPOSCOL SRP compartment as root user and start `saposcol` by calling "`/sapmnt/<SID>/exe/saposcol`"

As described in the section of this white paper, "Known limitations", different problems are related with the `saposcol` executable. The `DEFAULT_SAPOSCOL.h` rule file is configured in such a way that no SRP compartment may start or stop `saposcol`, leading to display issues of the `saposcol` status in transaction ST06.

If it is necessary to view the actual status in ST06 and accept the risk of one SAP system stopping `saposcol` for all systems installed on this host, modify the include file for `saposcol` in the following way:

- Add the entry "`send signal <SAPOSCOL-compartment>`" in the SRP compartment rule file for each SRP compartment that requires execution rights for `saposcol`
- Delete the entry "`perm none /sapmnt/_SAP_SID_/exe/saposcol`" in the respective SRP compartment rule files
- Set the new rules with the command "`setrules`"

For systems using saphostctrl, follow these guidelines:

Starting with NetWeaver 7.10, a new monitoring tool was introduced by SAP, the SAPHOST agent. It controls the start and stop of an executable such as `saposcol`. It is also run only once on a system and therefore must be included in a separate SPR compartment.

1. If `saphostctrl` is already running, stop it as root user from the compartment it was started from with the command:
`/usr/sap/hostctrl/exe/saphostexec -stop`
2. Create a new SRP compartment with its own IP Address and LAN interface. The default SRP settings can be used for the other settings
Customize the rule file `DEFAULT_SAPHOST.h` and include it in the compartment.
3. Create an empty `sapservice` file, e.g. `/usr/sap/sapservice_saphostctrl`
4. To enable the automatic startup of the `sapstartsrv` service after an SRP compartment start, the `sapinit` script has to be relocated.
 - Copy the file `/sbin/init.d/sapinit` or any other `sapinit` file from another SAP SRP compartment to the `init.d` directory of your compartment:
`cp /sbin/init.d/sapinit /var/hpsrp/<compartment>/sbin/init.d`
 - In the `sapinit` file, edit the value for the parameter `PATH`. Change the value from "`/sbin`" to "`/var/hpsrp/<compartment>/sbin`"
 - Change the value of `SAPSERVICE_PATH` to the new `sapservice` name, in this case "`/usr/sap/sapservice_saphostctrl`"
 - Create a logical link to enable the automatic start of the `sapstartsrv` process:
`ln -s /var/hpsrp/<SAPOSCOL-compartment>/sbin/init.d/sapinit /var/hpsrp/<SAPOSCOL-compartment>/sbin/rc<n>.d/SAPINIT`

5. In the rule file for the SRP compartment, change the “*perm none*” entry from
`/sapmnt/<SID>/exe/saposcol`
to
`/usr/sap/hostctrl`
6. Activate the change to the SRP compartment with the command “*setrules*”
7. Either start the SAPHOST agent by restarting the SAPOSCOL SRP compartment or by logging in to the SAPOSCOL SRP compartment and calling “`/usr/sap/hostctrl/exe/hostexecstart`”.

If it is required to see the actual status in ST06 and accept the risk of one SAP system stopping saposcol for all systems installed on this host, modify the include file for saposcol in the following way:

- Add the entry “*receive signal <SAP-compartment>*” in the SAPHOST rule file for each compartment that requires execution rights for saposcol
- Delete the entry “*perm none /usr/sap/hostctrl*” in the respective SRP compartment rule files
- Set the new rules with the command “*setrules*”

Best practices for configuring SRP for upgrading an SAP system

The permissions for upgrading an SAP system to 7.01 or 7.10 or higher are the same as for running an SAP system.

After upgrading to NetWeaver 7.10, where the new SAPHOST structure for monitoring the system is used, the permissions and structure of the saposcol compartment have to be adapted. See SAP Note: 1031096 to determine how to setup the SAPHOST agent.

Note

If the target release is based on NetWeaver 7.00 or lower, other file system permissions might be required. Adoption of the compartment rule file might be necessary in this case; this particular setup was not tested in the scenario presented here.

Best practices for configuring SRP for de-installing an SAP system

In the life cycle of an SAP system, the last step is the de-installation of the system. With respect to SRP, the de-installation can be accomplished in one of two ways.

1. De-install SRP compartment first.
If it is not necessary to have a high isolation level during the de-installation of the SAP system, the easiest way is to delete the SRP compartment for the respective SAP system and start the SAP de-installation from the INIT compartment.
2. De-install SAP system first:
For a successful de-installation of an SAP system within an SRP compartment, the rule file for the installation (*SAP_INST.h*) is used. As de-installing an SAP system requires the same privileges as an installation, call the script *add_new_sap.h* with the option INST.

The directories */sapmnt/<SID>* and */usr/sap/ccms* will not be removed by the *sapinst*.

The directory */sapmnt* does not have full privileges to delete the directory *<SID>*. If there are still other systems running on the host, the directory */sapmnt/<SID>* has to be deleted manually after *sapinst* finishes. For this deletion, full access to */sapmnt* has to be given for the SRP compartment and revised afterwards if the SRP compartment will not be deleted.

