Virtualvault Transition to HP-UX with Security Containment

Administrator’s Guide

Printed in U.S.A. 2006 (Version 1.2.2)
April 17, 2006
Legal Notices

The information contained in this document is subject to change without notice. Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Copyright © 2000-2006 Hewlett-Packard Development Company, L. P.

This document contains information which is protected by copyright. All rights are reserved. Reproduction, adaptation, or translation without prior written permission is prohibited, except as allowed under the copyright laws.

Corporate Offices:
Hewlett-Packard Co.
3000 Hanover St.
Palo Alto, CA 94304

Use, duplication or disclosure by the U.S. Government Department of Defense is subject to restrictions as set forth in paragraph (b)(3)(ii) of the Rights in Technical Data and Software clause in FAR 52.227-7013.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Use of this manual and flexible disc(s), compact disc(s), or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs may be made for security and back-up purposes only. Resale of the programs in their present form or with alterations is expressly prohibited.

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office.
# Table of Contents

Welcome to the Virtualvault Transition 5  
- Introduction 5  
  - Transition Overview 6  
  - User Defined Applications 7  
  - Unsupported Functionality 7  
- Virtualvault Transition Support 8  
- Identifying the Transition Steps 8  

Part I – The Transition Process 11  
1. Preparing to Move from Virtualvault to Security Containment 11  
  - Overview 11  
  - Export Virtualvault Configuration 11  
    - VVSnapshot Files 12  
    - Downloading and Installing the VVSnapshot Depot 13  
    - Executing the VVSnapshot Script 14  
    - Troubleshooting the VVSnapshot Script 14  
2. Installing the Security Containment System 15  
  - Overview 15  
  - Installing HP-UX 11iv2 15  
  - Required Software Depots 17  
  - Downloading Software Depots 18  
  - Using HP-UX Secure Shell (SSH) 18  
  - Hardening the Security Containment System 18  
    - HPUXHarden Files 19  
    - Installing the HPUXHarden Depot 20  
    - Executing the HPUXHarden Script 20  
    - Troubleshooting the HPUXHarden Script 21  
    - Manual Steps to Complete the Hardening Process 21  
  - Setting up Security Containment 22  
    - Installing Security Containment 22  
    - Enabling Security Containment Compartments 22  
    - Installing the VVTransition Depot 22  
    - Verify Security Containment LAN Setup 23  
  - Create System Recovery Tape 23  
3. Importing Virtualvault Data and Configuration Files 25  
  - Overview 25  
  - Import Virtualvault Configuration 25  
    - Transferring the VVSnapshot Staging Archive to the Security Containment System 25  
    - VVSCAssist Files 26
<table>
<thead>
<tr>
<th>Table of Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executing the VVSC Assist Script</td>
<td>27</td>
</tr>
<tr>
<td>Troubleshooting the VVSC Assist Script</td>
<td>27</td>
</tr>
<tr>
<td>Manual Steps to Complete the Import Process</td>
<td>28</td>
</tr>
<tr>
<td>Create System Recovery Tape</td>
<td>28</td>
</tr>
<tr>
<td>4. Configuring HP-UX HIDS for Alarms functionality</td>
<td>29</td>
</tr>
<tr>
<td>Introduction</td>
<td>29</td>
</tr>
<tr>
<td>Lockdown of HP-UX-HIDS using Security Containment</td>
<td>30</td>
</tr>
<tr>
<td>Configuring HP-UX HIDS</td>
<td>30</td>
</tr>
<tr>
<td>Creating HP-UX HIDS compartment</td>
<td>31</td>
</tr>
<tr>
<td>Configuring RBAC for HP-UX HIDS</td>
<td>31</td>
</tr>
<tr>
<td>Configuring the HP-UX HIDS user role</td>
<td>31</td>
</tr>
<tr>
<td>Configuring the HP-UX HIDS authorizations</td>
<td>32</td>
</tr>
<tr>
<td>Configuring the HP-UX HIDS command authorizations</td>
<td>32</td>
</tr>
<tr>
<td>Starting and Stopping the HP-UX HIDS Agent</td>
<td>33</td>
</tr>
<tr>
<td>5. Configuring stunnel for TGP functionality</td>
<td>35</td>
</tr>
<tr>
<td>Introduction</td>
<td>35</td>
</tr>
<tr>
<td>Lockdown of stunnel using Security Containment</td>
<td>36</td>
</tr>
<tr>
<td>Creating the stunnel user and group</td>
<td>37</td>
</tr>
<tr>
<td>Installing stunnel</td>
<td>37</td>
</tr>
<tr>
<td>Creating the stunnel chroot directory</td>
<td>37</td>
</tr>
<tr>
<td>Creating the stunnel log directory</td>
<td>38</td>
</tr>
<tr>
<td>Setting file permissions for stunnel</td>
<td>38</td>
</tr>
<tr>
<td>Setting IPFilter rules for stunnel</td>
<td>38</td>
</tr>
<tr>
<td>Creating the stunnel compartment</td>
<td>38</td>
</tr>
<tr>
<td>Configuring RBAC for stunnel</td>
<td>39</td>
</tr>
<tr>
<td>Configuring the stunnel user role</td>
<td>39</td>
</tr>
<tr>
<td>Configuring the stunnel authorizations</td>
<td>40</td>
</tr>
<tr>
<td>Configuring the stunnel command authorizations</td>
<td>40</td>
</tr>
<tr>
<td>Starting and stopping stunnel</td>
<td>41</td>
</tr>
<tr>
<td>6. User Installed Applications</td>
<td>43</td>
</tr>
<tr>
<td>Overview</td>
<td>43</td>
</tr>
<tr>
<td>Application Types</td>
<td>43</td>
</tr>
<tr>
<td>Application Integration: Architecture and Design</td>
<td>44</td>
</tr>
<tr>
<td>Designing the security architecture</td>
<td>45</td>
</tr>
<tr>
<td>Application Implementation and Debugging</td>
<td>48</td>
</tr>
<tr>
<td>7. Post Transition Clean Up</td>
<td>51</td>
</tr>
<tr>
<td>Overview</td>
<td>51</td>
</tr>
<tr>
<td>Part II – Ongoing Administration</td>
<td>53</td>
</tr>
<tr>
<td>8. Compartment Concepts</td>
<td>53</td>
</tr>
<tr>
<td>About Compartments</td>
<td>53</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>About Virtualvault Compartments</td>
<td>54</td>
</tr>
<tr>
<td>About Security Containment Compartments</td>
<td>55</td>
</tr>
<tr>
<td>About Starting Processes in Compartments</td>
<td>59</td>
</tr>
<tr>
<td>Starting a Process in Compartments - Virtualvault</td>
<td>60</td>
</tr>
<tr>
<td>Starting a Process in Compartments - Security Containment</td>
<td>60</td>
</tr>
<tr>
<td>Configuration Examples</td>
<td>60</td>
</tr>
<tr>
<td>Example 1: CGI</td>
<td>60</td>
</tr>
<tr>
<td>Example 2: CGI and Tomcat</td>
<td>62</td>
</tr>
<tr>
<td>9. File Rule Generator</td>
<td>65</td>
</tr>
<tr>
<td>Overview</td>
<td>65</td>
</tr>
<tr>
<td>10. Configuring Role Based Access Control</td>
<td>69</td>
</tr>
<tr>
<td>Basic configuration</td>
<td>69</td>
</tr>
<tr>
<td>VV Transition and user roles</td>
<td>69</td>
</tr>
<tr>
<td>Adding Webadmin role to the a user</td>
<td>69</td>
</tr>
<tr>
<td>Performing tasks with the Webadmin role</td>
<td>69</td>
</tr>
<tr>
<td>Advanced Configuration</td>
<td>70</td>
</tr>
<tr>
<td>Reporting process status in a compartment</td>
<td>70</td>
</tr>
<tr>
<td>Configuring a command to execute in a compartment</td>
<td>70</td>
</tr>
<tr>
<td>Configuring and executing a command in more than one compartment</td>
<td>71</td>
</tr>
<tr>
<td>11. Web Server Transition</td>
<td>73</td>
</tr>
<tr>
<td>Web Server Functionality Overview</td>
<td>73</td>
</tr>
<tr>
<td>Securing an Apache web server instance using Security Containment</td>
<td>74</td>
</tr>
<tr>
<td>Using Role Based Access Control to Start and Stop the Web server Instances</td>
<td>77</td>
</tr>
<tr>
<td>Virtualvault to Security Containment Transition Web Server</td>
<td>80</td>
</tr>
<tr>
<td>Architectures</td>
<td>80</td>
</tr>
<tr>
<td>Apache/TGA on Virtualvault to Apache/CGI on Security Containment</td>
<td>80</td>
</tr>
<tr>
<td>Apache/Webproxy on Virtualvault to Apache/Webproxy on Security Containment</td>
<td>83</td>
</tr>
<tr>
<td>Apache/Tomcat on Virtualvault to Apache/Tomcat on Security Containment</td>
<td>84</td>
</tr>
<tr>
<td>Post Transition Web Server Administration</td>
<td>86</td>
</tr>
<tr>
<td>Creating a Secure Web Server Instance</td>
<td>87</td>
</tr>
<tr>
<td>Removing a Secure Web Server Instance</td>
<td>90</td>
</tr>
<tr>
<td>Undoing the Actions of vvscassist.sh</td>
<td>90</td>
</tr>
<tr>
<td>Beyond Virtualvault Web Server Security</td>
<td>91</td>
</tr>
</tbody>
</table>
## Table of Contents

Welcome to the Virtualvault Transition

Part III – Appendices 93

Appendix A: About Auditing 93

  Overview 93

  Correspondence between Virtualvault Audit Groups and Security Containment system calls 94

Appendix B: HPUXHarden Configuration 96

Appendix C: VVSnapshot Configuration 103

Appendix D: VVSCAssist Configuration 113

Appendix E: VVSCAssist TODO 125

Appendix F: Complete HP-UX Software Bundles List 134

Appendix G: Disk Space Requirement Template 139

  Space for user installed applications 139
  Space for web server chroot instances 139
    Directory locations 139
    Finding chroot web server directories 140
    Determining number of web server chroot instances to be created 141
    Calculating required space for web server chroot instances 142

Appendix H: Updating Completed Transitions 143

  Transitions completed prior to the March 15, 2006 Release 144
    IPFilter Related Changes 144
    Client Certificate Related Changes 146
  Transitions completed prior to the April 17, 2006 Release 146
Welcome to the Virtualvault Transition

Introduction

Purpose

The purpose of this document is to guide a Virtualvault administrator through the transition from Virtualvault to HP-UX with Security Containment. Before executing any steps in this guide, it is recommended that the document be fully reviewed and understood by the reader. To ensure a successful transition, an understanding of the entire process presented in this document is essential.

Audience & Assumptions

This document has been written for use by current Virtualvault system administrators preparing to transition to a Security Containment system.

Although Virtualvault and Security Containment offer similar functionality, the mechanisms used differ greatly. HP recommends that the person carrying out the transition familiarize themselves with the mechanisms and procedures for administering a Security Containment system. HP offers training and documentation on the concepts and administration of Security Containment.

Note: The transition process is not designed to recreate all settings and configurations that may be present on the Virtualvault system. The process creates a Security Containment framework similar to that on VV. The target system may require additional manual configuration. Please review the VVSCAssist configuration file, documented in Appendix D: VVSCAssist Configuration for a complete listing of settings and configurations that are set up on the Security Containment system for each subsystem.

Background

Virtualvault has provided a high level of security for companies around the globe. In 2003, Hewlett-Packard announced the end of life of the Virtualvault product family. In parallel to this announcement, Hewlett-Packard began incorporation of additional security mechanisms into the base HP-UX operating system. These security mechanisms represent the next generation of security technology being offered as part of the base HP-UX operating system. The base HP-UX platform with the security extensions is referred to as Security Containment.
The new security functionality found in Security Containment can be configured to enforce a security policy similar to that of the Virtualvault product. HP is providing a transition toolkit to assist customers in transitioning existing Virtualvault systems to Security Containment systems.

HP-UX is a general purpose operating system. In contrast, Virtualvault is a special version of HP-UX, tailored to serve as a secure front-end web server. The transition toolkit targets configuring the Security Containment system as a secure web server and implements appropriate configuration changes to deliver a high degree of security for a web server environment.

This document is divided into three major sections. The purpose of each section is as follows:

- **Part I:**
  This part describes the step by step procedures to transition from a Virtualvault to a Security Containment system. This includes procedures to be performed on the Virtualvault system to capture relevant information, and procedures to install and configure the Security Containment system. The procedures described in this part have been targeted to meet the security needs of most customers and have been automated to the extent possible. For customers that wish to obtain a deeper understanding of the automated procedures, descriptions can be found in the appendices in Part III. Note: This part describes a one-time procedure to transition from Virtualvault to Security Containment.

- **Part II:**
  This part describes administrative procedures that may be performed on an ongoing basis. For instance, the addition of web servers or the addition of compartments is described in this part.

- **Part III:**
  This part contains appendices, which provide sample reports and configuration file listings. This information is provided for those customers that desire more detailed information on the transition.

**Transition Overview**

As described in the previous paragraphs, Part I of this document provides step-by-step instructions for transitioning a Virtualvault system to a Security Containment system. The transition process requires two separate systems, one running Virtualvault and a separate system for Security Containment. The transition process does not support upgrading an existing Virtualvault system to Security Containment. Execution of these processes will result in the following:
• Information is collected in an archive on the existing Virtualvault platform. The administrator moves the archive to the Security Containment system, after initial installation. This archive includes various files and configuration information. Some of this information is used during automated and manual procedures to set up the Security Containment system. Additionally, some of this information is collected to serve as a source of reference for the Security Containment system administrator.

• Setting up the Security Containment system
  o The HP-UX 11i v2 operating system is installed on the target platform.
  o The VV transition toolkit is used to configure the system in a secure state (i.e., system hardening).
  o The Security Containment bundle is installed
  o The archive information collected on the Virtualvault is transferred to the Security Containment system
  o The transition toolkit configures the Security Containment mechanisms such that the security policy is similar to a Virtualvault system.
  o Additional configuration may be required for user installed applications

• Security Containment system and applications are tested to ensure proper behavior.

**User Defined Applications**

The work to achieve a security policy similar to Virtualvault on Security Containment will be handled by the transition toolkit. The toolkit also addresses certain well known aspects of applications that are deployed on Virtualvault (e.g., CGI scripts, Apache web server). However, not all scenarios of user installed code (whether developed in house or by third parties) can be covered due to the great number of variables. For instance, the toolkit cannot determine all of the files that make up a user defined application.

Therefore, it is important to review and understand how the user defined applications currently running on the Virtualvault platform are configured and how they operate. Additional actions (e.g., moving over additional files or setting additional configuration parameters) may be required to enable a user defined application on a Security Containment system.

**Unsupported Functionality**

HP has attempted to make available all Virtualvault functionality that is required by our customers on Security Containment. However, some functionality will not be available on Security Containment.
Welcome to the Virtualvault Transition

- **File System Integrity** – Security Containment does not contain a file system integrity mechanism similar to Virtualvault.

  The third party tool Tripwire may potentially be used to provide similar functionality. Tripwire may be used to monitor file attributes such as file owner, group, permissions and modification of the file contents. Tripwire cannot address privileges associated with a file or integrity of rules that determine Mandatory Access Control for a file.

- **Retire Root** – This functionality was limited on Virtualvault and an improved implementation may be included in a future HP-UX release.

**Virtualvault Transition Support**

Support issues related to the Virtualvault Transition should be directed to: issl-support@issl.atl.hp.com

Issues specific to Virtualvault and/or Security Containment should be handled using standard support procedures.

**Identifying the Transition Steps**

HP has automated many of the tasks required for the transition from Virtualvault to HP-UX with Security Containment.

**Note:**

- There are tasks which the user must perform manually; this includes the testing and verification of the installed applications as well as the verification of the security integrity before the machine is placed into a production environment.

The following table describes the major activities accomplished in each step of the transition process. Note that there are steps performed on the Virtualvault system and steps performed on the Security Containment system. The transition process requires two separate systems, one running Virtualvault and a separate system for Security Containment. The transition process does not support upgrading an existing Virtualvault system to Security Containment. Each step number relates to a section within Part I of this guide.
<table>
<thead>
<tr>
<th>Step</th>
<th>Activities</th>
</tr>
</thead>
</table>
| 1    | • Perform a complete system backup  
      • Install transition software depot  
      • Export Virtualvault configuration settings and files  
      • Perform backup of VVSnapshot archive |
| 2    | • Install HP-UX 11iv2  
      • Download the additional required software depots  
      • Configure HP-UX Secure Shell (SSH)  
      • Execute the hardening script  
      • Complete identified manual hardening tasks  
      • Enable compartments  
      • Create a System Recovery Tape |
| 3    | Import Virtualvault configuration settings and files. |
| 4    | Integrate applications |
| 5    | Test and verify functionality |
Part I – The Transition Process

1. Preparing to Move from Virtualvault to Security Containment

Overview

This chapter details the tasks performed on the Virtualvault system to capture configuration and data files. That information is later used in configuring the new Security Containment system.

Getting Started

This process is non intrusive and will not modify any Virtualvault files. All commands must be executed as root in the Syslo compartment. The utilities will scan the different subsystems within Virtualvault and create a separate staging of the files which need to be transferred to the Security Containment system. However, it is best practice to ensure that a current system backup exists. A complete backup should include:

- Data files
- Bootable backup tape (bbt)

Export Virtualvault Configuration

What is VVSnapshot?

VVSnapshot is a utility that captures Virtualvault setup and configuration files which are used in the configuration of the target Security Containment system.

In general, VVSnapshot performs the following:

- Scans functional areas of the system
- Writes configuration details to the VVSnapshot Report file
Preparing to Move from Virtualvault to Security Containment

- Copies configuration files and directories to the Staging Directory
- Creates the Staging Archive to be transferred to the target Security Containment system

**What are the functional areas?**

VVSnapshot scans the following functional areas for configuration details:

- Boot Time
- Kernel Tunables
- Security Databases
- Users and Groups
- SSH
- Filesystem
- Network
- Audit Events
- Cron Files
- Openview VPO Client
- Web Server
- Web Proxy
- Java Servlet
- TGA
- TGP
- Trusted IPC
- Chroot

**VVSnapshot Files**

**Install Directory**

The VVSnapshot utilities are installed in the directory /opt/hpvv/vvsnapshot/.

**Script File - vvsnapshot.sh**

vvsnapshot.sh is the script for capturing Virtualvault setup and configuration files.

**Configuration File - vvsnapshot.config**

The VVSnapshot configuration file specifies which modules will be executed by the VVSnapshot script. The default values within the configuration file are the HP recommended. See Appendix C: VVSnapshot Configuration on page 103 for more information.

**Log File - vvsnapshot.log**

All VVSnapshot input and output (stdin, stdout, stderr) is recorded in the VVSnapshot log file.
1. Preparing to Move from Virtualvault to Security Containment

The report file contains configuration details of the functional areas scanned on the Virtualvault system. It is included in the Staging Archive that is transferred to the Security Containment system. It is used along with the Staging Directory as input for VVSCAssist, a program which will be run on the Security Containment system.

Configuration files and directories of the areas scanned are copied to the staging directory.

The Staging Archive is a compilation of all configuration and data files discovered on the Virtualvault by the VVSnapshot utility. This archive is used in Chapter 3. Importing Virtualvault Data and Configuration.

**Downloading and Installing the VVSnapshot Depot**

The VVSnapshot depot contains the utilities needed to transition from Virtualvault to the target Security Containment system.

For security reasons, a Virtualvault server, by default, does not allow access to an outside network for administrative operations. Therefore, you must download the software component files from the Internet to a host on the internal network, and then move the downloaded files over the internal network onto the Virtualvault server.

To download the VVSnapshot depot follow these steps. These steps assume that your download host is not a Virtualvault machine for the security reasons described above.

2. Download the file: VVSnapshot.depot
3. Optionally, download the file: Readme.txt
   It contains a description of the depot and its cksum(1M) value to verify the integrity of the downloaded file.
4. Log onto the Virtualvault server as root at the Syslo sensitivity level.
5. Create a download directory (for example, /download) and change to it. Use the ftp command to copy the file from the download host onto the Virtualvault server. Change the sensitivity level of the copied file to system. For example:
   
   # chlevel syslo /download/VVSnapshot.depot
6. Install the VVSnapshot depot using the swinstall program
1. Preparing to Move from Virtualvault to Security Containment

Executing the VVSnapshot Script

Before running the VVSnapshot script, it is recommended that the configuration file be reviewed. See Appendix C: VVSnapshot Configuration on page 103 for more information.

Execute the VVSnapshot script as root in the Syslo compartment by typing the following commands:

```
# /opt/hpvv/vvsnapshot/vvsnapshot.sh -v
```

User Prompts

During execution, VVSnapshot will prompt for information.

When prompted to change the default Staging Directory, enter n to retain the default value (`/opt/hpvv/vvsnapshot_staging/`) or y to change the default staging location.

When prompted to change the default Staging Archive, enter n to retain the default value (`/opt/hpvv/vvsnapshot_staging.tar`) or y to change the default Staging Archive location.

Note:

- Entering y will give the option to save the Staging Archive to a tape device or to an alternate disk location.

At the end of a successful execution, an exit banner is displayed.

```
#-----------------------------
## Exit vvsnapshot.sh
#-----------------------------
```

Troubleshooting the VVSnapshot Script

If the VVsnapshot script encounters an error it will stop execution with the message,

```
Script terminating prematurely! Please resolve the problem(s) above and re-run this script.
```

To troubleshoot the VVSnapshot script, review the log file `/opt/hpvv/vvsnapshot/vvsnapshot.log.`
2. Installing the Security Containment System

Overview

The following chapter describes procedures for a new installation of HP-UX Security Containment system. These steps include:

- Installing the HP-UX 11iv2 Mission Critical Operating Environment
- Hardening the system
- Installing and configuring Security Containment.

Installing HP-UX 11iv2

To perform a new installation of the HP-UX Security Containment system for the Virtualvault transition, follow these procedures.

**Note:** A fresh installation of the operating system is required for a successful transition.

- Boot from the *HP-UX 11i Version 2 Mission Critical Operating Environment March 2006 DVD 1 of 2 (p/n 5013-4906)*:
  - Select **Install HP-UX**
  - Select **Advanced Installation**
  - From the **Basic** tab:
    - Select **Configurations**: HP-UX B.11.23 Default
    - Select **Environments**: Foundation OE-64-bit
    - Select target disk
From the **Software** Tab change the following bundles as indicated in the table below. Entries not listed in this table should not be modified:

<table>
<thead>
<tr>
<th>Marked?</th>
<th>Bundle</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>B5725AA</td>
<td>HP-UX Installation Utilities (Ignite-UX)</td>
</tr>
<tr>
<td>No</td>
<td>B6848BA</td>
<td>Ximian GNOME 1.4 GTK+ Libraries for HP-UX</td>
</tr>
<tr>
<td>No</td>
<td>DSAUtilities</td>
<td>HP-UX Distributed Systems Administration Utilities</td>
</tr>
<tr>
<td>Yes</td>
<td>HPUX-HIDS</td>
<td>HP-UX Host IDS E.04.00</td>
</tr>
<tr>
<td>No</td>
<td>hpuxwsWebmin</td>
<td>HP-UX Webmin-based Admin</td>
</tr>
<tr>
<td>No</td>
<td>ISEEPlatform</td>
<td>ISEE Platform</td>
</tr>
<tr>
<td>No</td>
<td>J4269AA</td>
<td>LDAP-UX Integration</td>
</tr>
<tr>
<td>Opt</td>
<td>Judy</td>
<td>Judy Library and Related files</td>
</tr>
<tr>
<td>Opt^{1}</td>
<td>MOZILLA</td>
<td>Mozilla for HP-UX</td>
</tr>
<tr>
<td>No</td>
<td>MOZILLAsrc</td>
<td>Mozilla Source distribution</td>
</tr>
<tr>
<td>No</td>
<td>ParMgr</td>
<td>Partition Manager - HP-UX</td>
</tr>
<tr>
<td>No</td>
<td>SysMgmtWeb</td>
<td>HP-UX Web Based System Management User Interfaces</td>
</tr>
<tr>
<td>No</td>
<td>T2743AA</td>
<td>HP Global Workload Manager Agent</td>
</tr>
</tbody>
</table>
2. Installing the Security Containment System

<table>
<thead>
<tr>
<th>No</th>
<th>UtilProvider</th>
<th>HP-UX Utilization Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VMGuestLib</td>
<td>Integrity VM Guest Support Libraries</td>
</tr>
<tr>
<td></td>
<td>VMProvider</td>
<td>WBEM Provider for Integrity VM</td>
</tr>
<tr>
<td></td>
<td>vParProvider</td>
<td>vPar Provider - HP-UX</td>
</tr>
</tbody>
</table>

1. Mozilla 1.4 may be optionally installed if a web browser client is desired on the system.

For a complete listing of bundle markings, see Appendix F.

- From the **System** Tab, enter specific configuration information, such as your hostname and networking information.
- From the **File System** Tab, enter customer specific configuration information, such as partition information. For most installations, the following amounts of additional space for the transition toolkit are sufficient:
  - `/var`: an additional 2 GB. To get a site-specific estimate for the space needed by web server chroot instances, see Appendix G.
  - `/opt`: add the size of the VVSnapshot Staging Directory (normally `/opt/hpvv/vvsnapshot_staging`) on the Virtualvault system. If the Staging Archive was saved to disk (as `vvsnapshot_staging.tar`), add the size of that file.
- Select **Go**.
- Mount the *HP-UX 11iv2 Mission Critical OE DVD 2 of 2* when requested.

### Required Software Depots

Log in as root and download, but **do not install** the depots listed below. This is required because network access will be restricted later in the process.

**HPUXHarden depot**

This depot contains the HPUXHarden tool, which will configure the HP-UX 11iv2 operating environment to make it more secure.
2. Installing the Security Containment System

VVTransition depot

This depot contains the transition tools which will assist in configuring the Security Containment system to resemble the Virtualvault system.

Downloading Software Depots

To download the three required depots follow these steps.

1. Log onto the Security Containment system as root
2. Create a download directory (for example, /download) and change to it.
3. From your download host, access Hewlett-Packard’s ftp site at:
4. Download the file: VVTransition.depot
5. Download the file: Readme.txt This file contains a description of the depots, version history and their cksum(1M) value to verify the integrity of the downloaded file.
6. From your download host, access Hewlett-Packard's ftp site at:
7. Download the file: HPUXHarden.depot

Using HP-UX Secure Shell (SSH)

After completing the next step, hardening the system, the telnet service will be disabled. In order to get an interactive shell prompt from the hardened system, it is necessary to use SSH or a console (a terminal or a remote console).

At this point, SSH has been installed and configured in the Security Containment system. To use SSH, access the Security Containment system using an SSH client. Accept the new server key when prompted by the client.

Hardening the Security Containment System

What is HPUXHarden? HPUXHarden is a utility that modifies system configuration to make the Security Containment system more secure.
In general, HPUXHarden does the following:

- Initially hardens the system using Bastille
- Further hardens the system by modifying configuration settings and file permissions, disabling services, and removing filesets
- Creates a TODO list of additional hardening tasks to be manually performed

**HPUXHarden Files**

**Install Directory**

The HPUXHarden files can be found in the directory `/var/opt/hpuxharden/`.

**Configuration File - hpuxharden.config**

The configuration file is used to specify which modules are executed by the HPUXHarden script. The default values within the configuration file are the HP recommended settings. For a complete listing of the default configuration file, see Appendix B: HPUXHarden Configuration on page 96 for more information.

HPUXHarden performs the following tasks:

- Strengthen Bastille lockdown configuration
- Remove global chown privileges
- Remove unneeded pseudo-accounts
- Change root home directory to `/root`
- Configure `nsswitch.conf(4)` policy
- Prevent syslogd from listening on the network
- Disable rpcbind daemon
- Disable console logging
- Disable pwgrd daemon
- Examine Set-id Programs
- Examine File Permissions
- Remove Software Filesets

HPUXHarden modifies the system configuration to make the Security Containment system more secure.

All HPUXHarden input and output (`stdin, stdout, stderr`) is recorded in the HPUXHarden Log File.

Three sections of the hardening script potentially alter file permissions using the `chown(1)` and `chmod(1)` commands. Prior to altering any permission, the pre-hardened values are captured in the `prehardenperms.log` file.
Installing the Security Containment System

If it is determined at a later time that the original permissions are needed, refer to the prehardenperms.log file for the original permission values.

HPUXHarden removes users and groups, and saves a copy of the original database files using the following naming conventions:

- /etc/passwd.preharden_yyyymmdd:hhmmss
- /etc/group.preharden_yyyymmdd:hhmmss

Bastille creates the following log files during execution. See the Bastille documentation in /opt/sec_mgmt/bastille/docs/user_guide.txt for information about these files:

- /var/opt/sec_mgmt/bastille/TODO.txt
- /var/opt/sec_mgmt/bastille/log/action-log
- /var/opt/sec_mgmt/bastille/log/error-log

swremove(1M) creates the following log files during execution. These log files provide information on the filesets which were removed by the “Remove Filesets” HPUXHarden module.

- /var/adm/sw/swremove.log
- /var/adm/sw/swagent.log

Installing the HPUXHarden Depot

Install the HPUXHarden depot as root using the following swinstall(1M) command:

```
# swinstall -s /download/HPUXHarden.depot
```

Executing the HPUXHarden Script

Before running the HPUXHarden script, it is recommended that the script actions be reviewed. See Appendix B: HPUXHarden Configuration on page 96 for more information.

Execute the HPUXHarden script as root with the following command:

```
# /var/opt/hpuxharden/hpuxharden.sh -v
```

Note:

- Upon successful execution of the script, the system will reboot.

User Prompts

During execution, HPUXHarden will prompt for information.
2. Installing the Security Containment System

When prompted to “Remove Software Filesets”, enter y to continue.

When prompted to “Run Security Patch Check”, enter y to continue.

If Security Patch Check is run:

- When prompted “does the machine require a proxy to ftp to the Internet?”, enter y to enter the proxy information or n if a proxy is not required
- When prompted to “confirm Bastille Security Patch Check configuration:” enter y to confirm or n to change the configuration.

At the end of a successful execution, an exit banner is displayed

##### Exit hpuxharden.sh
NOTE: Script execution details are recorded in logfile:
/var/opt/hpuxharden/hpuxharden.sh.log

After completion of the HPUXHarden script, the log file
/var/opt/hpuxharden/hpuxharden.sh.log should be reviewed.

Troubleshooting the HPUXHarden Script

If the HPUXHarden script encounters an error it will stop execution with the message,

##### Abort <module name> #####
Script terminating prematurely! Please resolve the problem(s) above and re-run this script.
##### Exit basename <filename> <date> #####

To troubleshoot the HPUXHarden script review the log file
/var/opt/hpuxharden/hpuxharden.sh.log.

Manual Steps to Complete the Hardening Process

There are manual configuration steps which are required to complete the hardening of the system. These steps are listed in the Bastille TODO file. Examine this file and follow the recommendations. The TODO list could contain sections such as “Writable Directory Audits”, “Install Security Patches”, and “IP Filter Configuration”.

The Bastille TODO file is located at:
Setting up Security Containment

The following sections detail the steps necessary to install, configure and verify Security Containment compartment functionality.

Installing Security Containment

The Security Containment depot is located on the HP-UX 11i v2 March 2006 Software Pack DVD. To install Security Containment, use the following commands:

```
# mkdir /dvdrom
# ioscan -fnC disk
# mount -r /dev/dsk/<dvdrom> /dvdrom
# swinstall -x autoreboot=true -s /dvdrom SecurityExt
```

Notes:

- The installation of the Security Containment depot will require a reboot of the system.
- Due to changes made during the Hardening process, running `swverify SecurityExt` will produce error messages related to file permissions. This is expected behavior.

Enabling Security Containment Compartments

Use the following steps to enable Security Containment compartments.

1. Type the following command line

   ```
   # /usr/sbin/cmpt_tune -e
   ```

   `cmpt_tune` may issue a message warning about the creation of the `ifaces.rules` file containing a compartment for the network interfaces.

2. Reboot the system to complete the activation process

   ```
   # /usr/sbin/reboot
   ```

Installing the VVTransition Depot

The VVTransition depot contains the Virtualvault to Security Containment transition tools. The transition tools configure a target
HPUX 11iv2 system to resemble a Virtualvault system security policy.

Install the VVTransition depot by typing the following `swinstall(1M)` command.

```
# swinstall -x autoreboot=true -s /download/VVTransition.depot #
```

**Note:**
- The installation of the VVTransition depot will require a reboot of the system.

**Verify Security Containment LAN Setup**

The VVTransition depot installation configured the first two network interface cards (NIC’s), to resemble the Virtualvault inside and outside LAN setup.

**Warning:**
- This NIC configuration may not be the correct setup for your environment.

The NIC connecting to the Internet should be assigned to the outside interface compartment, referred to as `out_iface`. The NIC connecting to the intranet should be assigned to the inside interface compartment, referred to as `in_iface`.

**Verify Outside LAN**

Verify that the LAN number is correct in the outside rules file `/etc/cmpt/out_iface.rules`:

```plaintext
compartment out_iface { 
   interface lan0
 }
```

**Verify Inside LAN**

Verify that the LAN number is correct in the inside rules file `/etc/cmpt/in_iface.rules`:

```plaintext
compartment in_iface { 
   interface lan1
 }
```

**Create System Recovery Tape**

It is strongly recommended that a system recovery tape be created at this point in the transition process. Should a mistake be made
after this point, do not try to reverse the steps performed. Instead, restore the latest backup and restart the process from here. It is recommended that a different tape be used for each backup.

Refer to your backup procedures to create a full system backup. For reference documentation, go to http://docs.hp.com and use the search function. For example, search for a document called “Backing up your data using HP-UX commands”. As an example the following command can be used to create a backup tape:

```
# /opt/ignite/bin/make_tape_recovery -AI
```
3. Importing Virtualvault Data and Configuration Files

Overview

This section details how to use the VVSCAssist script to import the data collected from the Virtualvault system.

Import Virtualvault Configuration

What is VVSCAssist?

VVSCAssist is a utility which configures the Security Containment system to resemble a Virtualvault system. All the instructions in this chapter must be performed as root.

In general, VVSCAssist does the following:

- Extracts the VVSnapshot Report File and Staging Directory from the VVSnapshot Staging Archive
- Configures the Security Containment system from the VVSnapshot data
- Creates a TODO list of manual tasks

Transferring the VVSnapshot Staging Archive to the Security Containment System

The VVSnapshot staging archive must be copied from the Virtualvault system to the Security Containment system.

If the Staging Archive was written to tape, insert the tape on the Security Containment system for use by the VVSCAssist script.

If the Staging Archive was written to a file, use ftp to transfer vvsnapshot_staging.tar to the directory /opt/hpvv on the Security Containment system.

If the archive file is not located in the /opt/hpvv directory, the VVSCAssist script will search the filesystem looking for a file named vvsnapshot_staging.tar. If the search is unsuccessful, it will prompt to retrieve the archive from a tape device.
VVSCAssist Files

The VVSCAssist files can be found in /opt/hpvv/.

**Install Directory**

**Script File - vvscassist.sh**

vvscassist.sh extracts the configuration and data files from the Staging Archive and uses them to configure the Security Containment system.

**Configuration File - vvscassist.config**

The configuration file is used to specify which modules will be executed by the VVSCAssist script. The default values within the configuration file are the HP recommended settings.

VVSCAssist configures the following functional areas:

- Boot Time
- Kernel Tunables
- Security Databases
- Users and Groups
- SSH
- Filesystem
- Network
- Audit Files
- Cron Files
- Openview VPO
- CGI
- Java Servlet
- Webproxy
- Web Server
- Chroot
- TGP
- Trusted IPC

**Note:**

- The CGI functional area is configured by VVSCAssist using the TGA configuration captured by VVSnapshot on the Virtualvault system.

All VVSCAssist input and output (stdin, stdout, stderr) is recorded in the VVSCAssist Log File.

**Web server creation Log File - vvws_createxsec.log**

VVSCAssist uses the vvws_createxsec script to create web servers. All VVSCAssist interaction with this script is recorded in the vvws_createxsec log file.


**TODO File - vvscassist.todo**

The TODO file is created by the VVSCAssist script and is used as a guide to further configure the Security Containment system.
Executing the VVSCAssist Script

Before running the VVSCAssist script, it is recommended to review what the script will do. See Appendix D: VVSCAssist Configuration on page 113 for more information.

Execute the VVSCAssist script with the following command:

```
# /opt/hpvv/vvscassist/vvscassist.sh -v
```

Upon execution, VVSCAssist will prompt for information.

When prompted:
Checking for Staging Archive...
Found /opt/hpvv/vvsnapshot_staging.tar
Use this archive? [y]:
Enter y to continue.

When prompted:
Default Staging Directory:
/opt/hpvv/vvsnapshot_staging/
Accept the default location? [y]:
Press enter to accept the default on enter n to change the staging area.

At the end of a successful execution, an exit banner is displayed

```
#####################
## Exit vvscassist.sh
#####################
```

Troubleshooting the VVSCAssist Script

If the VVSCAssist script encounters an error it will stop execution with the message,

```
##############################
## Abort (220) vvscassist.sh [DATE/TIME]
##############################
```

To troubleshoot the VVSCAssist script, review the following log files:

```
/opt/hpvv/vvscassist/vvscassist.log
/opt/hpvv/vvscassist/vvws_createxsec.log
```

For information about undoing the actions taken by VVSCAssist script, refer to the section, Undoing the Actions of vvscassist.sh, on page 90.
3. Importing Virtualvault Data and Configuration Files

Manual Steps to Complete the Import Process

There are manual configuration steps which are required to complete the configuration of the Security Containment system. The manual steps are listed in the VVSCAssist TODO file. Examine the file and follow the recommendations.

/opt/hpvv/vvscassist/vvscassist.todo

See Appendix E: VVSCAssist TODO on page 125 for a sample VVSCAssist TODO list.

Create System Recovery Tape

It is strongly recommended that a system recovery tape be created at this point in the transition process. Should a mistake be made after this point, do not try to reverse the steps performed, instead, restore the latest backup and restart the process from here. It is recommended that a different tape be used for each backup.

Refer to your backup procedures to create a full system backup. For reference documentation, go to http://docs.hp.com and use the search function. For example, search for a document called “Backing up your data using HP-UX commands”. As an example the following command can be used to create a backup tape:

```
# /opt/ignite/bin/make_tape_recovery -AI
```
4. Configuring HP-UX HIDS for Alarms functionality

Introduction

On Virtualvault, the Alarm subsystem is a flexible mechanism for transforming the audit data from a passive to an active security feature. It allows the administrator to specify the event to be monitored, when to monitor it, and what frequency and severity should trigger the alarm.

On Security Containment, HP-UX Host Intrusion Detection System (HIDS) provides a fixed set of templates to monitor for specific attack patterns. HP-UX HIDS uses sources of data that include HP-UX audit events as well as system logs.

Both Virtualvault and Security Containment systems allow the administrator to specify notification and response actions. Notification actions include Openview notifications in both systems. Response actions can use system commands as well as user programs.

For Virtualvault customers transitioning to Security Containment, HP-UX HIDS is an adequate replacement of Virtualvault alarms in most cases. The most common purpose of alarms is to detect potential attacks and react accordingly.

When HP-UX HIDS detects an intrusion attempt, it issues an alert, so the administrator can immediately investigate the situation, and when necessary, take action against the intrusion. In addition, the administrator can customize a local response to an alert.

HP-UX HIDS provides several detection templates that can be used to monitor specific event patterns. The detection templates are included in the package and are not created by the administrator. The administrator can specify the response action.

For more information on HP-UX HIDS refer to the *HP-UX Host Intrusion Detection System Administrator's Guide.*

In a system monitored by HP-UX HIDS, there is an agent process constantly monitoring the system. When the agent detects an event,
it sends a notification to a server process called the System Manager.

The steps detailed below configure the HP-UX HIDS agent process in its own Inside compartment. The HP-UX HIDS System Manager is run on a separate host to monitor the Security Containment system.

**Lockdown of HPUX-HIDS using Security Containment**

HP-UX HIDS on Security Containment is isolated in its own compartment in order to limit its vulnerability. By limiting its write access to the filesystem, the probability of an attack that requires the creation of certain temporary files is reduced.

Network access restriction is a different matter. HP-UX HIDS requires cross-compartmental IPC access to gather information about processes, such as process name, arguments, etc. This information is included in the notification sent to the System Manager to be read and analyzed by the system administrator.

The privilege COMMALLOWED is granted to the `idsagent` process to allow the `pstat_getproc(2m)` calls to access process tables in other compartments. COMMALLOWED allows a process to override compartment rules in the IPC and networking subsystems.

The consequence of granting this privilege to the HP-UX HIDS agent process is that it will have cross-compartment network access. Since the agent is configured to communicate to the Intranet, in most cases this is not a security concern. In addition, the HP-UX HIDS compartment rules allow communication only to and from the predefined HP-UX HIDS ports.

**Configuring HP-UX HIDS**

After following the installation instructions earlier in this document, HIDS version 4.00 is installed in the system. Login in as root in the INIT compartment of the Security Containment system.

Follow the instructions in the *HP-UX Host Intrusion Detection System Administrator's Guide* to configure HP-UX HIDS. Specifically, make sure the following actions are performed according to the instructions in the manual:
4. Configuring HP-UX HIDS for Alarms functionality

- Configure the `IDS_LISTEN_IFACE` parameter to listen on the Intranet NIC.
- Set up the HP-UX HIDS secure communications.
- Set up IPFilter rules.