The CCMS directory */usr/sap/ccms* may not be removed, if another SAP system is still using this directory, so this error message can be ignored.

Directories created manually for using SRP, like */oracle/client_<SID>* and */usr/sap/trans_<SID>* also have to be removed manually. *Sapinst* does not recognize non-default directories.

Example configurations

How to enable login to the INIT compartment

After calling `srp_setup`, login problems might occur if the default root user is not used to login to the INIT compartment.

Starting with SRP 2.1 the following process can be used to add additional users and groups to login to the INIT compartment:

- 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`

For the SRP version 2.0 one additional step is required before enabling any user or groups:

- `roleadm add SRPlogin-init`
- `authadm assign SRPlogin-init hpux.security.compartment.login "init"`

SRP rule file `SAP_RUN.h` for a running system

```
// Variables to be replaced:
//   _SAP_SID_ = SAP SID in capital letters
//   _DB_SID_ = database SID in capital letters
//   _SAP_SYSNUM_ = SAP System number
//   _sap_sid_ = SAP SID in lower letters

// read access rules

perm nsearch /oracle

perm nsearch,read /oracle/client__SAP_SID_
perm nsearch,read /usr/sap
perm nsearch,read /usr/sap/tmp

// all access rules

perm all /oracle/_DB_SID_
perm all /oracle/_DB_SID_/102_64

perm all /home/_sap_sid_adm

perm nsearch /sapmnt
perm all /sapmnt/_SAP_SID_

perm nsearch /usr/sap
perm all /usr/sap/_SAP_SID_
perm all /usr/sap/trans__SAP_SID_

//not necessary for 6.40 systems
perm all /usr/sap/sapservices__SAP_SID_

//not necessary for 6.40 systems
```

```

perm nsearch /usr/sap/ccms
perm all /usr/sap/ccms/_SAP_SID__SAP_SYSNUM_

//none access rules
perm none /sapmnt/_SAP_SID_/exe/saposcol

```

SRP rule file SAP_INST.h

```

// Variables to be replaced:
//   _SAP_SID_ = SAP SID in capital letters
//   _sap_sid_ = SAP SID in lower letters
//   _DB_SID_ = database SID in capital letters

// read access rules
perm nsearch,read /oracle

// all access rules

perm all /opt

perm nsearch /sapmnt
perm all /sapmnt/_DB_SID_

perm nsearch /oracle
perm all /oracle/_DB_SID_
perm all /oracle/_DB_SID_/102_64
perm all /oracle/client
perm all /oracle/oralInventory
perm all /oracle/stage

perm nsearch /home
perm all /home/_sap_sid_adm

perm nsearch /usr
perm all /usr/sap
perm all /usr/sap/ccms/
perm all /usr/sap/trans
perm all /.sdtgui

perm all /opt/java1.4
perm all /sbin/init.d

//installation directory, might be different in customer systems
perm all /tmp/install

```

SRP rule file *DEFAULT_SAPOSCOL.h* for non-saphostctrl use

```
// Edit this file manually
// Variables to be replaced:
//   _SAP_SID_ for one SAP system on this host
//   _SAP_COMPARTMENT_ for each SAP SRP compartment

//for one SAP system on the host
Perm all      /sapmnt/_SAP_SID_/exe/saposcol
Perm all      /usr/sap/tmp

//for each SAP SRP compartment on the host
grant ipc <_SAP_COMPARTMENT_ >
...
access ipc,fifo,uxsock *
```

SRP rule file *DEFAULT_SAPHOST.h* for saphostctrl use

```
// Edit this file manually
// Variables to be replaced:
//   _SAP_COMPARTMENT_ for each SAP compartment
//   _sap_sid_ = SAP SID in lower letters

Perm nsearch,read /usr/sap
Perm all          /usr/sap/hostctrl
Perm all          /usr/sap/tmp

//for each SAP compartment on the host
grant ipc <_SAP_COMPARTMENT_ >
...
access ipc,fifo,uxsock *
```

Script *add_new_system*

With the following script, the SAP directories of a new SAP system will be added to a global BLOCK File. This block rule file will be included in the rule file for the compartment created for the SAP system. The directories will get the respective allow rights for the SAP system. The general idea behind this is to block all SAP-related directories on the system and then grant access only to the ones associated with the SRP compartment.