Creating HP-UX HIDS compartment

Create a new in_ids compartment by creating a file

```plaintext
/etc/cmpt/in_ids.rules with the following content:
```

```plaintext
/******************************
compartment in_ids {
    /* Inside file system rules */
    #include </etc/cmpt/inside.common>

    /* ids file system rules */
    perm  read, write, create, unlink /var/opt/ids
    perm  read, write, create, unlink /var/spool/cron/crontabs
    perm  read, write, create, unlink /var/adm/cron/FIFO

    /* ids IPC rules */
    grant client tcp peer port 2984 in_iface /* To HP-UX HIDS Admin */
    grant server tcp port 2985 in_iface      /* Local HP-UX HIDS Agent */
}
/******************************
```

Run `/usr/sbin/setrules` to make the new compartment effective. The warning about the INIT compartment being redefined is expected and requires no action.

Configuring RBAC for HP-UX HIDS

In order to have the HP-UX HIDS agent process operate in its own compartment, RBAC must be configured. A new role, authorization and command authorization are created and associated with each other to allow three basic operations:

- Start the HP-UX HIDS agent
- List the processes in the HP-UX HIDS compartment
- Stop the HP-UX HIDS agent.

Since the HP-UX HIDS agent must be started by the user `ids`, the new role is associated to that user.

All the commands listed below are executed as root in the INIT compartment unless otherwise specified.

Configuring the HP-UX HIDS user role

Use the following command to create the HP-UX HIDS role:

```
# roleadm add vv-idsagent
```
roleadm: added role vv-idsagent

Assign the new role to user ids:

```
roleadm assign ids vv-idsagent
roleadm assign done in /etc/rbac/user_role
```

### Configuring the HP-UX HIDS authorizations

The authorizations `vv-ids.agent.start` and `vv-ids.agent.stop` are created to start and stop the agent. The authorization `vv-ids.agent.ps` is created to allow the system administrator to verify that the idsagent process and its children are running:

```
authadm add vv-ids.agent.start
authadm added auth: (vv-ids.agent.start,*)
authadm add vv-ids.agent.stop
authadm added auth: (vv-ids.agent.stop,*)
authadm add vv-ids.agent.ps
authadm added auth: (vv-ids.agent.ps,*)
```

Associate the newly created authorizations with the HP-UX HIDS user role:

```
authadm assign vv-idsagent vv-ids.agent.start
authadm added 1 auth for role vv-idsagent
authadm assign vv-idsagent vv-ids.agent.stop
authadm added 1 auth for role vv-idsagent
authadm assign vv-idsagent vv-ids.agent.ps
authadm added 1 auth for role vv-idsagent
```

### Configuring the HP-UX HIDS command authorizations

Add entries to the command authorizations table for each of the commands to be used, as shown below. Note that the UID and not the username must be used. In this example, the user `ids` has UID 101:

```
# cmdprivadm add cmd=/opt/ids/bin/idsagent op=vv-ids.agent.start ruid=101\ rgid=101 euid=101 egid=101 compartment=in_ids privs=COMMALLOWED,LIMIT
/opt/ids/bin/idsagent:dflt:\
(vv-ids.agent.start,*):101/101/101/101:in_ids:COMMALLOWED:dflt:
cmdprivadm added the entry to /etc/rbac/cmd_priv

# cmdprivadm add cmd=/usr/bin/kill op=vv-ids.agent.stop ruid=101 rgid=101 \ euid=101 egid=101 compartment=in_ids
/usr/bin/kill: (vv-ids.agent.stop,*):101/101/101/101:in_ids::
cmdprivadm added the entry to /etc/rbac/cmd_priv
```
Starting and Stopping the HP-UX HIDS Agent

To start the HP-UX HIDS agent, as user ids, use the command:

```bash
$ privrun /opt/ids/bin/idsagent [options]
```

See the *HP-UX Host Intrusion Detection System Administrator's Guide* for details about the options for the `idsagent` command.

To stop the agent, as user ids, use the command:

```bash
$ privrun kill -TERM $(cat /var/opt/ids/idsagent.pid)
```

To list the processes running in the *in_ids* compartment, as user *ids*, use the command:

```bash
$ privrun ps -ef
```
5. Configuring stunnel for TGP functionality

Introduction

Customers transitioning from Virtualvault to Security Containment face a new set of features to achieve a similar or higher level of security than provided by Virtualvault. In the case of transitioning TGP configurations, configurable compartment rules are essential. By using compartment rules to allow traffic in a particular direction between compartments, administrators do not need to grant broad privileges to an application. In Virtualvault, granting a privilege to cross compartment boundaries allowed the application to cross any compartment boundary. With Security Containment compartment rules, the application can be restricted to communicate to a specific compartment, on a specific port, in a specific direction (client or server, or both).

Security Containment rules can be used to allow the data flow from the Internet client to the Intranet server through the user application, however, to get security equivalent to that of a TGP configuration, three key features are required:

- **Two hop connection between Internet and Intranet.** Having a second hop avoids the need to have the application in a single compartment with access to both Internet and Intranet.
- **Port mapping.** The TGP can listen on a port in one side and forward it to a different port on the other side.
- **SSL encryption.** The TGP can receive and send SSL encrypted data.

Since Security Containment does not include tools to achieve such functionality, a third party utility must be used.

stunnel is a utility provided by Hewlett-Packard as part of the hpuxiexpress package available from http://software.hp.com. Stunnel provides the functionality that complements Security Containment to replace the Virtualvault TGP functionality.

From the stunnel manual page:

*The stunnel program is designed to work as SSL encryption wrapper between remote clients and local (inetd-startable) or*
remote servers. The concept is that having non-SSL aware daemons running on your system you can easily set them up to communicate with clients over secure SSL channels.

stunnel can work by either:

- Receiving unencrypted data and sending it to an SSL server
- Receiving encrypted data and
  - Sending the decrypted data to an arbitrary port on that or another machine
  - Launching a local program (as does inetd) to talk to the remote machine over the encrypted channel.

In addition, stunnel allows port mapping, where the client connects to a port number that is different than the port that Stunnel is forwarding the data to.

Support for stunnel is not covered by HP-UX support contracts. From the iExpress download page: "HP is not responsible for the ported defects from the Open Source communities. However, HP will report the defects to the related Open Source community and incorporate the appropriate defect repairs for each new release."

The combination of Security Containment and Stunnel provides a sufficient, if not superior, solution for Virtualvault TGP users transitioning to Security Containment.

**Lockdown of stunnel using Security Containment**

Stunnel on Security Containment is isolated in its own compartment in order to limit vulnerabilities. By limiting its write access to the filesystem, the probability of an attack requiring the creation of certain temporary files is reduced.

Network access is restricted to only the inbound and outbound ports stunnel is configured for.

Two privileges are granted to the stunnel process, one of them may not be needed depending on the specific configuration. CHROOT is needed by stunnel to run in its own a chroot jail. Optionally, NETPRIVPORT is used when stunnel is configured to bind to a network port lower than 1024.

For debugging and test purposes, the privilege COMMALLOWED is granted to stunnel to allow it to log to syslog. It is not recommended to grant this privilege permanently, since it grants unneeded cross-compartment IPC and network access to stunnel.
Creating the stunnel user and group

The stunnel process will run as user stunnel. File permissions will be set for group stunnel.

To create the user stunnel, use the command:

```bash
# useradd stunnel
```

To create the group stunnel, use the command:

```bash
# groupadd stunnel
```

Examine the newly created entries in `/etc/passwd` and `/etc/group` and take note of the UID and GID. They will be used later in this chapter.

Installing stunnel

Login in as root in the INIT compartment of the Security Containment system.

Use `swinstall` to install stunnel v4.09 from the *HP-UX 11i Version 2 Internet Express Non-COE Components* DVD (p/n 5013-6013). After installation, the Stunnel files will be located under `/opt/iexpress/stunnel`.

Note:

- A vital source of information about stunnel is http://www.stunnel.org.

The following stunnel parameters need to be configured before stunnel can be used:

- **Cert**: Certificate/key is needed in server mode and optional in client mode. For TGP functionality purposes, Stunnel works in server mode.
- **Chroot**: stunnel should always run in a chroot jail. See below for instructions.
- **Setuid, setgid**: set them to stunnel user and stunnel group.
- **Service level configuration**: configure the service and its `accept` and `connect` parameters.
- **Debug, output**: Set desired debug level and specify the log (output) file. The output file path is outside the chroot environment. See below for instructions.

Creating the stunnel chroot directory
A chroot directory should be created for the stunnel process. For example `<stunnel install dir>/var/stunnel`. In this case, if the stunnel installation directory is `/opt/iexpress/stunnel`, the chroot directory would be `/opt/iexpress/stunnel/var/stunnel`.

### Creating the stunnel log directory

A directory should be created for the stunnel log. The directory should be outside of the chroot jail. For example `<stunnel install dir>/var/log`. In this case, if the stunnel installation directory is `/opt/iexpress/stunnel`, the log directory would be `/opt/iexpress/stunnel/var/log`.

### Setting file permissions for stunnel

Since the stunnel process will be running as user stunnel, there are certain DAC permissions that need to be modified. Use the following table as reference. The default stunnel installation directory (SID) is `/opt/iexpress/stunnel`.

<table>
<thead>
<tr>
<th>Permissions</th>
<th>Owner</th>
<th>Group</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>0040</td>
<td>bin</td>
<td>stunnel</td>
<td><code>&lt;SID&gt;etc/stunnel/stunnel.pem</code></td>
</tr>
<tr>
<td>0040</td>
<td>bin</td>
<td>stunnel</td>
<td><code>&lt;SID&gt;etc/stunnel/stunnel.conf</code></td>
</tr>
<tr>
<td>0070</td>
<td>bin</td>
<td>stunnel</td>
<td><code>&lt;SID&gt;var/stunnel</code></td>
</tr>
<tr>
<td>0070</td>
<td>bin</td>
<td>stunnel</td>
<td><code>&lt;SID&gt;var/log</code></td>
</tr>
</tbody>
</table>

Having user `bin` as owner of the objects prevents the stunnel user from having ownership capabilities (chmod, chgrp, chown).

### Setting IPFilter rules for stunnel

IPFilter rules need to be added to allow the network traffic between STunnel and systems on either the Intranet or the Internet.

### Creating the stunnel compartment

Create a file with the following content and save it as `/etc/cmpt/in_stunnel.rules`. The IPC rules will be different for each implementation.
This example sets up communication for stunnel to listen to port 443 on the outside NIC and pass the data to port 80 of a compartment called *iws_app*. This setup would be useful to provide SSL Internet connections for an application that does not support SSL:

```
/*****************************/
compartment in_stunnel {
    /* Inside file system rules */
    #include </etc/cmpt/inside.common>

    /* stunnel file system rules */
    perm  read, write, create, unlink /opt/iexpress/stunnel/var/stunnel/
    perm  read, write, create, unlink /opt/iexpress/stunnel/var/log/

    /* stunnel IPC rules */
    grant server tcp port 443 out_iface  /* From client */
    grant client tcp peer port 80 iws_app /* To server app*/
}
/*****************************/
```

Run `/usr/sbin/setrules` to make the new compartment effective. The warning about the INIT compartment being redefined is expected and requires no action.

### Configuring RBAC for stunnel

In order to have the stunnel process operate in its own compartment, RBAC must be configured. A new role, authorizations and command authorizations are created and associated with each other to allow three basic operations:

- Start stunnel
- List the processes in the stunnel compartment
- Stop stunnel

Since stunnel must be started by user *stunnel*, a new role is associated with this user.

All the commands listed below are executed as root in the INIT compartment unless otherwise specified.

#### Configuring the stunnel user role

Use the following command to create the stunnel role:

```
# roleadm add vv-stunnel
roleadm: added role vv-stunnel
```

Assign the new role to user *stunnel*:

```
# roleadm assign stunnel vv-stunnel
```
roleadm assign done in /etc/rbac/user_role

Configuring the stunnel authorizations

The authorizations `hpux.vv-stunnel.start` and `hpux.vv-stunnel.stop` are created to start and stop the agent. The authorization `hpux.vv-stunnel.ps` is created to allow the system administrator to verify that the stunnel process is running:

```bash
# authadm add hpux.vv-stunnel.start
authadm added auth: (hpux.vv-stunnel.start,*)

# authadm add hpux.vv-stunnel.stop
authadm added auth: (hpux.vv-stunnel.stop,*)

# authadm add hpux.vv-stunnel.ps
authadm added auth: (hpux.vv-stunnel.ps,*)
```

Associate the newly created authorizations with the stunnel user role:

```bash
# authadm assign vv-stunnel hpux.vv-stunnel.start
authadm added 1 auth for role vv-stunnel

# authadm assign vv-stunnel hpux.vv-stunnel.stop
authadm added 1 auth for role vv-stunnel

# authadm assign vv-stunnel hpux.vv-stunnel.ps
authadm added 1 auth for role vv-stunnel
```

Configuring the stunnel command authorizations

Add entries to the command authorizations table for each of the commands to be used. Note that the UID and not the username must be used. In this example, the user `stunnel` has UID 105 and the group `stunnel` had GID 103:

```bash
# cmdprivadm add cmd=/opt/iexpress/stunnel/sbin/stunnel op=hpux.vv-stunnel.start
ruid=105 euid=105 rgid=103 egid=103 compartment=in_stunnel
prives=CHROOT,NETPRIVPORT
/opt/iexpress/stunnel/sbin/stunnel:dflt:(/hpux.vv-stunnel.start,*):105/105/103/103:in_stunnel:CHROOT,NETPRIVPORT:dflt:cmdprivadm added the entry to /etc/rbac/cmd_priv

# cmdprivadm add cmd=/usr/bin/kill op=hpux.vv-stunnel.stop ruid=105 euid=105
rgid=103 egid=103 compartment=in_stunnel
/usr/bin/kill: (/hpux.vv-stunnel.stop,*):105/105/103/103:in_stunnel:::
```
5. Configuring stunnel for TGP functionality

Command Priv/Administration (cmdprivadm) added the entry to /etc/rbac/cmd_priv

```bash
# cmdprivadm add cmd=/usr/bin/ps op=hpux.vv-stunnel.ps ruid=105 rgid=105 \
euid=103 egid=103 compartment=in_stunnel
/usr/bin/ps::(hpux.vv-stunnel.ps,*):105/105/103/103:in_stunnel::
```

cmdprivadm added the entry to /etc/rbac/cmd_priv

### Starting and stopping stunnel

To start the stunnel agent, as user `stunnel` or `root`, use the command:

```bash
$ privrun /opt/iexpress/stunnel/sbin/stunnel
```

To stop the agent, as user `stunnel`, use the command:

```bash
$ privrun kill -TERM $(cat \
/opt/iexpress/stunnel/var/stunnel/stunnel.pid)
```

To stop the agent, as user `root`, use the command:

```bash
$ privrun –c in_stunnel kill -TERM $(cat \
/opt/iexpress/stunnel/var/stunnel/stunnel.pid)
```

To list the processes running in the `in_stunnel` compartment, as user `stunnel`, use the command:

```bash
$ privrun ps -ef
```

To list the processes running in the `in_stunnel` compartment, as user `root`, use the command:

```bash
$ privrun –c in_stunnel ps -ef
```
6. User Installed Applications

Overview

At this point, the Security Containment system is enforcing a security policy similar to Virtualvault. Certain application level components and configuration information have been transferred from the Virtualvault system to the Security Containment system. For instance, CGI scripts defined in the Virtualvault tga.conf file have been copied to the equivalent location on the Security Containment system.

User installed applications have a high degree of implementation variability; therefore it is not possible to automatically move all application components from Virtualvault to Security Containment. Depending on the complexity of the application deployment, the Virtualvault transition toolkit may not be able to transition all application components to the Security Containment system. In such a case manual steps will need to be carried out by the administrator to get the application functioning. See Appendix E: VVSCAssist TODO on page 125 for more information. All other aspects of the application will need to be addressed manually.

Application Types

User installed applications can be characterized by the following criteria:

1. The application is supplied by a third party (e.g., an ISV, Independent Software Vendor)
2. The application is designed and implemented in house

Some deployments may have both kinds of applications.

Since Virtualvault is derived from HP-UX, there is a high degree of binary compatibility with Security Containment. However, Virtualvault is based on previous version of HP-UX (HP-UX 11.0). Therefore, it is necessary to ensure that the application being transferred from Virtualvault is supported on HP-UX 11iv2. For instance, an ISV may not support older versions of their product on newer operating system platforms.
Note:

- If the application is not supported on the HP-UX 11iv2 with Security Containment system, an updated version is required. For in house applications, recompilation may be needed. For third party applications, a supported version of the application must be obtained from the third party supplier.

**Application Integration: Architecture and Design**

Integrating an application on Security Containment is conceptually similar to application integration on Virtualvault. The integration involves configuration changes to ensure that the application operates as it should in an environment with containment, fine grained privileges and role based access control. There is a process defined to integrate an application on Virtualvault and a conceptually similar process can be carried out for Security Containment.

The rest of this section will describe the process for integrating an application. The application integration work that was done for the Virtualvault will be helpful in speeding up this process. Readers that are familiar with the integration of their application on Virtualvault may wish to skip to the last section of this chapter, Application Implementation and Debugging.

Notes:

- Although the application process is conceptually similar for Virtualvault and Security Containment, the two systems utilize different security mechanisms and APIs
- The HP Consulting and Integration organization offers fee based consulting to support the transition from Virtualvault to Security Containment, including user-installed applications.
- A good understanding of the Security Containment functionality is strongly recommended before attempting to integrate applications. Specifically, the integrator should be familiar with Security Containment compartments, Role Based Access Control (RBAC), fine grained privileges and auditing.
- Please see Chapter 8. Compartment Concepts on page 53 for more information.

The following are the recommended integration tasks:

- Update applications
- Design the security architecture
- Implement, debug and test the configuration
Designing the security architecture

A detailed examination of the application is needed to define an architecture that will address potential issues while maintaining functionality. These steps were likely carried out when the application was integrated with Virtualvault. Though all steps may not be necessary, HP recommends that each step be considered in order to maintain a high level of application security on the Security Containment system.

Most user installed applications are designed to work in an HP-UX environment and do not expect additional security mechanisms such as containment or fine grained privileges. Both Virtualvault and Security Containment utilize additional security mechanisms that can cause applications to not function correctly. For instance, applications expect to have full network access and be restricted to file system objects only by Discretionary Access Control (DAC).

It is up to the integrator to design an architecture that enables the application to fully function, while making it more secure.

It is strongly recommended that an acceptance test plan be developed. This plan will not only confirm the success of the integration, but can also serves as a requirements list for the architecture.

When designing the architecture, the following areas need to be evaluated:

- **Interaction between components**: The Interprocess Communication (IPC) between processes in different compartments has to be specifically allowed.
- **Access to files and network**: Modules should be placed in the different compartments in a way that permits the most access to file system and network resources with minimal compartment rules and without the need for privileges.
- **Use of network ports**: Some applications communicate over the network using a random port for each session. Use of a fixed TCP or UDP port to connect to other systems is preferable to random ports because more specific compartment rules can be implemented.
- **User id requirements**: Applications should run with their own user id. Furthermore, independent application modules should run with different user ids. The new user ids will be created by the integrator. Only the required RBAC authorizations and privileges should be assigned to that user ID.
6. User Installed Applications

- **Enterprise features**: Enabling functionality such as load balancing or distributed management may require additional effort. Some agents may require additional compartment rules to communicate over the network.

- **Administration**: The application may have an administration module and may require special attention. For instance, it may require simultaneous write access to a group of resources that are not write-accessible from any given compartment. Therefore, the administration module may require its own separate compartment.

Different compartments are used for different types of processes, depending on the function that a process has and what type of access to the different system resources it needs. Files and directories will be assigned to compartments according to the type of access (read, write) needed. For example: for an application running in the Outside compartment, configuration files are assigned to INIT, while log files are assigned to an Outside compartment. Following is a description of the compartments that have been mapped from Virtualvault to Security Containment. These compartments will also be configured on the Security Containment system by the transition toolkit. However, multiple instances of the Outside and Inside compartments are supported on the Security Containment system.

- **Syssh compartment**: This compartment should not be used for application integrations. It is reserved for the audit subsystem. Running an application process in this compartment would risk the integrity of the audit subsystem.

- **Outside compartments**: These compartments should contain the application modules that receive Internet client requests. The Outside compartments isolate applications from the Intranet. In case the application is compromised, the attack will be contained in the Outside compartment with no direct access to the Intranet. At the same time, modules in this compartment are protected from attacks coming from the Intranet. Log files written to by the module should be assigned to the same compartment (see note below).

- **Inside compartments**: These compartments contain the application modules that communicate with back end servers. These modules have restricted access to the Intranet via compartment rules. Log files written to by the module should be assigned to the same compartment (see note below).

- **INIT compartment**: This compartment is equivalent to Syslo in Virtualvault. All the application and system configuration files are assigned to the INIT compartment. No application related processes should run in this compartment because they would have read/write access to the configuration files.
The Security Containment documentation refers to a special compartment called the INIT compartment. The transition toolkit redefines the INIT compartment and configures it to serve as the Virtualvault Syslo compartment to be consistent with the Virtualvault security policy.

**Note:**

- It is possible that application modules in separate compartments need write access to the same file. In that case, file system compartment rules can be added to each compartment to allow write access to it. This effectively assigns the same file to multiple compartments simultaneously. For instance, this can be used so that processes in the Outside and Inside compartments could write to the same file. In Virtualvault, multilevel directories are used to address this case.

**Discretionary Access Control Considerations**

As part of the hardening process, DAC permissions were modified throughout the system, limiting access to the minimum required. Concurrently, DAC permissions on application files should be changed to the most restrictive setting, without interfering with required functionality.

DAC permissions should be modified using the following criteria:

- **Files and directories should be owned by a non-login user different than the application UID**: The owner of a file has the ability to change the DAC permissions of that file. Assigning the file to a non-login user prevents an attacker from using that UID to log in and modify the permissions. Setting the owner to a UID different than the UID of the executing application, provides protection if the application is compromised. Otherwise, compromise of the application could allow the intruder to have access to the system with the application’s UID.

- **Permissions should be set by group**: Since the owner permissions are set for a non-login user, the application will access the file via group permissions.

- **Configuration files should be read-only**: The only process that should have write access to these files is the application’s administration program. Write access can be assigned by group permission to a group that only the application’s administration process belongs to and read access to others. It is recommended that the DAC override privilege not be assigned to the application’s administration program.

- **Libraries, scripts and binary executables should be read-execute**: These are the minimum permissions required to execute code. These permissions should be set for the group.
6. User Installed Applications

- **Log files should be read-write**: No execute permissions are needed for log files.
- **All other files should be read-only (exceptions are possible)**: There may be application specific exceptions to this rule, but in general, all other types of files should be read.
- **Directories where files are created at run time should be read-write-execute**: In order to create an entry in a directory, write access is needed. Read access is needed to list the content of the directory and execute access is needed to traverse it.
- **All other directories should be read-execute**: If no files are normally created in a directory (e.g. a directory with code only, like /usr/bin or /usr/lib), there is no need for write permission to the directory.

Application Implementation and Debugging

Before starting the implementation phase on the Security Containment System, you should examine the application on the Virtualvault system. Collecting the following information will help to implement the application on the Security Containment system.

1. Determine all of the files that comprise the application
   - Determine file system attributes associated with each file (e.g. owner, group, permissions, compartment, privileges)
2. Determine system configuration changes that were made to accommodate the application
3. Determine other utilities (not delivered with Virtualvault) that are used by the application
4. Determine application related processes that require privileges to operate on Virtualvault

The application on the Security Containment system will typically be set up similarly to the way it was on the Virtualvault system. However, there are exceptions. Two key exceptions are the following:

1. Using a newer version of a component on Security Containment than was used on Virtualvault. For instance, if a newer version of the Apache Web server or Tomcat is used, the configuration may be different than it was on Virtualvault, directory and file names may change between versions.
2. Each file is not associated with a compartment on a Security Containment system, as is the case on a
Virtualvault system. Instead file system rules are used to enforce file containment.

However, the transition toolkit provides a utility, rulegen.sh, which accepts the pathname of a file and the compartment it was associated with on Virtualvault. When executed on the Security Containment system, appropriate rules are configured to match the security policy of Virtualvault.

Once the application installation is complete, all application related files have been placed on the Security Containment system and all attributes have been set appropriately, the debugging phase can begin. At this point the integrator makes incremental changes to the environment and tests the functionality of the application until the desired level of functionality is achieved. As the system is secured by assigning processes to compartments and changing rules, the application will fail when it cannot access a resource. The integrator needs to determine the resource and the type of access needed and grant the minimum access required.

Sometimes it is necessary to investigate the source of an application error due to access restrictions. In addition to documented Security Containment commands, there are tools that can aid in this process. The following is a list of some of those tools:

- **TRIALMODE:** TRIALMODE is a special attribute that can be assigned to executables on the Security Containment system, in the same way privileges are assigned. TRIALMODE is not a privilege, it is a flag signaling Security Containment to log all privilege related activity to /var/adm/syslog/syslog.log. To debug using TRIALMODE, the integrator assigns TRIALMODE to the application executable. Then the application is run as root (all privileges granted). The system will log all privileged operations to the log file. The integrator can then examine the log file to determine which compartment rules need to be added to allow the desired operation. If for some reason compartment rules are inadequate, the log file indicates the privileges needed. It is recommended that specific compartment rules be assigned to applications when possible, rather than broad privileges.

- **Audit trail:** The audit trail may show an unsuccessful attempt to access an object. Audit collects a great deal of information that can be overwhelming. It is usually a good practice to start the audit process, reproduce the error, and stop the audit process to limit the size of the audit trail to be examined. See the `audit(4)` man page for more information.

- **Tusc:** tusc will trace the execution of a process and report it to the user. It shows useful information as system calls
and return codes. This information can be used to determine if the application needs access to certain objects. Tusc can be downloaded from:
http://h21007.www2.hp.com/dspp/tech/tech_TechDocumentDetailPage_IDX/1,1701,2894,00.html
7. Post Transition Clean Up

Overview

After the Virtualvault to Security Containment transition is completed and all applications integrated, the administrator should remove unneeded files and reclaim disk space used in the transition on both the Virtualvault and Security Containment systems. The directory `vvsnapshot_staging/` and the file `vvsnapshot_staging.tar` can be deleted by the administrator. The default location for these is `/opt/hpvv/`. However, alternate locations may have been specified during the VVSnapshot script (VV system), and/or the VVSCAssist script (SC system) executions.

To remove unneeded files, the administrator should use the `swremove(1M)` command to remove the following software selections:

- `hp_hpxharden depot` (SC system)
- `hp_vv_sc.vvscassist fileset` (SC system)
- `hp_vvsnapshot depot` (VV system)

Note:

- Administrators should not `swremove (1M)` the `vватьсяtion depot`. Doing so will undo the compartment definitions set up by the Virtualvault to Security Containment transition kit.
Part II – Ongoing Administration

The following chapters refer to the ongoing administration of the Security Containment system. These procedures will help the system administrator maintain a security policy similar to that of Virtualvault on the Security Containment system.

8. Compartment Concepts

About Compartments

Virtualvault has four predefined compartments with well defined and fixed access rules. Security Containment has one predefined compartment and supports the creation of additional compartments with user defined access rules.

Security Containment compartments can be created to closely resemble Virtualvault compartments. Access rules can be created to mimic those of Virtualvault.

Notes:

- There is a caveat. Security Containment requires the existence of the INIT compartment. By default, INIT has full access to all other compartments, however INIT can be redefined and the system will treat it as any other compartment. For Virtualvault transition systems, the INIT compartment has been redefined and is equivalent to the Virtualvault Syslo compartment.
About Virtualvault Compartments

In addition to Discretionary Access Control (DAC), Virtualvault determines access to objects via Mandatory Access Control (MAC). Virtualvault uses MAC to define four predetermined compartments. Every subject and object has a sensitivity label. The security label contains information necessary to determine MAC access. The sensitivity label has two parts: classification and compartment. In Virtualvault, the classification is the same (SYSTEM) for all sensitivity labels. Virtualvault compartments are OUTSIDE and INSIDE. There are four possible labels for Virtualvault objects:

- SYSTEM (Syslo)
- SYSTEM INSIDE
- SYSTEM OUTSIDE
- SYSTEM INSIDE OUTSIDE (Syshi)

Virtualvault compartments are illustrated as follows

According to MAC rules, if a label has all of the compartments of another label, the first label dominates the second. Access between subject and objects (for example, a process and a file) is determined by the following rules:

**MAC Rule 1:** If the label of a subject dominates the label of the object, read access to the object is granted

**MAC Rule 2:** If the label of the subject equals the label of the object, write access to the object is granted
The following matrix lists the access a process has to a file according to MAC rules:

<table>
<thead>
<tr>
<th>Process Label</th>
<th>Syslo</th>
<th>Outside</th>
<th>Inside</th>
<th>Syshi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syslo</td>
<td>Read, Write</td>
<td>Read</td>
<td>Read</td>
<td>Read</td>
</tr>
<tr>
<td>Outside</td>
<td>No access</td>
<td>Read, Write</td>
<td>No access</td>
<td>Read</td>
</tr>
<tr>
<td>Inside</td>
<td>No access</td>
<td>No access</td>
<td>Read, Write</td>
<td>Read</td>
</tr>
<tr>
<td>Syshi</td>
<td>No access</td>
<td>No access</td>
<td>No access</td>
<td>Read, Write</td>
</tr>
</tbody>
</table>

By default, Virtualvault systems have two network interface cards (NIC) with labels Outside and Inside for the Internet and Intranet networks respectively. The following matrix lists the access a process has to the networks according to MAC rules:

<table>
<thead>
<tr>
<th>Process Label</th>
<th>Syslo</th>
<th>Outside</th>
<th>Inside</th>
<th>Syshi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syslo</td>
<td>No access</td>
<td>Read, Write</td>
<td>No access</td>
<td>No access*</td>
</tr>
<tr>
<td>Outside</td>
<td>No access</td>
<td>No access</td>
<td>No access</td>
<td>No access*</td>
</tr>
<tr>
<td>Inside</td>
<td>No access</td>
<td>No access</td>
<td>Read, Write</td>
<td>No access*</td>
</tr>
</tbody>
</table>

* Can be read from the NIC

About Security Containment Compartments
A compartment is associated with each Security Containment process. Security Containment objects do not have a sensitivity label associated with them. Instead, compartments are defined with rules that determine the access to objects from processes running in the compartment. There are no classifications in Security Containment, only compartments.

For transitioning Virtualvault customers, HP has defined compartments during the transition process that resemble the Virtualvault architecture, while taking advantage of Security Containment compartment features. Security Containment has a permanent, default compartment named INIT. For Virtualvault transition purposes, INIT is reconfigured to serve as the Syslo compartment. The Syshi compartment is created for the same purposes as in Virtualvault (audit). The Outside and Inside compartments have been partitioned into multiple compartments for more security and ease of management. The NICs are isolated in their own compartments.

The above illustration is not quite accurate, since it does not illustrate that all Outside compartments are at the same compartment hierarchy level. The same applies to Inside compartments. It’s easier to interpret the illustration as depicting one partitioned Outside compartment and one partitioned Inside compartment.

Note that in Virtualvault, all processes in the Outside compartment could communicate with each other via IPC. In the architecture illustrated above, processes in different Outside compartments cannot communicate with each other via IPC unless explicitly allowed by a compartment rule.
In Security Containment, the INIT compartment always exists and by default has full access to the other compartments. However, when defined explicitly, the INIT compartment behaves like any other user-defined compartment.

For the Virtualvault transition, the INIT compartment has been explicitly defined with rules to make it equivalent to Syslo. The redefined INIT has full access to most of the filesystem and no access to the NICs, except for certain (optional) rules for SSH, DNS, SWAgent and similar services.

Virtualvault provided client-server privileges to allow communication between processes in different compartments (usually Outside and Inside). A process with the client privilege could only communicate across compartments with a process with the server privileges.

Instead of using privileges, Security containment users are expected to add compartment rules to achieve the same functionality. This approach is most secure when processes are isolated in their own compartments. However, the more compartments in the system, the more complex system management becomes. A reasonable middle ground is to isolate groups of related processes with the same trust level in their own compartment. For example, a web server process and all its children are a process group; a second web server with its children is a separate process group.

For transitioning customers, HP creates separate compartments for each Outside and Inside process group. Therefore, instead of having multiple web servers running in one Outside compartment (as in VV), each of the web servers will have their own compartment. All compartments on the Outside will have similar restrictions and permissions, resembling the Virtualvault Outside compartment restrictions and permissions. The same applies to Inside compartments with respect to the Virtualvault Inside compartment restrictions and permissions. There is no interprocess communication (IPC) between Outside compartments by default. The same applies to Inside compartments.

This architecture creates multiple instances of similarly restricted Outside and Inside compartments. It can be thought of as partitioning the Virtualvault Outside and Inside compartments.

Having multiple Outside and Inside compartments has security and management advantages. The more isolated the processes are, the more secure. Having separate compartments (and their rulesets) for applications makes it simpler to add or delete rules without the risk of unnecessarily exposing or restricting another application.
Compartment rules apply to all objects in the compartment. In the Virtualvault model, the NICs are in the same compartments as processes. In Security Containment this is less desirable because access rules apply to all processes as well as NICs in the compartment. Therefore, if a rule allows the NIC to access the Internet, all processes in that compartment will be exposed to the (hostile) Internet traffic.

Having NICs isolated in their own compartment allows the creation of access rules from any compartment without having any process unnecessarily exposed by that rule. For transitioning Virtualvault customers, HP creates a compartment for each NIC.

To facilitate the transition from a static four-compartment model to a multi-compartment model, HP has a list of recommendations:

- **Include files for common rules**: HP provides two files with rules that are common to all Outside and Inside compartments respectively. When adding a rule that applies to all Outside compartments, for example, the user adds the rule to the include file, instead of having to add the rule to each compartment definition file. The include files are named `inside.common` and `outside.common`.

- **Naming conventions**: HP has specified naming conventions for compartments and compartment definition files. The compartment definition files are the files (in `/etc/cmpt`) with extension “.rules” that the command `setrules(1M)` uses to create compartments and set their rules. These conventions make it easier to manage compartments from the command line. Note that Security Containment compartment names are case sensitive. The conventions are:
  - Each compartment definition file contains one compartment definition.
  - Compartment definition files are named the same as the compartment with the addition of the extension “.rules”.
  - Outside web server compartment names start with “ows_”.
  - Inside web server compartment names start with “iws_”.
  - Non-web server Outside compartment names start with “out_”.
  - Non-web server Inside compartment names start with “in_”.
  - The Outside NIC compartment is named “outside_iface”.
  - The Inside NIC compartment is named “inside_iface”.

There are several advantages to using the above recommendations:
8. Compartment Concepts

- The naming conventions simplify administration. With a command like `ls /etc/cmpt` the user will know how many compartments are defined, their names, which ones are outside and inside and which ones are for web servers.
- Having one compartment definition per "rules" file makes it easy to locate compartment definitions.
- Deleting a compartment is accomplished by removing the corresponding "rules" file and making the rules change effective with the `setrules(1M)` command.
- Adding a compartment is accomplished by creating a corresponding "rules" file, following the conventions listed above, and making the rules change effective with the `setrules(1M)` command.

File and network access rules have been defined to replicate those on Virtualvault. In the case of network rules, they don’t need much maintenance. File access changes need more administrator action than in Virtualvault. To assign a file to a compartment, Virtualvault users changed the file’s MAC label. Security Containment users will have to perform more than one step depending on which compartment the file is assigned to. The Rulegen tool simplifies this task. For more information about Rulegen see Chapter 7, File Rule Generator.

- **INIT**: The redefined INIT compartment grants processes in the INIT compartment full access to all files. All other compartments are granted read access to all files by default. For Virtualvault transition purposes, INIT is reconfigured to be equivalent to the Syslo compartment.
- **Inside and Outside**: The administrator needs to add a rule (to the common rules files) granting full access to the compartments (Outside or Inside compartments) the file will be assigned to. Another rule is needed to deny access to the complementary compartments (Inside and Outside are considered complementary). A rule needs to be added to the INIT compartment to deny access from it. In the case of assigning a file to only one of the Outside compartments, a rule is added to the other Outside compartments denying access to that file. The same applies when assigning a file to only one Inside compartment.
- **Syshi**: The administrator adds a rule to the Syshi compartment granting full access and adds rules to all other compartments denying access to the file.

About Starting Processes in Compartments

This section describes and compares the concepts of starting processes in compartments as it relates to Virtualvault and Security Containment.
Starting a Process in Compartments - Virtualvault

By default, when a Virtualvault process is started, it inherits the compartment from its parent process. Sometimes it is necessary to start a process in a different compartment than its parent. Virtualvault provides the `epa(1M)` command to start processes in a user specified compartment and additional process credentials (e.g. privileges, UID, GID, etc.).

Starting a Process in Compartments – Security Containment

By default, when a Virtualvault process is started, it inherits the compartment from its parent process. Security Containment provides two ways to start processes in a specific compartment: RBAC and `setfilexsec(1M)`.

With RBAC, the administrator can define the compartment in which the process will be started and additional process credentials (e.g. privileges, UID, GID, etc.). RBAC requires the use of the command `privrun(1M)` to take effect. RBAC can restrict use of the commands based on user authorizations.

`setfilexsec(1M)` can be used to specify the compartment and privilege sets that will take effect when the binary is executed (this is similar to the set-UID bit functionality). These settings remain in effect until the binary is modified, in which case the settings will be restored the next time `setfilexsec -l` is executed. By default, `setfilexsec -l` is executed at boot time. In the Virtualvault transition configuration for Security Containment, `setfilexsec -l` execution at boot time is disabled, so that a modified executable is not given privilege without the administrator taking explicit action.

Configuration Examples

The following examples illustrate how a Virtualvault application architecture will translate to the Security Containment compartments proposed in this document.

Example 1: CGI

In Virtualvault, a CGI request from the Internet travels the following path:
1. The CGI request is received by a web server in the Outside compartment. The server raises a privilege to talk as a client to a process in another compartment (this privilege is not specific to a particular destination compartment) and connects to the TGA port.

2. The TGA process is running in the Inside compartment with a raised privilege to receive connections from other compartments. The TGA receives the request and determines which CGI needs to be executed.

3. The CGI executes in the Inside compartment (same compartment as the TGA) and connects to an Intranet server.

4. The reply from the Intranet server travels the reverse path to the Internet client.

This configuration works well for a single web server. Optionally, a firewall (not shown in the diagram) can be used to prevent attempts to connect from the Internet to ports other than those intended for the Outside web server. Another firewall (not shown in the diagram) could be used to control what servers and which ports are accessed from the Inside NIC.

The Security Containment configuration provides the same functionality with a higher level of isolation between processes:
1. The CGI request is received by a web server in the Outside compartment using a rule allowing traffic from the Outside NIC’s compartment to the web server’s at a specific port. The request is passed to the Inside CGI web server, allowed by a compartment rule.

2. The CGI web server receives the request and determines which CGI needs to be executed.

3. The CGI executes in the same compartment as the CGI web server and connects to an Intranet server using a compartment rule that allows communication between that Inside compartment and the Inside NIC’s compartment, controlling which destinations and ports are accessed.

4. The reply from the Intranet server travels the reverse path to the Internet client.

Since Security Containment compartment rules are port-specific, attempts to connect from the Internet to ports other than those intended for the Outside web server will be blocked. In a Virtualvault setting, a firewall was required to accomplish this functionality.

This configuration allows for more applications to run on the system while maintaining the security of individual applications, as illustrated in the next example.

**Example 2: CGI and Tomcat**

The previous configuration was relatively simple and works well on both platforms. However, adding more web servers may have undesirable security implications. For example, if there is more than one Outside web server and one is supposed to run CGIs while the other talks to (Inside) Tomcat, they can both access the
TGA and Tomcat, since both will have the client privilege raised. At the same time, TGA and Tomcat can receive connections from any process in the Outside compartment with the client privilege raised.

This example shows how the Virtualvault and Security Containment environments compare when using multiple mechanisms to provide information from Intranet servers.

Virtualvault applications share the Outside and Inside compartments, as illustrated below.

This diagram illustrates two independent implementations. The first implementation receives web client requests from the Internet and uses the TGA to execute CGI programs, which in turn communicate with an Intranet server.

The second implementation uses Tomcat to execute Java servlets, which in turn communicate with an Intranet server different than the server accessed by the CGI application.

The inter-compartment communication between these components is allowed by client-server privileges.

There are some unintended consequences in this setting. Processes have full access to the network connected to their compartment, allowing requests to be received on any port. Furthermore, with client-server privileges, any privileged client can talk to any privileged server. For example, in the CGI path, there is nothing preventing an Internet peer from trying to connect to any port of
the Outside web server 1 (OWS1). The OWS1 can connect to Tomcat, since OWS1 has client privilege. Tomcat can connect to any Intranet server in any port. The diagram illustrates other communication paths that were not originally intended.

The connections from the Internet and to the Intranet can be controlled by firewalls, but not the connections between privileged clients and servers inside Virtualvault.

The same applications on a Security Containment system look as follows:

Rules between the NIC compartment and the web server compartments specify which ports are allowed (in the example above, those ports are 80 and 443). Rules specify which Outside compartment can connect to the CGI web server compartment and which can connect to the Tomcat compartment. Rules between the Inside compartments and the Inside NIC compartment specify the allowed destinations and ports.
9. File Rule Generator

Overview

Virtualvault has a MAC label (compartment) associated with each file, Security Containment does not. Security Containment uses rules to enforce access to the file by a process.

`rulegen.sh` is a tool used to generate filesystem access rules that comply with the Virtualvault security policy on a Security Containment system.

In the Virtualvault system, a file in a compartment can only be written to by processes in that compartment. Processes in other compartments have read-only access or no access at all.

In the Security Containment system, filesystem objects are not associated with a compartment. A compartment with no filesystem rules, which is the default, has full access to the filesystem. In order to limit MAC access to a filesystem object for all system processes, a rule must be added to each compartment in the system.