To use this script, copy the files *SAP_RUN.h* and *SAP_INST.h* to the directory */etc/cmpt/include*. The variables will be replaced for you by the script.

```
#!/usr/bin/csh
# Script to create an entry in the block file for the new SAP system and an allow file for the
compartment

if ( $#argv == 5 ) then
    echo "Start rule file creation and activation for SAP system $1\n\n"
else
    echo "usage: $0 <SRP> <SAP_SID> <DB_SID> <SAP_SYSNUM> <RUN|INST>"
    exit 1
endif
```

```

set BLOCK_FILE='/etc/cmpt/include/SAP_BLOCK.h'

# Build custom include file for SAP to allow access to directories

set sap_sid = `echo "$2" | tr "[A-Z]" "[a-z]"`
set sidadm = `echo "$sap_sid"`adm

if (-e "/etc/cmpt/$1_$5.h") then
    echo "File /etc/cmpt/$1_$5.h already exists. Existing rule file will be used.\nlf it will be
replaced, delete it manually"
else
    sed -e "s/_DB_SID_/$3/g" -e "s/_SAP_SID_/$2/g" -e "s/_sap_sid_/$sap_sid/g" -e
"s/_SAP_SYSNUM_/$4/g" /etc/cmpt/include/SAP_$5.h >> /etc/cmpt/
$1_$5.h
endif

# add SAP directories to global block file so all other SAP instances don't
# have access to these directories.

if ("grep $2 $BLOCK_FILE" == 1) then
    echo "\n/* added for SAP instance $1 on `date` */\n" >> $BLOCK_FILE
    echo "perm none /sapmnt/$2">> $BLOCK_FILE
    echo "perm none /usr/sap/$2">> $BLOCK_FILE
    echo "perm none /oracle/$3">> $BLOCK_FILE
    echo "perm none /home/$sidadm">> $BLOCK_FILE
    echo "perm none /oracle/client_$3">> $BLOCK_FILE
    echo "perm none /usr/sap/ccms/$2_$4">> $BLOCK_FILE
    echo "perm none /usr/sap/trans_$2">> $BLOCK_FILE
    echo "perm none /sapmnt/$2/exe/saposcol">> $BLOCK_FILE
    echo "" >> $BLOCK_FILE
else
    echo "Block file $BLOCK_FILE already exists with an entry for SAP system $2.\n Current Block file
will be included in compartment $1\n\n"
endif

if ("grep cust_inc_$4 /etc/cmpt/$1.rules" == "1") then
    srp -b -a $1 -t custom -s cmpt -id cust_inc_$4 cmpt_rule_file=/etc/cmpt/$1_$5.h
else
    srp -b -r $1 -t custom -s cmpt -id cust_inc_$4 cmpt_rule_file=/etc/cmpt/$1_$5.h
endif

exit 0

```

Summary

Using SRP for several SAP systems isolates the SAP systems from each other by using compartments. It is possible to define different levels of isolation, from user login and directory access to port blocking, by using IPFilter. Although the SRP compartments always use the same OS environment, virtual hostnames can be realized by using SRP and the resource management can be accomplished on a very granular basis per SRP compartment using PRM.

Even though SRP compartments can be configured very easily with a few commands, several adaptations are required when using several SAP systems in different SRP compartments on one host. With these adaptations, SAP can be fully run within an SRP compartment.

vPar or HP VM should be used if different OS environments are required for different SAP systems or if the manual adaptations of the SAP systems are not preferable.

Appendix – Useful commands

- Report the current compartment
getprocxsec -c
- Report the compartment a process is running in
getprocxsec -c <pid>
- Add custom template to compartment
srp -add <compartment> -t custom
- Replace custom template in compartment
srp -r <compartment> -t custom
- Start SRP compartment
srp -start <compartment>
- Stop SRP compartment
srp -stop <compartment>
- List the IPFilter configuration
/etc/opt/ipf/ipf.conf
or
srp -l <compartment> -v -s ipfilter
- Login to SRP compartment
ssh -l <user> <SRP-IP Address>
- Add the sshd template to an existing SRP compartment with the service cmpt and provision
srp -add <compartment> -template sshd -s cmpt,provision
- Setup SRP
srp_setup
- Show base template parameters
srp -help -template base
- Create a base SRP compartment
srp -add <compartment>
- List the configuration data
srp -list <compartment>
- List the configuration data for sshd
srp -list <compartment> -v -t sshd
- Replace the PRM configuration values
srp -replace <compartment> -s prm
- Delete SRP compartment
srp -delete <compartment> -batch

For more information

HP-UX Secure Resource Partitions (SRP) Administrator's Guide HP-UX 11i v3

<http://docs.hp.com/en/5992-4679/5992-4679.pdf>

Security Containment Administrator's Guide and Release Notes

<http://docs.hp.com/en/internet.html#Security%20Containment>

Role-Based Access Control Administrator's Guide and Release Notes

<http://docs.hp.com/en/internet.html#Role-Based%20Access%20Control>

Secure Shell login white paper

<http://docs.hp.com/en/5992-5374/5992-5374.pdf>

Secure Shell docs

<http://docs.hp.com/en/internet.html#Secure%20Shell>

SAP installation documents

<http://service.sap.com/instguides>

SAP Notes: 837670 and 172747

<http://service.sap.com/notes>

TCP/IP Ports Used by SAP Applications

<http://www.sdn.sap.com/irj/scn/index?rid=/library/uuid/4e515a43-0e01-0010-2da1-9bcc452c280b>

To help us improve our documents, please provide feedback at

http://h20219.www2.hp.com/ActiveAnswers/us/en/solutions/technical_tools_feedback.html.



© 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.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Java is a US trademark of Sun Microsystems, Inc. UNIX is a registered trademark of The Open Group.

4AA0-2630ENW, Revision 3, March 2010