For example, if a file needs to be assigned to an Inside compartment (named Inside-1; for example), rules denying access to it need to be added to the other compartments. For the Sysh compartment, a rule granting read only access is needed. For all the Outside compartments, a rule denying all access is needed. For the Inside compartments other than Inside-1, a rule denying all access is needed. For the INIT compartment, a rule denying all access is needed. Finally, even though the default is full access to the filesystem, a rule granting full access to the file should be added to Inside-1. This prevents a more general rule in Inside-1 (for example, a rule that grants read-only access to a parent directory of the file) from limiting access to the file.

For Virtualvault transitioning customers, HP provides, as part of the transition kit, `rulegen.sh`. This tool simplifies the task of adding compartment rules that comply with the Virtualvault security policy.
General Information:

- The Security Containment INIT compartment is equivalent to the Virtualvault Syslo compartment. Most files belong to the INIT compartment by default. Files belonging to INIT are readable from all other compartments.
- Files which belong to inside or outside compartments are not readable by other inside or outside compartments nor by the INIT compartment. They are readable from syshi.
- Files which belong to syshi are not readable by any other compartment.

rulegen.sh requires write access to the compartment definition files. rulegen.sh can only be executed by root in the INIT compartment.

Required environment

The rulegen tool requires the presence of some of the VV Transition compartment definition files. The required files are found in /etc/cmpt. Rulegen.sh assumes that the following files exist:

- /etc/cmpt/INIT.rules - definition of the INIT compartment
- /etc/cmpt/syshi.rules - definition of the syshi compartment
- /etc/cmpt/inside.common - compartment rules common to all inside compartments
- /etc/cmpt/outside.common - compartment rules common to all outside compartments
- /etc/cmpt/<name> - definition of the <name> compartment. The <name> compartment is any compartment given to the Rulegen tool as a parameter. See Chapter 8, Compartment Concepts on page 53 for more information.

Note:

- If a needed file does not exist or is not writeable, rulegen.sh will print an error message and terminate.

Rulegen Syntax

/opt/hpvv/bin/rulegen.sh takes two forms of arguments:

rulegen.sh (-f filename | -c compartment path ...)

- -f filename: read the specified file for information on what compartments to assign paths to. Each line of the file should contain a compartment name, whitespace, and a path to assign to that compartment.
- -c compartment path [path ...] assign each path listed to the specified compartment.

The two forms may be combined and/or repeated in one command, e.g. rulegen.sh -f file -c comp1 path1 -c comp2 path2
Warning:

- `rulegen.sh` has no facility to detect duplicate or conflicting rules. This means that if you try to assign a path to two compartments, it will appear to succeed, but the results will be unpredictable.
10. Configuring Role Based Access Control

Basic configuration

VV Transition and user roles

VV Transition currently has two defined user roles for role based access control (RBAC): Administrator and Webadmin. The Administrator role is able to execute all the commands defined in the RBAC database. The Webadmin role is a smaller subset of these commands and contains abilities to manage the Apache web servers and the Tomcat server.

Adding Webadmin role to a user

To add the Webadmin role to a user, log in as root and add the user and the role with the following command:

```
# roleadm assign user1 Webadmin
roleadm assign done in /etc/rbac/user_role
```

Performing tasks with the Webadmin role

A user with the Webadmin role is able to execute administration commands for the Apache and Tomcat servers. Commands are executed using `privrun(1M)`.

Apache web server commands:

There are several methods to start the Apache web server, using the 
–k argument to specify the desired method: start, restart, graceful, updatecrl, or start –DSSL

```
$ privrun /opt/hpws/apache/bin/httpd \
   -f /opt/hpws/apache/webproxy/servers/wp-[SERVER NAME]/conf/httpd.conf \
   -k start
```

To stop the Apache web server:

```
$ privrun /opt/hpws/apache/bin/httpd \
   -f /opt/hpws/apache/webproxy/servers/wp-[SERVER NAME]/conf/httpd.conf \
   -k stop
```
To test the apache configuration file:

$ privrun /opt/hpws/apache/bin/httpd \
-f /opt/hpws/apache/webproxy/servers/wp-[SERVER NAME]/conf/httpd.conf -t

To show the Apache help file:

$ privrun /opt/hpws/apache/bin/httpd \
-f /opt/hpws/apache/webproxy/servers/wp-[SERVER NAME]/conf/httpd.conf -h

Tomcat server commands:

For the following commands, the environment variable JAVA_HOME should be set to /opt/java1.5.

To start the Tomcat server:

$ privrun /opt/hpws/tomcat/bin/startup.sh

To stop the Tomcat server:

$ privrun /opt/hpws/tomcat/bin/shutdown.sh

Advanced Configuration

By design, the compartments on the VV transition target system are not visible by many of the standard UNIX commands. For example, the ps(1M) command will only display processes running in the current compartment and not any of the processes in other compartments. Therefore, when root executes ps(1M) from the INIT compartment, the processes in the web server compartments are not listed. For more information read the rbac(5) man page.

Reporting process status in a compartment

The ps(1M) command capability has been added to RBAC when a web server instance is created by the vvws_createxsec script. By executing the following command, all processes in the specified compartment will be displayed.

$ privrun -c [COMPARTMENT NAME] ps -ef

Configuring a command to execute in a compartment

To execute a command in a specific compartment, the command must be added into RBAC. The following steps document how a command can be added into RBAC.
NOTE: The `ps(1M)` command is used in this example. This command capability will already exist for each web server instance. For other commands, use the following steps as an example.

1. Add an authorization to RBAC, such as `hpux.hpws.ps`
   
   `authadm add hpux.hpws.ps`

2. Assign the authorization to Webadmin role
   
   `authadm assign Webadmin hpux.hpws.ps`

3. Add the command and privileges into RBAC. Note that the UID/GID and not the username/groupname must be used. In this example, the user `iwww` has UID 104:
   
   `cmdprivadm add cmd=/usr/bin/ps op=hpux.hpws.ps 
   ruid=104 euid=104 rgid=57 egid=57 
   compartment=iws_cgi`

4. Execute the `ps(1)` command in the compartment
   
   `$ privrun -c iws_cgi ps -ef`

### Configuring and executing a command in more than one compartment

To have a command executable in multiple compartments repeat steps 3 and 4 from the previous section using each compartment name.
11. Web Server Transition

Web Server Functionality Overview

Overview of HPWS Apache


The Apache-based web server and the Tomcat-based Servlet Engine form the base functionality required for this transition. Included with the Apache-based web server product are a number of webproxy scripts to facilitate creation of separate, chrooted web server instances. When combined with the HPUX Security Containment features, these separate chrooted web server instances form the base architectures required to implement Virtualvault transition functionality.

The following chapter details the background and procedures that have been incorporated into the automated transition scripts to set up and configure the web server on Security Containment. The procedures are carried out during the transition by the VVSCAssist script. Additional web servers may be added after the transition using the vvws_createxsec script.

It should be noted that the webproxy utilities use the same server binary as the standard configured Apache server instance. The standard Apache instance and GUI interface are designed to execute a single server instance with multiple virtual servers. The webproxy server instances are geared towards multiple server instances, each executing in a separate chroot environment.

While not a requirement, webproxy server chroot environments are generally at /var/jail/wp_internet, /var/jail/wp_intranet, or /var/jail/wp_<instancename>/.

A webproxy server will read its configuration files and open log files from its configuration root in /opt/hpws/apache/webproxy/servers/wp_<instancename>/, and then chroot(2) to its runtime environment in /var/jail/.

The webproxy instances are started and stopped separately from the main Apache web server instance during system startup and shutdown. Webproxy has its own /sbin/init.d/ script to traverse
the /opt/hpws/apache/webproxy/servers/ directory and execute
the server-specific apachectl script for each server found.

Securing an Apache web server instance using Security Containment

The goal of web server lockdown is to isolate the web server from external processes, as well as to contain the web server in the event that it becomes compromised. No matter its purpose, each web server instance is similarly locked down and then selectively opened up to allow it to do its specified job. That job may be to serve CGIs (Apache/TGA on Virtualvault), to proxy information from the hostile environment to the non-hostile environment (Apache/Webproxy on Virtualvault), or to pass information to a secure servlet server (Apache/Tomcat on Virtualvault). There are many other configurations that are possible, but the majority of those configurations are similar to these three specific cases.

1. **File Isolation**: two components: chroot and compartment rules confine the web server and strictly control access to files in the chroot environment.

   The HP-UX Web server suite server is a modified version of the Apache Server Foundation’s base server and therefore able to utilize a chroot directive. Generally, the web server obtains handles to necessary files (configuration, logs, etc) before chrooting to a directory designated by the chroot directive. While chrooting is reasonably secure, the Security Containment feature of HP-UX allows for further isolation of the web server instances by placing it in a compartment with kernel-enforced rules that strictly define access to files on the system.

2. **Network Isolation**: compartment rules restrict access to specific protocols and ports.

   While the Apache Listen directive forces Apache to listen only on specified addresses and/or ports, this is not sufficient to prevent Apache from unauthorized network access should the process be compromised. The goal of network isolation is to prevent Apache from making unauthorized outbound connections, and to prevent unauthorized connections to Apache from accepted listening ports (example: a web server listening on the loopback connector will accept connections from ANY process on the system capable of requesting a connection through the loopback.).

   To increase security beyond the Apache Listen directive, the Security Containment system can restrict
communication to a specific listening protocol, IP address, and port number, as well as the source compartments from which connections are allowed. Thus, compartment rules specify exactly the compartments, protocol and ports from which the web server will accept.

3. **IPC Isolation**: compartment rules prevent IPC based attacks on processes in other compartments as well as protect the web server from IPC based attacks launched by processes in other compartments.

On most servers, even with the web server properly configured, a malicious process may still attack a server through signals and other interprocess communication methods. Base web server directives provide no protection from these types of attacks. Security Containment, though, allows rules that provide a kernel-enforced method to prevent IPC attacks.

The following is a compartment configuration rule set that provides good general protection for a web server instance chrooted at `/var/jail/wp_internet/`:

```plaintext
/*****************************/
 compartment iws_apache {
 /* General File System Rules */
 perm read /
 perm all /dev/null
 perm read /dev/random
 perm read /dev/urandom
 perm read /opt/hpws/apache
 perm all /opt/hpws/apache/webproxy
 perm all /tmp
 perm read /usr/lib
 perm read /var/run/egd-pool
 perm all /var/tmp
 /* Server-specific File System Rules */
 perm all /opt/hpws/apache/webproxy/run
 perm read /opt/hpws/apache/webproxy/servers/wp-inside/conf
 perm all /opt/hpws/apache/webproxy/servers/wp-inside/logs
 /* Server-specific Chroot environment File System Rules */
 perm all /var/jail/wp_intranet
 perm all /var/jail/wp_intranet/dev/null
 perm all /var/jail/wp_intranet/opt/hpws/apache/webproxy/run
 perm all /var/jail/wp_intranet/tmp
 perm all /var/jail/wp_intranet/usr/bin
 /* IPC Rules */
```
/* Give Apache a network interface to listen on. * A typical configuration would be to: * * server to the outside compartment on designated http port(s) ** server to the outside compartment on designated https port(s) * client to the inside interface compartment on any port(s) * the server might be proxying or connecting to ** server to the inside interface compartment on any port(s) * that the inside network users might need to connect to. * may need an IPC rule for the cgisock interface */

grant bidir tcp peer port 80 in_iface */
server to the inside network compartment/interface */
grant server tcp port 9080 in_iface
grant server tcp port 9443 in_iface
client to tomcat on port 8009 rule is in the inside cmpt*/

In this set of rules, the server is given read-only access to its own configuration files, write access to its log files and runtime files, and read access to required libraries and random devices. The server is also allowed to accept connections on port 80 and on port 443 from another compartment named out_iface.

Processes in the ows_apache compartment may not initiate communication with any other compartment unless the other compartment grants specific access to the ows_apache compartment. Processes executing in the ows_apache compartment have write access only to directories where a typical apache web server will write to files (/dev/null, run, logs, and tmp directories). Therefore, both the apache configuration and the kernel (via the compartment configuration) restrict actions of the apache web server executing in this compartment. Should the apache process become compromised the kernel and
the chroot environment maintain system security from the compromised process.

This general configuration forms the basis for the more specific web server functionality required to transition Virtualvault web architectures to Security Containment.

**Using Role Based Access Control to Start and Stop the Web server Instances**

Even though the web server has been locked down, there still remains the question of who and how the server can be stopped, started and configured. On Virtualvault, the `webadmin` and `hostadmin` command authorization limited the ability to start and stop servers to those users with these authorizations. Servers started in their appropriate Virtualvault compartments the `epa(1M)`.

Security Containment Role Based Access Control (RBAC) provides similar functionality to the command authorizations of the Virtualvault. RBAC allows the administrator to define a `Webadmin` role to limit the users who may start and stop web servers, and to start the Apache/Tomcat httpd/java processes in the assigned compartment. Instead of the Virtualvault `epa(1M)` command, Security Containment provides a program called `privrun(1M)`, which allows the user to run a particular command in a particular compartment as a specified user and/or with privilege.

1. The list of binaries that will be executed is determined

   All httpd instances utilize the standard HPWS Apache, `/opt/hpws/apache/bin/httpd` binary, and are invoked through a server’s associated `apachectl` script.

   Jakarta Tomcat by default utilizes the Java 1.5 JDK for HP-UX java instance. It is invoked universally through the `/opt/hpws/tomcat/bin/startup.sh` script.

2. An authorization scheme and naming convention is determined

   Since there are no default authorizations for web server start and stop on Security Containment by default, one must be created. In the case where a single user should have the authorization to stop and start all web servers and the tomcat servlet server, some sample server authorizations that might be applicable are:

   `hpux.hpws.webproxy.start`
   `hpux.hpws.webproxy.stop`
3. The role is created

A role is created to allow a user to perform web administration duties; it is called **Webadmin**. This role, with description, is added to the `/etc/rbac/roles` file. This file can be manually edited, or one can use the `roleadm` command to manage role-related information in the `roles`, `user_role`, and `role_auth` databases. The following is a sample entry to place in the `roles` file:

```
Webadmin: Role to administer web and servlet servers
```

4. Authorizations are created

The authorizations created in step 2 are placed in the `/etc/rbac/auths` file. This file can be manually edited, or the command `authadm(1M)` can be used to manage authorization information in the `auths`, `role_auth`, and `cmd_priv` databases. The following are sample entries placed in the `auths` file:

```
(hpux.hpws.webproxy.start,*)
(hpux.hpws.webproxy.stop,*)
(hpux.hpws.tomcat.start,*)
(hpux.hpws.tomcat.stop,*)
(hpux.hpws.ps,*)
```

5. Authorizations are assigned to the role

To associate the **Webadmin** role created in step 3 with the authorizations created in step 4, edit the `role_auth` file and associate the authorizations with the role:

```
Webadmin: (hpux.hpws.webproxy.start,*) (hpux.hpws.webproxy.stop,*)
(hpux.hpws.tomcat.start,*) (hpux.hpws.tomcat.stop,*) (hpux.hpws.ps,*)
```

The Webadmin role has all of the authorizations needed to administer the web related servers on the system. Different implementations may require an operator to be authorized to restart a server that stopped running, but not authorized to stop a server. RBAC allows the creation of roles with different levels of authorization to fine tune who may perform which action.

6. The role is assigned to users

To associate roles with users, a `user_role` entry is created. For example:
root: Administrator, Webadmin

This example is somewhat redundant in that root already has hpux.* by default, and has (by extension) hpux.hpws.*, but it shows that mutually exclusive authorization sets can be created such that root does not necessarily have authorization to execute any binary on the system.

7. The command privilege rules that utilize the authorizations are created

This example requires the ability to start each server in its own compartment and chrooted environment. All or most targets of the associated apachectl file must be supported, and the server must run as a non-root UID. A list of command privileges is needed for each server instance that will permit a user with the Webadmin role to securely start the server as an unprivileged user in the correct compartment. The entry that allows a user to start a server, wp-apache_server, in the ows_apache compartment might look like:

```
/opt/hpws/apache/bin/httpd:
-f /opt/hpws/apache/webproxy/servers/wp-apache/conf/httpd.conf
-k start: (hpux.hpws.webproxy.start,apache):105/105/58/58:
ows_apache:CHROOT,NETPRIVPORT:dlft:
```

In this example, if the user is assigned to a role with the hpux.hpws.webproxy.start authorization, the web server binary /opt/hpws/apache/bin/httpd is started with the /opt/hpws/apache/webproxy/servers/wp-apache/conf/httpd.conf configuration file running as uid=105 (owww) gid=58 (www) in the ows_apache compartment chrooted to /var/jail/wp_internet/.

The CHROOT privilege is required to enable the web server chroot to occur.

The NETPRIVPORT privilege is required to enable the web server to bind to a port lower than 1024.

If the second argument field of the cmd_priv file is not set to dlft, then it must be an exact string match of the command argument for the rule to match.

To invoke the httpd binary using the RBAC cmd_priv configuration, it is necessary to use the privrun(1M) command. For example, to start the wp-apache server instance from apachectl start, it is necessary to modify the apachectl script to prefix the httpd binary with privrun(1M):

```
/usr/bin/privrun -a hpux.hpws.webproxy.start,apache 
```
When a user implicitly executes the above `privrun` command via `apachectl`, the kernel checks the `cmd_priv` database to determine if the user has the RBAC authorization required to start the web server. A user assigned the `Webadmin` role will pass the RBAC authorization check and the web server will be started with the uid, gid, compartment, and privilege attributes found in the `cmd_priv` configuration. If the user fails the authorization check, the web server will not be started, rather an RBAC authorization error will be returned to the user.

**Virtualvault to Security Containment Transition**

**Web Server Architectures**

Extending the previous example of a chrooted web server isolated using compartments; the following sections show how various web server architectures commonly found on Virtualvault may be constructed on a Security Containment system using Compartments and Role Based Access Control.

Three web server architectures will be examined:

- Executing CGIs via the Inside Trusted Gateway Agent Daemon
- Proxying http requests via the Inside Webproxy server
- Executing Java servlets via the Inside Jakarta Tomcat servlet server

Each of the three architecture models uses an unprivileged non-root chrooted Outside Apache web server to isolate the hostile Outside Internet interface from the Inside server processes and Intranet interface. Should the outside server become compromised, the server is isolated, unprivileged, and lacking any method of gaining access to the Inside compartment and Intranet network.

**Apache/TGA on Virtualvault to Apache/CGI on Security Containment**

The Trusted Gateway Agent Daemon (tgad) was developed when Virtualvault utilized the Netscape Enterprise Server (NES) to service HTTP requests. The `tgad(1M)` is a privileged program designed to securely communicate across the Mandatory Access Control (MAC) boundary between Inside and Outside compartments of the Virtualvault system.

The tgad executes a CGI in the Inside compartment on behalf of the Outside server. In its original form, the NES executed a tga CGI stub program, and the stub program made contact with the tgad daemon process and passed along program parameters, environment
variables, etc. In its next form, the CGI stub program was replaced by a privileged NSAPI module that connected directly to the \texttt{tgad} daemon. Later, an Apache module, \texttt{mod_tga}, was developed when the Virtualvault moved from utilizing the NES to utilizing Apache as the bundled web server.

At the time that the original \texttt{tgad} was developed, the HTTP protocol and CGI execution were still developing technologies. Generations of web servers and HTTP protocols later, web servers are now able to handle HTTP request proxying, TCP connection persistence, forwarded headers, etc., in ways that were not defined in earlier versions.

For example, a CGI program in 1996 could generally expect that the HTTP HOST header would be the actual host (the Outside server) the HTTP client would have connected to. CGI programs written later recognize that one or more Internet proxies might exist between the client and the server that executes the CGI, and are coded to also look for a HOST header in the HTTP\_X headers. In the current Internet environment, CGI programs should handle the case where an HTTP request is proxied to another server for execution.

There are now many open source and commercial products that can accomplish the functionality of Apache/tgad, such as Apache/TomcatCGI.

A transition of the VV/tgad architecture to Security Containment can be made using the previously discussed webproxy architecture. Instead of the inside server being an intermediary between outside server and Intranet LAN, the inside server executes CGI programs on the inside server's runtime chroot using the included \texttt{mod_cgid} module. For this Apache/Apache CGI architecture, the inside server will have no connection to the inside LAN compartment. Name translation, formerly provided by the \texttt{tgad.conf} configuration file, will occur in the inside server and will be performed by the Apache Rewrite module/engine.

The following figure illustrates how two Apache web servers can be configured to securely execute CGI programs on Security Containment, followed by a table listing the compartment rules in effect:

\textbf{Figure 1:}  
\textit{Apache/CGI}
To enable CGI execution on the SSL virtual server, insert the following lines in the `<VirtualHost _default_:443>` section of the Apache `ssl.conf` file. The `ssl.conf` file is in the same directory as the `httpd.conf` file.

```
<VirtualHost _default_:443>
  RewriteEngine On
  RewriteRule ^/cgi-bin/(.*)$ http://127.0.0.1:11340/cgi-bin/$1 [P]
  ProxyPassReverse / http://127.0.0.1:11340/
...
</VirtualHost>
```

### Using MAC to restrict CGI execution

There may be cases where it is necessary to restrict web server access to CGI executables selectively. An example of this is when the application administration CGI resides in the same directory as the user access CGI. This functionality is achieved on Virtualvault by setting appropriate parameters in the tga configuration file (`tga.conf`). On SC, there are different ways to prevent a CGI from executing, such as modifying the web server configuration or changing DAC permissions. Security Containment compartment rules add another layer of security to restrict access to specific CGI executables. This is accomplished by adding a rule to the CGI web server compartment to deny access to the CGI executable file. For example, if access to CGI `mywebappadm` needs to be restricted, assuming that the CGI directory is `/var/jail/wp_cgi/cgibin`, the

| A | Grant server tcp port 80 out_iface |
| B | Grant client tcp peer port 8080 iws_cgi |
11. Web Server Transition

add the following line to the `/etc/cmpt/iws_cgi.rules` compartment rule file:

```
perm none /var/jail/wp_cgi/cgibin/mywebappadm
```

In order to pass client certificates to the CGI web server on a SC system, the Apache directives `ProxyClientCertificate` and `SSLOptions` should be set as follows:

```
ProxyClientCertificate On
SSLOptions +ExportCertData
```

For more information on these and other related directives see the `Webproxy Administrator’s Guide`, available at `/opt/hpws/apache/hpws_docs/webproxy.admin.guide.pdf`.

Apache/Webproxy on Virtualvault to Apache/Webproxy on Security Containment

Previously, HP-UX was geared towards a single Apache web server configuration with multiple virtual servers. Implementing webproxy functionality allowed for the easy and automated creation of multiple server instances, each executing in a separate chrooted environment. With Security Containment features, web servers executing in chrooted compartments achieve isolation similar to that of the Virtualvault web servers.

The following figure illustrates the Webproxy architecture on Security Containment, followed by a table listing the compartment rules in effect:

The following figure illustrates the Webproxy architecture on Security Containment, followed by a table listing the compartment rules in effect:

**Figure 2:** Apache/Webproxy

![Diagram of Apache/Webproxy architecture on Security Containment](image-url)
In this example, the Outside wp-apache web server is chrooted to /var/jail/wp.internet/. This web server listens only to \(<\text{out_iface}>:80\) and proxies to 127.0.0.1:8080. For the compartment configuration, the web server is a tcp server for port 80 of the out_iface compartment and is a tcp client to port 8080 of the iws_webproxy compartment.

The Inside wp-webproxy web server is chrooted to /var/jail/wp.webproxy/. The web server is a tcp server listening on port 127.0.0.1:8080 from the ows_apache compartment and a tcp client to port 80 to the in_iface compartment. Neither apache compartment has access to both interface compartments. The ows_apache compartment is completely isolated from the Intranet network while the iws_webproxy compartment is completely isolated from the hostile Internet network. The servers may communicate only over ports and protocols permitted by compartment rules and enforced by the kernel.

### Apache/Tomcat on Virtualvault to Apache/Tomcat on Security Containment

On Virtualvault, the Outside web server communicates with the user-installed Jakarta Tomcat servlet server by means of the mod_jk module. Mod_jk was specially built for Virtualvault to communicate across the mandatory access control boundary to a standard Tomcat servlet server listening in the Inside compartment.

The following figure illustrates the Apache/Tomcat architecture on Security Containment, followed by a table listing the compartment rules in effect:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Grant server tcp port 80 out_iface</td>
</tr>
<tr>
<td>B</td>
<td>Grant client tcp peer port 7080 iws_webproxy</td>
</tr>
<tr>
<td>C</td>
<td>Grant client tcp peer port 80 in_iface</td>
</tr>
</tbody>
</table>
This architecture maps directly to Security Containment. For Apache/Tomcat, apache mod_jk runs in its own Internet-facing compartment and communicates with Tomcat executing in the Inside compartment.

If a Tomcat server with a mod_jk connector was found on the Virtualvault system, then vvscassist.sh automatically enables an inside Tomcat server running in the in_tomcat compartment on the SC system. The in_tomcat compartment definition defaults as:

```plaintext
/* ***********************************************/
compartment in_tomcat {
  /* File System Rules */
  #include </etc/cmpt/inside.common>

  /* Tomcat file system rules */
  perm  read, write, create, unlink /opt/hpws/tomcat
  perm  read, write, create, unlink /opt/java1.5/

  /* IPC Rules */
  /*
  * Tomcat IPC Rules
  * Known Needed:
  * tomcat ajp13 channelsocket connector port  8009
  * put rules to connect to 8009 in client cmpt
  */
  /* tomcat standalone connector port     8081 */
  /* uncomment the following line to access the connector port
```
Post Transition Web Server Administration

HP has made the transition of web application architectures from Virtualvault to HPUX Security Containment easier by supplying enhancements to augment the HP Web server Suite webproxy scripts in order to create secure web server instances. These enhancements include:

- `/opt/hpvv/bin/vvws_createxsec`
  `vvws_createxsec` creates a secure web server instance. The script queries the administrator with a series of questions and then automatically generates the following:

  - A web server instance
  - A chroot runtime environment for the web server instance
  - A compartment definition
  - RBAC command privileges to start the server
  - A custom apachectl that uses the RBAC configuration

- `/opt/hpvv/bin/vvws_removexsec`
  `vvws_removexsec` removes a secure web server instance. The script accepts an argument of the serverID or queries the administrator for it, and then automatically removes the following for the specified server:

  - The web server instance
Creating a Secure Web Server Instance

Because `vvws_createxsec` creates a secure web server instance, it is important to know what it does and how it interacts with HPUX security components to create web server instances for VV transition architecture components.

`vvws_createxsec` performs the following actions to create a secure web server instance:

1. `vvws_createxsec` gathers information from the administrator. Specifically, it queries the administrator for the following:
   - **ServerID**: The ServerID is used to differentiate the web server instance on the file system. The server instance is created in the `/opt/hpws/apache/webproxy/servers/wp-<ServerID>/` directory. The ServerID may contain alphanumeric, dot, dash, or underscore characters.
   - **ServerName**: The ServerName or fully-qualified-domain-name[:port] that the server uses to identify itself. It will be placed in the ServerName directive of the apache configuration file. It may contain alphanumeric, dot, dash, underscore, or colon characters.
   - **HTTP port number**: The HTTP port number is the port number on which the server instance will listen. The port number will be placed in the Listen directive of the apache configuration file. If the server instance needs to listen on more than one port (for example, listening on several ports for separate virtual servers), the httpd.conf file can be edited later to add those Listen ports and servers.
   - **SSL port number**: The SSL port number is the port number on which the server will listen for SSL encrypted requests. The port number will be placed in the Listen directive of the apache configuration file. If the server will not be listening for SSL encrypted requests, entering 0 at this prompt signals the script to ignore further SSL operations.
   - **User**: User is a user account name. The user account must already be created and must exist in the `/etc/passwd` file. The User variable will be the user that the server will execute as at runtime. Further, the User variable will be used to set up ownership on files.
in both the web server instance directory and the server’s runtime chroot directory.

- **Group.** Group is an HPUX system group name. The group must already exist on the system. The Group variable will be the group that the server will execute as at runtime. Further, the Group variable will be used to set up ownership on files in both the web server instance directory and the server’s runtime chroot directory.

- **Chroot.** The Chroot variable is appended to /var/jail/wp_ to form a chroot runtime environment location. A web server runtime environment is created at the chroot location, and the Chroot Apache directive for this location is placed in the httpd.conf file. Multiple servers can share the same chroot environment.

- **Will the server listen to the Internet for connection requests?** Answer yes to add the compartment tcp server rules to communicate with the out_iface compartment.

- **Will this server communicate with an intranet-facing server for the purpose of proxying requests to the intranet LAN?** Answer yes to add the apache proxy rules and compartment tcp client rules to connect to the iws_webproxy server.

- **Will this server communicate with the CGI server for CGI program execution?** Answer yes to add the apache proxy rules and compartment tcp client rules to connect to the iws_cgi server.

- **Will this server communicate with a Tomcat servlet server for servlet execution?** Answer yes to add the compartment tcp client rules to connect to the in_tomcat server.

- **Will this server communicate through the INTRANET LAN for web (HTTP or HTTPS) replies or requests?** Answer yes to add the compartment rules to communicate with the in_iface compartment.

- **Will this server submit request to INTRANET LAN webservers or applications?** Answer yes to add the compartment tcp client rules to communicate with the in_iface compartment.

- **Will this server receive requests from INTRANET LAN webservers or applications?** Answer yes to add the compartment tcp server rules to communicate with the in_iface compartment.

2. **vvws_createxsec** creates the web server instance. Web server instances are created deterministically in /opt/hpws/apache/webproxy/servers. Servers created in this manner can be started and stopped using the standard /sbin/init.d scripts, and during startup and shutdown. Web
server instances are started via `apachectl` scripts in their server-specific directories. The `init.d` scripts traverse the `/opt/hpws/apache/webproxy/servers/wp-` directory, starting or stopping all servers found there.

3. `vvws_createxsec` creates the runtime chroot environment for the web server. The chroot runtime environment contains files necessary for apache server execution. All html documents and server CGI programs should be placed in the chroot environment of the server so that the server will have access to them at runtime.

4. `vvws_createxsec` configures RBAC to securely start and stop web servers. The `vvws_createxsec` script configures authorizations, a Webadmin role, and command privileges to allow users to start/stop web server instances. Two authorizations, `hpux.hpws.webproxy.start` and `hpux.hpws.webproxy.stop`, are created to start and stop web servers. Users with the newly created Webadmin role may use these authorizations. Finally, command privileges that match `apachectl` server invocations are placed in the `cmd_priv` file. By default, only the root account is granted the Webadmin role.

5. `vvws_createxsec` configures the `apachectl` script to utilize RBAC entries for start/stop. The web server instance `apachectl` files are modified to utilize the `privrun`(1M) command in combination with command privileges to perform web server start and stop activities. This ensures that the server starts in the correct compartment with the correct privileges.

6. `vvws_createxsec` creates a compartment definition with file system and communication rules for the web server instance. The compartment definition entries are constructed using knowledge of the server instance directory, server chroot directory, ports and purposes of the server that were input during the query section. While the compartment definition is created, it is not incorporated into the system by default. After execution of the `vvws_createxsec` script, the administrator must execute `setrules`(1M) to incorporate the new compartment definition into the kernel.

7. `vvws_createxsec` adds IPFilter rules to pass inbound tcp (HTTP) network communication from the out_iface or in_iface interface to the web server HTTP_PORT and SSL_PORT. While the IPFilter rules are added to the `ipf.conf` and Bastille’s `ipf.customrules` configuration files, they are not activated by default. After execution of the `vvws_createxsec` script, the administrator must load the new IPFilter rules into the kernel. For example:
Removing a Secure Web Server Instance

The `vvws_removeexec` script is provided to automate the removal of a secure web server instance. The script accepts an argument of the serverID or will prompt the administrator for the name.

`vvws_removeexec` syntax:

```
vvws_removeexec takes one optional argument:

vvws_removeexec serverID

- serverID: existing secure web server instance to be removed.
```

If the serverID is not provided, the script will prompt the administrator for the name. The servername is checked and if invalid, a list of valid serverIDs is provided. The script then carries out the following actions:

- Removes the web server instance
- Removes the associated chroot instance
- Removes the associated RBAC configuration
- Removes the associated compartment rules
- Runs `setrules` to activate compartment rule changes
- Removes the IPFilter HTTP_PORT and SSL_PORT rules
- Runs `ipf(8)` to activate the removal of IPFilter rules

Undoing the Actions of `vvscassist.sh`

If required, the actions taken by `vvscassist.sh` can be undone using the `vvscassist_cleanup.sh` script. The script is located in the `/opt/hpvv/vvscassist` directory and must be run as root.

The script removes the configuration changes made to the Security Containment system including:

- Apache Web Servers instances
- Apache Webproxy instances
- CGI Server instance
- Apache Chroot Instances
- Apache RBAC configuration
- Apache Compartment instances
- Apache IPFilter configuration
- VVSC Java Servlet configuration
- VVSC Audit configuration
- VVSC User and Group configuration
When run, the script warns the user about the actions to be taken and requests confirmation before continuing. The script logs execution details to \texttt{/opt/hpvv/vvscassist/vvscassist.log} for review.

**Beyond Virtualvault Web Server Security**

The architectures and methodologies explained in this chapter are designed to provide functionality and security similar to that of Virtualvault on the Security Containment system.

Security Containment allows for higher degrees of security. For example:

- Chroot Tomcat servlet servers
- Use of other HP security tools. For example, HP-UX Process Resources Manager (PRM). PRM has been made Security Containment aware. PRM provides kernel enforced system resource limits, a feature that was not available on Virtualvault.

These subjects are beyond the scope of this document. It is up to the Security Containment administrator to investigate further. A suggested source of information is the HP documents site:

\texttt{http://docs.hp.com/}
Appendix A: About Auditing

Overview

On a Virtualvault system, auditing is configured by selecting Audit Groups to monitor. An Audit Group consists of a set of system calls to monitor and a result that is either success or a certain type of failure.

Security Containment uses a different audit system. System call auditing is configured individually per system call. A system call can be recorded by the audit subsystem in case of success, failure or both.

The VV Transition toolkit uses the audit configuration from Virtualvault to configure the Security Containment audit subsystem in an equivalent manner.

For each Audit Group that was monitored on Virtualvault, the system calls involved and whether it was audited on success or failure is determined. Those system calls are audited for success or failure on the Security Containment system. This may result in some system calls generating audit records more often on Security Containment than on Virtualvault (especially on failure). However, any system call result which would have generated an audit record on Virtualvault should also generate one on Security Containment.
The Security Containment audit configuration is placed in 
```
/etc/rc.config.d/audit
```
and in general consists of three groups of system calls:

- Audited on success
- Audited on failure
- Audited on both success and failure

The flag to activate the audit subsystem on the Security Containment system is set based on whether the audit system on Virtualvault was active.

**Correspondence between Virtualvault Audit Groups and Security Containment system calls**

Below are tables listing Virtualvault Audit Groups and the set of equivalent Security Containment system calls. A given Audit Group monitors either successful or unsuccessful calls, but not both, so the tables are separated into success and failure groups.

<table>
<thead>
<tr>
<th>VVOS Audit Group</th>
<th>Security Containment System Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Object</td>
<td><code>accept</code> <code>bind</code> <code>creat</code> <code>link</code> <code>mkdir</code> <code>mknod</code> <code>msgget</code> <code>open</code> <code>pipe</code> <code>semget</code> <code>shmget</code> <code>socket</code> <code>socketpair</code> <code>symlink</code></td>
</tr>
<tr>
<td>Delete Object</td>
<td><code>mq_unlink</code> <code>msgctl</code> <code>rmdir</code> <code>semct1</code> <code>shmct1</code> <code>unlink</code></td>
</tr>
<tr>
<td>Discretionary Access Change</td>
<td><code>chmod</code> <code>chown</code> <code>fchmod</code> <code>fchown</code> <code>msgctl</code> <code>semct1</code> <code>shmct1</code></td>
</tr>
<tr>
<td>Interprocess Communication</td>
<td><code>kill</code> <code>msgrcv</code> <code>msgsnd</code> <code>semop</code></td>
</tr>
<tr>
<td>Make Object Available</td>
<td><code>bind</code> <code>connect</code> <code>creat</code> <code>modload</code> <code>mount</code> <code>mq_open</code> <code>open</code> <code>shmat</code></td>
</tr>
<tr>
<td>Make Object Unavailable</td>
<td><code>close</code> <code>fdetach</code> <code>moduload</code> <code>shmdt</code> <code>umount</code></td>
</tr>
<tr>
<td>Map Object to Subject</td>
<td><code>exec</code> <code>execve</code> <code>fstat</code> <code>fstat</code> <code>getdirent</code> <code>lstat</code> <code>modstat</code> <code>read</code> <code>readdir</code> <code>readv</code> <code>recv</code> <code>recvfrom</code> <code>recvmsg</code> <code>stat</code></td>
</tr>
<tr>
<td>Modify Object</td>
<td><code>creat</code> <code>ftruncat</code> <code>lockf</code> <code>modpath</code> <code>open</code> <code>rename</code> <code>send</code> <code>sendmsg</code> <code>sndto</code> <code>truncate</code> <code>utime</code> <code>write</code></td>
</tr>
<tr>
<td>Modify Subject</td>
<td><code>chdir</code> <code>chroot</code> <code>dup</code> <code>dup2</code> <code>fchdir</code> <code>fcntl</code> <code>setgid</code> <code>setgroups</code> <code>setuid</code></td>
</tr>
</tbody>
</table>
## Appendix A: About Auditing

### Categories of Unsuccessful System Calls:

<table>
<thead>
<tr>
<th>Category</th>
<th>VVOS Audit Group</th>
<th>Security Containment System Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Access Denial</strong></td>
<td>access acct bind chdir chmod creat exec execv execve fattach fchdir fchmod fchown fcntl fdetach fstat ftruncate getaccess getdirent getpriority ioctl kill link lockf lstat mkdir mknod modload mount mq_open mq_unlink msqctl msgget msgrcv msgsnd open pathconf pstat ptrace read readlink readdir rename rmdir semctl semget semop shmctl shmget stat statfs symlink truncate unlink utime write</td>
<td></td>
</tr>
<tr>
<td><strong>Insufficient Privilege</strong></td>
<td>acct adjtime chmod chroot clock_getres clock_settime exec execv execve sintime fchdir fchmod fchown getfh getpriority ioctl kill link lockf lstat mkdir modload modpath modstat moduload mount mptcl msqctl nice plock ptrace quotactl readlink reboot rename rtprio sched_setparam sched_setschedule semctl serialize setdomainname setgid setgroups sethostname setpriority setrlimit setsockopt settimeofday setuid shmctl stime swapon umount umount utime vfsmount</td>
<td></td>
</tr>
<tr>
<td><strong>Resource Denial</strong></td>
<td>accept connect creat dup dup2 exec execv execve fattach fcntl fork getfh link lockf mkdir modload mount mq_open msqctl msgsnd open pipe plock putmsg putpmsg rmdir semget semop sendfile shmctl shmget socket socketpair truncate unlink vfork vfsmount write writev</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B: HPUXHarden Configuration

The following is a sample listing of an HPUXHarden configuration file used by the hpuxharden.sh script.

```
#!/bin/sh

# $HPUXHARDEN_BASE set in calling script. Default value is set as:
#   HPUXHARDEN_BASE=/var/opt/hpuxharden

##### Strengthen Bastille lockdown configuration #####

The Bastille product is included in the initial system installation. Bastille provides three pre-configured environments:

- Sec10Host   HOST.config   Host-based lockdown, no firewall, telnet ftp ok
- Sec20MngDMZ MANDMZ.config Lockdown: IPFilter blocks all but mgmt sw
- Sec30DMZ    DMZ.config    Full lockdown: IPFilter blocks all but SSH

An even tighter Bastille lockdown configuration is recommended for the Virtualvault to Security Containment transition. This module starts with the DMZ.config as the base and applies the following modifications:

"Add" the following lines to the original Bastille DMZ.config:

- # Q: What should the minimum length of NEW passwords be?
  AccountSecurity.MIN_PASSWORD_LENGTH="8"
- # Q: Enter the password history depth.
  AccountSecurity.PASSWORD_HISTORY_DEPTH="4"
- # Q: Would you like to set a password history depth?
  AccountSecurity.PASSWORD_HISTORY_DEPTHyn="Y"
- # Q: Enter the maximum number of days between password changes:
  AccountSecurity.PASSWORD_MAXDAYS="182"
- # Q: Enter the minimum number of days between password changes.
  AccountSecurity.PASSWORD_MINDAYS="7"
- # Q: Enter number of days user will be warned their password will expire.
  AccountSecurity.PASSWORD_WARNDAYS="28"

"Modify" the following lines of Bastille DMZ.config:

- # Q: Do you want to setup password policies?
  AccountSecurity.passwordpolicies="Y"
- # Q: Would you like to password protect single-user mode?
  AccountSecurity.single_user_password="Y"
- # Q: Do you want basic system security auditing enabled?
  AccountSecurity.system_auditing="Y"
- # Q: What umask would you like to set for users on the system? [077]
  AccountSecurity.umask="077"
- # Q: Should Bastille scan for world-writeable directories?
  FilePermissions.world_writeable="Y"

"Prompt" the user the option to config the following lines:

- # Q: Should Bastille run Security Patch Check for you?
```
Appendix B: HPUXHarden Configuration

# Set up a cron job to run Security Patch Check?
# Patches.spc_cron_run="Y"  # was "N"
# Q:  Should Bastille set up a cron job to run Security Patch Check?
Patches.spc_cron_run="Y"  # was Patches.spc_cron_norun="Y"
# Q:  Does this machine require a proxy to ftp to the Internet?
Patches.spc_proxy_yn="Y"  # was "N"
# Q:  Please enter the URL for the web proxy.
Patches.spc_proxy="http://myproxy.mynet.com:8088"

Do you want to strengthen the Bastille lockdown configuration?

STRENGTHEN_BASTILLE_CONFIG=1  # 1=Yes  0=No

###################################################
##### Remove global chown privileges ######

HP-UX has a feature known as privilege groups, which is mechanism to assign a privilege to a group (see privgrp(4)). By default the CHOWN privilege is a global privilege and applies to all groups:

$ getprivgrp
global privileges: CHOWN

Non-privileged users don't need to be able to chown files to other users. /sbin/init.d/set_prvgrp is executed by default at system startup and executes the command /usr/sbin/setprivgrp -f /etc/privgroup if /etc/privgroup exists. A configuration file that deletes privileges for all groups (see setprivgrp(1m)) is created.

Do you want to remove global chown privileges?

REMOVE_GLOBAL_CHOWN_PRIVILEGES=1  # 1=Yes  0=No

###########################################
##### Remove unneeded pseudo-accounts ######

Accounts which neither own files nor are used to run processes should be removed. It is possible that some of these accounts may need to be recreated at a future time, and it is therefore worthwhile to save a copy of the original password and group files.

Remaining pseudo-accounts (adm, bin, daemon, sys, nobody, www), the login shell is changed to /usr/bin/false in /etc/passwd. Additionally, the account is locked by removing the "@" from the :u_lock@: field in the account's prpwd file.

Do you want to remove unneeded pseudo-accounts?

REMOVE_UNNEEDED_PSEUDO_ACCOUNTS=1  # 1=Yes  0=No

The following Users and Groups will be removed from the system. To prevent HPUXHarden from removing an entry, remove its name from the list below. Make sure there are no empty values between the first and the last value on each list.

USER1="lp"
USER2="hpdb"
USER3="uucp"
USER4="nuucp"
USER5=""
USER6=""
GROUP1="lp"
GROUP2="nuucp"
GROUP3=""
GROUP4=""
GROUP5=""
GROUP6=""

# The following pseudo-accounts will be altered as follows:
#   - The login shell will be changed to /usr/bin/false
#   - The account will be locked in prpwd(4)
#
# To prevent a pseudo-account from being modified, remove it from
# the list, preserving the double quotes.

PSEUDO_ACCOUNTS="adm bin daemon sys nobody www iwww owww"

##################################################################
##### Change root home directory to /root #####
# The root home directory is changed from the default of / to /root.
# The creates a private home directory for the root account to lessen
# the possibility of files being placed unintentionally in /.
# It also permits a restrictive mode to be placed on the directory.
# The root entry in /etc/passwd is modified to:
#   root:*:0:3::/root:/sbin/sh
#
# Do you want to change root home directory to /root?

CHANGE_ROOT_HOME_DIRECTORY=1     # 1=Yes  0=No

##################################################################
##### Configure nsswitch.conf(4) policy #####
# Virtualvault uses nsswitch.conf to search the local /etc/hosts file
# for hostname resolution prior to searching DNS. The nsswitch.conf file
# does not exist by default.
#
# The nsswitch.conf hosts: entry appears as follows:
#   hosts: files [NOTFOUND=continue TRYAGAIN=continue] dns
#
# Do you want to configure the nsswitch.conf(4) policy?

CONFIGURE_NSSWITCH_CONF_POLICY=1     # 1=Yes  0=No

##################################################################
##### Prevent syslogd from listening on the network #####
# syslogd has the -N option to prevent it from listening on the
# network for remote log messages. The initialization options are
# set in /etc/rc.config.d/syslogd.
#
# Do you want to prevent syslogd from listening on the network?

PREVENT_SYSLOGD_NETWORK_LISTENING=1     # 1=Yes  0=No

##################################################################
##### Disable rpcbind daemon #####
# rpcbind should be disabled on hardened systems. Even though rpcbind
is running on Virtualvault systems, it is effectively disabled from INSIDE and OUTSIDE network communications via Virtualvault MAC policy since it runs in the SYSTEM compartment. rpcbind may be re-enabled later if this system is managed via Openview Operations. rpcbind is disabled by not executing /sbin/init.d/nfs.core at startup. Additionally /usr/bin/rpcbind is renamed to rpcbind.DISABLE.

Do you want to disable rpcbind daemon?

DISABLE_RPCBIND_DAEMON=1 # 1=Yes 0=No

#################################################################
##### Disable console logging #####
# Console logging should be disabled. This prevents the console message formatting daemon from being started. nettlconf(1m) command is used to disable console logging.
# Do you want to disable console logging?
DISABLE_CONSOLE_LOGGING=1 # 1=Yes 0=No

##################################################################
##### Disable pwgrd daemon #####
# pwgrd is a password and group caching daemon. Since there are few users and groups on the system, it is unnecessary.
# Bastille disables pwgrd by setting PWGR=0 in /etc/rc.config.d/pwgr. However, pwgrd also housekeeps the /var/spool/sockets/pwgr/ directory. Over time, without pwgrd garbage collection, the directory will grow to many thousands of stale unused sockets. The solution is to remove the pwgrd status file and spool directory.
# rm -f /var/spool/pwgr/status
# rm -f /var/spool/sockets/pwgr/*
# If it is decided at a later time that pwgrd service is desired, pwgrd will simply recreate the status file and spool directory automatically the first time it is started.
# Do you want to disable pwgrd daemon?
DISABLE_PWGRD_DAEMON=1 # 1=Yes 0=No

##################################################################
##### Examine Set-id Programs #####
# Many Unix systems, including HP-UX, ship with numerous programs that are set-uid or set-gid. Many of these programs are not used or are only used by the root user. Many of the vulnerabilities that are discovered in Unix utilities rely on the set-uid root bit to raise privilege. System security can be improved by removing these programs or by removing the set-id bit. To obtain a list of all files with either the set-uid or set-gid bit set on the system (and save the list to a file for future reference):
# find / ( -perm -4000 -o -perm -2000 ) -type f -exec ls -ld {} \; >> logfile
# The strategy is to remove the set-id bits from all files, then selectively
# add it back to just a few programs that need to be run by non-root users.
# The following commands will remove the set-uid set-gid bits from all files.
#   find / -perm -4000 -type f -exec chmod u-s {} \;
#   find / -perm -2000 -type f -exec chmod g-s {} \;
# If it is determined at a later time that the stripped setuid or setgid bit
# of a program is in fact needed; please review the logfile:
#   ${HPUXHARDEN_BASE}/prehardenperms.log
# to examine the original mode bit value. Alternatively, use chmod u+s g+s
# as appropriate to replace the setuid/setgid bit as is illustrated above.
# Do you want to examine Set-id Programs?

EXAMINE_SETID_PROGRAMS=1     # 1=Yes  0=No

if [ $EXAMINE_SETID_PROGRAMS -eq 1 ]; then
  # Some programs are known to require the setuid or setgid bit for successful
  # operation on VVSC systems. The set-id bit will therefore be added back.
  # To prevent HPUXHarden from restoring the set-id bit, remove the program
  # name from the appropriate list below. Likewise, to restore the set-id bit
  # for additional programs, simply add the program name to the appropriate
  # list below. In any event, use care to preserve the surrounding double quotes.

  RESTORE_SETUID_BIT="
  /usr/bin/su
  /usr/bin/passwd
  /usr/sam/1bin/rsam
  /usr/sbin/swlist
  /opt/ids/lbin/idssysdsp
  /opt/ids/lbin/updaterc
  /usr/bin/crontab
  /usr/bin/privedit
  /usr/bin/privrun
  /usr/sbin/authadm
  /usr/sbin/cmdprivadm
  /usr/sbin/roleadm
  "

  RESTORE_SETGID_BIT="
  fi

########################################################################
##### Examine File Permissions #####
########################################################################

#### Examine File Permissions ####

# A freshly installed HP-UX system will contain a number of files which are
# writable by other (the 002 bit is set in the mode bits). These files can
# be listed with the following (which also captures the output to a file
# for future reference):
#   find / -perm -002 ! -type l -exec ls -ld {} \; >> logfile
# Symbolic links with the OTHER_WRITE bit set need not be listed because
# the mode bits on the link are not used for permission checking.
# The preferred approach is to remove the OTHER_WRITE bit from all files,
# and then selectively add it back to those files and directories where it
# is necessary. The following can be executed to remove the OTHER_WRITE bit
# from all files with it set:
find / -perm -002 ! -type l -exec chmod o-w {} \;

If it is determined at a later time that the stripped permissions are in fact needed; please consult the logfile:

${HPUXHARDEN_BASE}/prehardenperms.log
to examine the original mode bit value. Alternatively, use chmod as appropriate to replace the permissions as is illustrated above.

Do you want to examine File Permissions?

EXAMINE_FILE_PERMISSIONS=1     # 1=Yes  0=No

###############################################

#### Remove Software Filesets ####

It is recommended that the following software, if present, be removed from Security Containment systems.

CIFS-Client          CIFS Client
CIFS-Development     HP CIFS Server Source Code Files
CIFS-Server          HP CIFS Server (Samba) File and Print Services
IntegratedLogin      Integrated Login Product
MOZILLAsrc           Mozilla 1.4 Source distribution
MySQL                MySQL open-source database
ParMgr               Partition Manager - HP-UX
PrinterMgmt          PrinterMgmt
SystemComm           System Comm. utilities - ct, cu, ptydaemon, vt, kermit
TechPrintServ        HP-UX Technical Image Printing Service
TerminalMngr         TerminalMngr
UUCP                  Unix to Unix CoPy

Do you want to remove the software filesets?

REMOVE_SOFTWARE_FILESETS=1     # 1=Yes  0=No

if [ $REMOVE_SOFTWARE_FILESETS -eq 1 ]; then

To prevent HPUXHarden from removing one of these entries, remove it from the list, preserving the double quotes.

SWREMOVE_LIST="
CIFS-Client
CIFS-Development
CIFS-Server
IntegratedLogin
MOZILLAsrc
MySQL
ParMgr
PrinterMgmt
SystemComm
TechPrintServ
TerminalMngr
UUCP
"
fi

###############################################
Appendix C: VVSnapshot Configuration

The following is a sample listing of a VVSnapshot configuration file used by the `vvsnapshot.sh` script.

```
#!/bin/sh
# (C) Copyright 2004-2005 Hewlett-Packard Development Company, LP
#
# VVSnapshot is a utility to assist with transitioning from Virtualvault to the Security Containment system. VVSnapshot runs on the Virtualvault system, performing the following tasks:
# - Scans functional areas of the system (see modules below for details)
# - Writes configuration details to the VVSnapshot Report file
# - Copies configuration files and directories to the Staging Directory
# - Creates the Staging Archive to be transferred to the target Security Containment system
#
# VVSnapshot uses the following files:
#
# Configuration File: vvsnapshot.config
# Script File: vvsnapshot.sh
# Log File: vvsnapshot.log
# Report File: vvsnapshot.report
# Staging Directory: vvsnapshot_staging/
# Staging Archive: vvsnapshot_staging.tar
#
# Configuration File - vvsnapshot.config
# The configuration file (this file) accompanies the VVSnapshot script. Script module control variables and default environment variables are configurable by the user. The default values are the HP recommended settings.
#
# Script File - vvsnapshot.sh
# VVSnapshot is executed on the Virtualvault system by the root user as:
# Usage: vvsnapshot.sh [-v]
# -v (verbose) option to display execution comments on the terminal screen. Execution comments are written to the Log file regardless of the -v option.
#
# Log File - vvsnapshot.log
# All VVSnapshot input and output (stdin, stdout, stderr) is recorded in the VVSnapshot Log File. Since the log data is appended to the end of the file, the log file therefore preserves the chronological history of all executions.
#
# Report File - vvsnapshot.report
```
The Report File captures the configuration details of the functional areas scanned on the Virtualvault system. The report file will be included in the Staging Archive that will be transferred to the Security Containment system. It will be used along with the Staging Directory as input for Security Containment-side configuration processing.

Each of the VVSnapshot modules writes configuration details to the report file. See the individual modules below for the details recorded by each module.

Staging Directory - vvsnapshot_staging/

Many configuration files and directories of the areas scanned are copied to the Staging Directory. Upon completion, it will be archived (tar) and copied to the Security Containment system. It will be used along with the Report File as input for Security Containment-side configuration processing.

Staging Archive - vvsnapshot_staging.tar

The Staging Archive is simply a tar archive of the Staging Directory and Report File that will be transferred to the Security Containment system to be used for Security Containment-side configuration processing. VVSnapshop by default creates a disk archive, however the user may choose to archive to a tape device (e.g. /dev/rmt/0m) instead.

VVSnapshot Modules:

VVSnapshot scans the following functional areas:

- Boot Time Configuration
- Kernel Tunables Configuration
- Security Databases Configuration
- Users and Groups Configuration
- Secure Shell SSH Configuration
- Filesystem Configuration
- Network Configuration
- Audit Files Configuration
- Cron Files Configuration
- Openview VPO Client Configuration
- Web Server Configuration
- Webproxy Configuration
- Java Servlet Configuration
- Trusted Gateway TGA Configuration
- Trusted Proxy TGP Configuration
- Trusted IPC Configuration
- Chroot Configuration

VVSnapshot scans each functional area as a separate module. Control variables allow any module to be optionally disabled. Documentation for each module, followed by a control variable, can be found below.

Boot Time Configuration

The Boot Time Configuration module performs the following tasks:

VVSnapshot Report captures non-default entries from:
- /etc/inittab
- /etc/rc.config.d/
- /sbin/init.d/

Appendix C: VVSnapshot Configuration

# Staging Directory captures: All files above
#
# Do you want to execute Boot Time Configuration tasks?

BOOT_TIME_CONFIGURATION=1     # 1=Yes  0=No

# The following /etc/inittab entries are the default entries released by # HP and are not recorded by VVSnapshot. To have VVV Snapshot record one of # these entries, remove it from the list, preserving the double quotes.

DEFAULT_1104_INITTAB="
init secd audt ioine muix brcl link crpr sqnc # powf iaws
tgad tgpdp ows prxy px2 cons # ttp1 # ttp2 # ttp3 # ttp4 # ttp5 # ups alrm
"

# The following /etc/rc.config.d/ entries are the default entries released by # HP and are not recorded by VVSnapshot. To have VVV Snapshot record one of # these entries, remove it from the list, preserving the double quotes.

DEFAULT_1104_ETC_RCCONFIGD="
LANG Rpcd SnmpHpunix SnmpMaster SnmpMib2 SnmpTrpDst SnmpVaultApache
acct cde clean clean_tmps crashconf cron dce desktop envd hparray
hpbase100conf hpetherconf hpfcgsc_lanconf hpfcmsconf hppci100conf
iforis kmninit list_mode mailsvrs namesvrs nddconf netconf netdaemon
nett1 nscconf orcramd proxy proxy2 ptydaemon pwgr savecrash set_date
ssl sshd supprinfo svconfig syncer vvmgrhttpd xfs xfs
"

# The following /sbin/init.d/ entries are the default entries released by # HP and are not recorded by VVSnapshot. To have VVV Snapshot record one of # these entries, remove it from the list, preserving the double quotes.

DEFAULT_1104_SBIN_INITD="
OspFmib Rpcd SnmpHpunix SnmpMaster SnmpMib2 SnmpTrpDst SnmpVaultApache
cleanadm clean_ex clean_tmps crashconf cron dce dtlogin/rc envd gated
hostname hparray hpbase100 hpether hpfc hpfcms hppci100 iforis inetd
inetsvcs iwshtpd killall kmbuild kmninit localmount maclan_init mrouted
named net net.define nscf nscf service owsdm owsdm svc proxy proxy2
ptydaemon pwgr rarpd rbootd rdpdp rwhod savecrash sendmail
set_date set_prvgrp ssla sps supprinfo svagent svconfig
svconfig syslogd template vvmgrhttpd vvos.rc xfs xfs xntpd
"

### Kernel Tunables Configuration

# The Kernel Tunables Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
#  - /stand/system Tunables
#  - System Definition (sysdef)
#
# Staging Directory captures: N/A
#
# Do you want to execute Kernel Tunables Configuration tasks?

KERNEL_TUNABLES_CONFIGURATION=1     # 1=Yes  0=No

### Security Databases Configuration

# The Security Databases Configuration module performs the following tasks:
# VVSnapshot Report captures: N/A
#
# Staging Directory captures:
#  - /etc/auth/system/
#  - /etc/auth/subsystems/
#  - /tcb/files/auth/
#
# Do you want to execute Security Databases Configuration tasks?
SECURITY_DATABASES_CONFIGURATION=1     # 1=Yes  0=No
#
# The following Security Database directories are recorded by VVSnapshot
DEFAULT_1104_SEC_DB_DIRS="
/etc/auth/system
/etc/auth/subsystems
/tcb/files/auth
"

###################################################################
## Users Groups Configuration                                    ##
###################################################################
# The Users and Groups Configuration module performs the following tasks:
# # VVSnapshot Report captures non-default entries to:
# #  - Users:
# #    - /etc/passwd
# #    - /tcb/files/auth/*/<user prpwd>
# #    - /home/<user>/
# #  - Groups:
# #    - /etc/group
# # Staging Directory captures: All files above
# # Do you want to execute Users Groups Configuration tasks?
USERS_GROUPS_CONFIGURATION=1     # 1=Yes  0=No
#
# The following /etc/passwd entries are the default entries released by
# HP and are not recorded by VVSnapshot. To have VVSnapshot record one of
# these entries, remove it from the list, preserving the double quotes.
DEFAULT_1104_USERS="
root daemon bin sys adm uucp lp nuucp audit auth cron
tcb www suser owww iwww tgp tga java
"
#
# The following /etc/group entries are the default entries released by
# HP and are not recorded by VVSnapshot. To have VVSnapshot record one of
# these entries, remove it from the list, preserving the double quotes.
DEFAULT_1104_GROUPS="
root other bin sys adm daemon mail lp tty nuucp users audit auth
backup cron mem tcb terminal nogroup swadm
"

###################################################################
## Secure Shell SSH Configuration                               ##
###################################################################
# The Secure Shell SSH Configuration module performs the following tasks:
Appendix C: VVSnapshot Configuration

# VVSnapshot Report captures:
# - Host-level SSH configuration:
#   - /opt/vaultTS/tools/OpenSSH/etc/
# - User-level SSH configuration:
#   - /home/<user>/apphome/.ssh/
# - NOTE comparing SSH on Virtualvault vs. Security Containment system
#
# Staging Directory captures: All files above
#
# Do you want to execute SSH Configuration tasks?

SSH.Configuration=1  # 1=Yes  0=No

#######################################
## Filesystem Configuration          ##
#######################################

# The Filesystem Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - Disk Free report (bdf)
# - LVM Volume Group configuration (vgdisplay --v)
# - DAC configuration:
#   - User Group Permissions  (ll -R / > ll.out)
#   - Access Control Lists    (lsacl -R / > lsacl.out)
# - MAC configuration:
#   - Sensitivity Levels      (lslevel -R / > lslevel.out)
#   - Privileges          (lspriv -R / > lspriv.out)
#
# Staging Directory captures:
# - ll.out
# - lsacl.out
# - lslevel.out
# - lspriv.out
#
# Do you want to execute Filesystem Configuration tasks?

FILESYSTEM.Configuration=1  # 1=Yes  0=No

#######################################
## Network Configuration             ##
#######################################

# The Network Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - /etc/hosts contents
# - /etc/nsswitch.conf contents
# - /etc/resolv.conf contents
# - /etc/rc.config.d/netconf contents
# - /etc/rc.config.d/nddconf contents
# - Network Interface Cards (NICs) configuration
#   - lanscan
#   - ifconfig <NICs>
# - Network Routing Table configuration
#   - netstat -rn
# - Network Listening Endpoints
#   - netstat -an | grep LISTEN
#
# Staging Directory captures:
# - /etc/services
# - /etc/nsswitch.conf
# - /etc/resolv.conf
# - /etc/rc.config.d/netconf
# - /etc/rc.config.d/nddconf
#
# Do you want to execute Network Configuration tasks?

NETWORK_CONFIGURATION=1 # 1=Yes 0=No

# The following Network files are recorded by VVSnapshot. To prevent # VVSnapshot from recording one of these entries, remove it from the list, # preserving the double quotes.

DEFAULT_1104_NETWORK_FILES="
/etc/hosts
/etc/services
/etc/nsswitch.conf
/etc/resolv.conf
/etc/rc.config.d/netconf
/etc/rc.config.d/nddconf
"

#######################################
## Audit Files Configuration         ##
#######################################
# The Audit Files Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - Audit subsystem statistics (auditcmd -c)
# - Audit Events enabled
# - Alarm configuration (alarm_defs)
#
# Staging Directory captures: N/A
#
# Do you want to execute Audit Files Configuration tasks?

AUDIT_FILES_CONFIGURATION=1 # 1=Yes 0=No

#######################################
## Cron Files Configuration          ##
#######################################
# The Cron Files Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - /var/adm/cron/at.allow
# - /var/adm/cron/cron.allow
# - List all at jobs scheduled:
#   - at -1 <all_users>
# - List all crontabs scheduled:
#   - crontabs -l <all_users>
#
# Staging Directory captures:
# - /var/adm/cron/at.allow
# - /var/adm/cron/cron.allow
# - /var/spool/cron/crontabs/
# - /var/spool/cron/atjobs/
#
# Do you want to execute Cron Files Configuration tasks?

CRON_FILES_CONFIGURATION=1 # 1=Yes 0=No

# The following Cron files are recorded by VVSnapshot. To prevent # VVSnapshot from recording one of these entries, remove it from the list, # preserving the double quotes.
DEFAULT_1104_CRON_FILES="
/var/adm/cron/at.allow
/var/adm/cron/cron.allow"

# The following Cron directories are recorded by VVSnapshot. To prevent
# VVSnapshot from recording one of these entries, remove it from the list,
# preserving the double quotes.

DEFAULT_1104_CRON_DIRS="
/var/spool/cron/atjobs
/var/spool/cron/crontabs"

#########################################################################
## Openview VPO Client Configuration ##
#########################################################################

# The Openview VPO Client Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - ITOAgent Software version
# - /var/opt/OV/conf/OpC/nodeinfo
#
# Staging Directory captures:
# - /var/opt/OV/
#
# Do you want to execute VPO Client Configuration tasks?

VPO_CLIENT_CONFIGURATION=1    # 1=Yes  0=No

#########################################################################
## Web Server Configuration          ##
#########################################################################

# The Web Server Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - Instances summary
# - Instance <ws-instance> configuration:
#   - Apache version
#   - conf/ contents
#   - httpd.conf contents
#   - ssl.conf contents (Apache 2.0 only)
#   - SSL server certificate contents
#
# Staging Directory captures:
# - /opt/vaultWS/servers/<ws-instance>/conf/
#
# Do you want to execute Web Server Configuration tasks?

WEB_SERVER_CONFIGURATION=1    # 1=Yes  0=No

#########################################################################
## Webproxy Configuration            ##
#########################################################################

# The Webproxy Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - Instance configuration:
#   - Apache version
#   - conf/ contents
# httpd.conf contents
# ssl.conf contents (Apache 2.0 only)
# SSL server certificate contents
#
# Staging Directory captures:
# /opt/vvproxy/conf/
# /opt/vvproxy2/conf/ (Virtualvault 4.7 only)
#
# Do you want to execute Webproxy Configuration tasks?
WEBPROXY_CONFIGURATION=1     # 1=Yes  0=No

#######################################
## Java Servlet Configuration        ##
#######################################

# The Java Servlet Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - Java SDK or RTE version
# - Java Servlet (Tomcat) configuration:
#  - conf/ contents
#  - webapps/ contents
#  - Web server mod_jk instance: <ws-instance>
#  - mod_jk.conf-<ws-instance> contents
#  - workers.properties contents
#   - Java Servlet (Jserv) configuration
#
# Staging Directory captures:
# /var/opt/vaultTS/inside/app/servlet/
# /var/opt/vaultTS/inside/app/tomcat/
#
# Do you want to execute Java Servlet Configuration tasks?
JAVA_SERVLET_CONFIGURATION=1     # 1=Yes  0=No

# The following Java Servlet directories are recorded by VVSnapshot. To prevent
# VVSnapshot from recording one of these entries, remove it from the list,
# preserving the double quotes.
DEFAULT_1104_SERVLET_DIRS="
/var/opt/vaultTS/inside/app/servlet
/var/opt/vaultTS/inside/app/tomcat
"

#######################################
## Trusted Gateway TGA Configuration ##
#######################################

# The Trusted Gateway TGA Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
# - tgd.conf contents
# - CGI directory contents
#
# Staging Directory captures: All files above
#
# Do you want to execute TGA Configuration tasks?
TGA_CONFIGURATION=1     # 1=Yes  0=No

#######################################
## Trusted Proxy TGP Configuration   ##
#######################################


# The Trusted Proxy TGP Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
#   - tgp.conf contents
#
# Staging Directory captures:
#   - tgp.conf
#
# Do you want to execute TGP Configuration tasks?

TGP_CONFIGURATION=1     # 1=Yes  0=No

#######################################
## Trusted IPC Configuration         ##
#######################################
# The Trusted IPC Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
#   - Executables with Trusted IPC privilege:
#     - Server privileges: netmultilevelserver, netsetid
#     - Client privilege: netprivsession
#     - SVIPC Facilities (ipcs)
#   - SVIPC Facilities (ipcs)
#
# Staging Directory captures: N/A
#
# Do you want to execute Trusted IPC Configuration tasks?

TRUSTED_IPC_CONFIGURATION=1     # 1=Yes  0=No

# VVSnapshot looks for user applications with Trusted IPC privileges.
# The following executables are given Trusted IPC privilege by HP and are
# not recorded by VVSnapshot. To have VVSnapshot record one of these entries,
# remove it from the list, preserving the double quotes.

DEFAULT_1104_TRUSTED_IPC_FILES="
/opt/vaultTS/ws-admserv/bin/httpd
/opt/vaultWS/bin/tcwdog
/opt/vaultWS/install/bin/httpd
/opt/vaultWS/install/libexec/libhttpd.ep
/opt/vaultWS/install2/bin/httpd
/opt/vvproxy/bin/httpd
/opt/vvproxy/libexec/libhttpd.ep
/opt/vvproxy2/bin/httpd
/sbin/privs
/tcb/lib/tgcd
/usr/bin/X11/X
/usr/bin/netstat
/usr/bin/grmd
/usr/sbin/cron
/var/opt/vaultTS/outside/app/cgibin/tga
"

#######################################
## Chroot Configuration              ##
#######################################
# The Chroot Configuration module performs the following tasks:
#
# VVSnapshot Report captures:
#   - NOTE comparing chroot locations on Virtualvault vs. Security Containment
#
# Staging Directory captures:
# - Web Server Chroot directories parsed from <ws-instance> httpd.conf:
#   - Chroot /var/opt/vaultTS/outside/app
#   - Webproxy Chroot directories parsed from httpd.conf:
#     - Chroot /var/opt/vvproxy
#   - TGA Chroot directories parsed from tgad.conf:
#     - gw_root=/var/opt/vaultTS/inside/app
#
# Do you want to execute Chroot Configuration tasks?

CHROOT_CONFIGURATION=1     # 1=Yes  0=No

# The following Chroot directories if present are recorded by VVSnapshot.
# To prevent VVSnapshot from recording one of these entries, remove it from
# the list, preserving the double quotes.

DEFAULT_1104_CHROOT_DIRS="
/var/opt/vaultTS/outside/app
/var/opt/vaultTS/inside/app
/var/opt/vvproxy
"

******************************************************************************
# The following code enforces module dependencies. Do not modify!
if [ $WEB_SERVER_CONFIGURATION -eq 1 -o $WEBPROXY_CONFIGURATION -eq 1 -o 
   $TGA_CONFIGURATION -eq 1 ]; then
   CHROOT_CONFIGURATION=1
fi

if [ $TRUSTED_IPC_CONFIGURATION -eq 1 ]; then
   FILESYSTEM_CONFIGURATION=1
fi

******************************************************************************
Appendix D: VVSCAssist Configuration

The following is a sample listing of a VVSCAssist configuration file used by the vvscassist.sh script.

```bash
#!/bin/sh
# (C) Copyright 2004-2005 Hewlett-Packard Development Company, LP

VVSCAssist is part of the set of tools provided by HP to transition from Virtualvault to Security Containment. VVSCAssist uses the output of VVSnapshot to provide assistance in setting up the HP-UX Security Containment system to replace an existing Virtualvault system.
VVSCAssist runs on the Security Containment system.
VVSCAssist examines the VVSnapshot report file and staging directory and uses the information to configure Security Containment and creates a manual TODO list for the user to subsequently follow.
VVSCAssist uses the following files:

VVSCAssist Configuration File: vvscassist.config
VVSCAssist Script File: vvscassist.sh
VVSCAssist Log File: vvscassist.log
VVSCAssist Todo File: vvscassist.todo

VVSnapshot Report File: vvsnapshot.report
VVSnapshot Staging Directory: vvsnapshot_staging/
VVSnapshot Staging Archive: vvsnapshot_staging.tar

VVSCAssist Configuration File - vvsnapshot.config

The configuration file (this file) accompanies the VVSCAssist script. While some modules of VVSCAssist configure the system to look and behave as Virtualvault was configured, other modules add steps to the TODO list for the user to perform manually. If a directive in this file is set to "0", the module will not be executed, even if all it does is add steps to the TODO list. In that case the TODO list will not be complete. The default "1" directive values are the recommended settings and should not be changed unless directed by HP.

VVSCAssist Script File - vvscassist.sh

VVSCAssist is executed on the Security Containment system by the root user as:

Usage: vvscassist.sh [-v]
```
-v (verbose) option to display execution comments on the terminal screen. Execution comments are written to the Log file regardless of the -v option.

VVSCAssist Log File - vvscassist.log

All VVSCAssist input and output (stdin, stdout, stderr) is recorded in the VVSCAssist Log File. Since the log data is appended to the end of the file, the log file therefore preserves the chronological history of all executions.

VVSCAssist Todo File - vvscassist.todo

VVSCAssist automates many configuration steps although not everything can be automated. Configuration steps requiring manual administration are recorded in the TODO checklist file. After VVSCAssist completes, the user should examine the TODO file and use it as a guide to further configure the Security Containment system.

VVSnapshot Report File - vvsnapshot.report

The Report File was generated by VVSnapshot on the Virtualvault system. The report file contains configuration details of the functional areas scanned on the Virtualvault system. It is included in the Staging Archive that is transferred to the Security Containment system. It is used along with the Staging Directory as input for VVSCAssist.

VVSnapshot Staging Directory - vvsnapshot_staging/

The Staging Directory was generated by VVSnapshot on the Virtualvault system. Configuration files and directories of the areas scanned were copied to the staging directory. Upon completion, it was archived (tar) to be copied to the Security Containment system. It is used with the Report File as input for VVSCAssist.

VVSnapshot Staging Archive - vvsnapshot_staging.tar

The Staging Archive is a tar archive of the Staging Directory and Report File that is transferred to the Security Containment system. The archive by default is written to a disk file although it may be archived to a tape device if desired.

VVSCAssist Modules:

There is a corresponding VVSCAssist module for each VVSnapshot module, although the modules do not execute in the same order:

- Boot Time Configuration
- Kernel Tunables Configuration
- Security Databases Configuration
- Users Groups Configuration
- Secure Shell SSH Configuration
- Filesystem Configuration
- Network Configuration
- Audit Files Configuration
- Cron Files Configuration
- Openview VPO Client Configuration
- Common Gateway CGI Configuration
- Java Servlet Configuration
- Webproxy Configuration
- Web Server Configuration
- Chroot Configuration
- Trusted Proxy TGP Configuration
Appendix D: VVSCAssist Configuration

- Trusted IPC Configuration

NOTE: The VVSnapshot Trusted Gateway TGA Configuration module is replaced in VVSCAssist by the Common Gateway CGI Configuration module

Further details of each module are documented within each module below.

Boot Time Configuration

VVSnapshot Report contains non-default entries to:
- /etc/inittab
- /etc/rc.config.d/
- /sbin/init.d/

Staging Directory contains: All files above

The VVSCAssist Boot Time Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute Boot Time Configuration tasks?

KERNEL_TUNABLES_CONFIGURATION=1  # 1=Yes  0=No

Kernel Tunables Configuration

VVSnapshot Report contains:
- /stand/system Tunables
- System Definition (sysdef)

Staging Directory contains: N/A

The VVSCAssist Kernel Tunables Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute Kernel Tunables Configuration tasks?

Security Databases Configuration

VVSnapshot Report contains: N/A

Staging Directory contains:
- /etc/auth/system/
- /etc/auth/subsystems/
- /tcb/files/auth/

The VVSCAssist Security Databases Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute Security Databases Configuration tasks?
SECURITY_DATABASES_CONFIGURATION=0  # 1=Yes  0=No

#######################################
## Users Groups Configuration        ##
#######################################
# VVSnapshot Report contains non-default entries to:
#   - Users:
#     - /etc/passwd
#     - /tcb/files/auth/*/<user prpwd>
#     - /home/<user>/
#   - Groups:
#     - /etc/group
#
# Staging Directory contains: All files above
#
# The VVSCAssist Users Groups Configuration module performs the
# following tasks:
#   - Create non-default groups found in the VV staging area
#   - Create non-default users found in the VV staging area
#   - Copy each user’s VV home directory contents into place
#   - Add entries to the TODO checklist file
#
# Do you want to execute Users Groups Configuration tasks?

USERS_GROUPS_CONFIGURATION=1  # 1=Yes  0=No

#######################################
## Secure Shell SSH Configuration    ##
#######################################
# VVSnapshot Report contains:
#   - Host-level SSH configuration:
#       /opt/vaultTS/tools/OpenSSH/etc/
#   - User-level SSH configuration:
#       /home/<user>/apphome/.ssh/
#   - A note comparing SSH on Virtualvault vs. Security Containment system
#
# The VVSCAssist Secure Shell SSH Configuration module performs the
# following tasks:
#   - Confirm host-level SSH configuration
#   - Copy user-level SSH configuration to user's home directory
#   - Add entries to the TODO checklist file
#
# Staging Directory contains: All files above
#
# Do you want to execute SSH Configuration tasks?

SSH_CONFIGURATION=1  # 1=Yes  0=No

#######################################
## Filesystem Configuration          ##
#######################################
# VVSnapshot Report contains:
#   - Disk Free report (bdf)
#   - LVM Volume Group configuration (vgdisplay --v)
#   - DAC configuration:
#     - User Group Permissions  (ll -R / > ll.out)
#     - Access Control Lists    (lsacl -R / > lsacl.out)
Appendix D: VVSCAssist Configuration

- MAC configuration:
  - Sensitivity Levels  
  - Privileges

Staging Directory contains:
- ll.out
- lsacl.out
- lslevel.out
- lspriv.out

The VVSCAssist Filesystem Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute Filesystem Configuration tasks?
FILESYSTEM_CONFIGURATION=1  # 1=Yes  0=No

Network Configuration

VVSnapshot Report contains:
- /etc/hosts contents
- /etc/nsswitch.conf contents
- /etc/resolv.conf contents
- /etc/rc.config.d/netconf contents
- /etc/rc.config.d/nddconf contents
- Network Interface Cards (NICs) configuration
  - lanscan
  - ifconfig <NICs>
- Network Routing Table configuration
  - netstat -rn
- Network Listening Endpoints
  - netstat -an | grep LISTEN

Staging Directory contains:
- /etc/hosts
- /etc/services
- /etc/nsswitch.conf
- /etc/resolv.conf
- /etc/rc.config.d/netconf
- /etc/rc.config.d/nddconf

The VVSCAssist Network Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute Network Configuration tasks?
NETWORKConfigurationException=1  # 1=Yes  0=No

Audit Files Configuration

VVSnapshot Report contains:
- Audit subsystem statistics (auditcmd -c)
- Audit Events enabled
- Alarm configuration (alarm_defs)
Staging Directory contains: N/A

The VVSCAssist Audit Files Configuration module performs the following tasks:
- For each VV event with audit enabled, an equivalent set of SC events is set to be audited. Depending on whether the VV event triggers on failure or success, the SC event is added to the "success" or "failure" audit list accordingly. Those lists are later examined and if there is an SC event in both lists, the event is added to the "both" audit list and removed from the other two lists.
- Add entries to the TODO checklist file

Do you want to execute Audit Files Configuration tasks?

AUDIT_FILES_CONFIGURATION=1  # 1=Yes  0=No

Cron Files Configuration

VVSnapshot Report contains:
- /var/adm/cron/at.allow
- /var/adm/cron/cron.allow
- List all at jobs scheduled: at -l <all_users>
- List all crontabs scheduled: crontabs -l <all_users>

Staging Directory contains:
- /var/adm/cron/at.allow
- /var/adm/cron/cron.allow
- /var/spool/cron/crontabs/
- /var/spool/cron/atjobs/

The VVSCAssist Cron Files Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute Cron Files Configuration tasks?

CRON_FILES_CONFIGURATION=1  # 1=Yes  0=No

Openview VPO Client Configuration

VVSnapshot Report contains:
- ITOAgent Software version
- /var/opt/OV/conf/OpC/nodeinfo

Staging Directory contains:
- /var/opt/OV/

The VVSCAssist Openview VPO Client Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute VPO Client Configuration tasks?
VPO_CLIENT_CONFIGURATION=1  # 1=Yes  0=No

#########################################################################
# Common Gateway CGI Configuration  #
#########################################################################

# NOTE: This CGI module replaces the VVSsnapshot Trusted Gateway TGA
# Configuration module captured in Virtualvault

# VVSsnapshot Report contains:
#   - tgad.conf contents
#   - CGI directory contents

# Staging Directory contains: All files above

# The CGI Module executes only if there is no existing CGI web server instance
# on the system. The VVSAssist Common Gateway CGI Configuration module performs
# the following tasks:
# - Calls vvws_createxsec to create the INSIDE CGI instance
#   - Create the CGI instance wp-cgi
#   - Create the CGI compartment iws_cgi
#   - Create the CGI chroot environment wp_cgi
#   - Log vvws_createxsec CGI creation details to vvws_createxsec.log
#   - Add 'Listen 127.0.0.1:11340' to wp-cgi instance
#   - Copy VVSsnapshot tgad.conf CGI Directories to new CGI chroot:
#     - /var/jail/wp_cgi/cgibin/ from VV /var/opt/vaultTS/inside/app/cgibin/
#     - /var/jail/wp_cgi/dir1_dir2_cgibin/ from VV /dir1/dir2/cgibin/ e.g.:  
#       /var/jail/wp_cgi/home_<user>_cgibin/ from VV /home/<user>/cgibin/
#   - Create symbolic link for default CGI directory (for naming consistency):  
#     - Copy VVSsnapshot tgad.conf CGI Directories to new CGI chroot:
#   - Call setrules to activate compartment iws_cgi
#   - Start CGI instance wp-cgi
#   - Add entries to the TODO checklist file

# Do you want to execute CGI Configuration tasks?
CGI_CONFIGURATION=1  # 1=Yes  0=No

#########################################################################
# Java Servlet Configuration  #
#########################################################################

# VVSsnapshot Report contains:
#   - Java SDK or RTE version
#   - Java Servlet (Tomcat) configuration:
#     - conf/ contents
#     - webapps/ contents
#     - Web server mod_jk instance: <ws-instance>
#     - mod_jk.conf<ws-instance> contents
#     - workers.properties contents
#   - Java Servlet (Jserv) configuration

# Staging Directory contains:
#   - /var/opt/vaultTS/inside/app/servlet/
#   - /var/opt/vaultTS/inside/app/tomcat/

# The VVSAssist Java Servlet Configuration module performs the
# following tasks:
# - Enables secured Tomcat server if Tomcat found in VVSsnapshot
Do you want to execute Java Servlet Configuration tasks?

JAVA_SERVLET_CONFIGURATION=1     # 1=Yes  0=No

#######################################
# Webproxy Configuration
#######################################

VVSnapshot Report contains:
- Instance configuration:
  - Apache version
  - conf/ contents
  - httpd.conf contents
  - ssl.conf contents (Apache 2.0 only)
  - SSL server certificate contents
- Staging Directory contains:
  - /opt/vvproxy/conf/
  - /opt/vvproxy2/conf/ (Virtualvault 4.7 only)

The VVSCAssist Webproxy Configuration module performs the following tasks:
- Module executes only if Webproxy instance previously non-existent
- Parse the following directives from each httpd.conf found in VVSnapshot:
  - Port (Listen)
  - SSL Port (Listen)
  - User
  - Group
  - Chroot
- Calls vvws_createxsec to create the INSIDE Webproxy instance
- Create the Webproxy instance iws_webproxy or iws_webproxy2
  NOTE: The wp-webproxy2 instance will only exist if VVSnapshot found both (1.3 and 2.0) Webproxy instances on VV
- Create the Webproxy compartment iws_webproxy or iws_webproxy2
- Create the Webproxy chroot environment wp_webproxy or wp_webproxy2
- Log vvws_createxsec Webproxy creation details to vvws_createxsec.log
- Add 'Listen 127.0.0.1:11345' to wp-webproxy instance
  NOTE: vvws_createxsec proxies new OUTSIDE web server client instances to this INSIDE Webproxy server using Port 11345
- Add commented VV Proxy* & Rewrite* directives to new httpd.conf
- Add note to new ssl.conf indicating ssl.key and ssl.crt was copied
- VV chroot directory contents are marked for copy later by the VVSCAssist Chroot Configuration module below
- Call setrules to activate compartment iws_webproxy or iws_webproxy2
- Start Webproxy instance wp-webproxy or wp-webproxy2
- Add entries to the TODO checklist file

Do you want to execute Webproxy Configuration tasks?

WEBPROXY_CONFIGURATION=1     # 1=Yes  0=No

#######################################
# Web Server Configuration
#######################################

VVSnapshot Report contains:
- Instances summary
- Instance <ws-instance> configuration:
  - Apache version
Appendix D: VVSCAssist Configuration

- conf/ contents
- httpd.conf contents
- ssl.conf contents (Apache 2.0 only)
- SSL server certificate contents

Staging Directory contains:
- /opt/vaultWS/servers/<ws-instance>/conf/

The VVSCAssist Web Server Configuration module performs the following tasks:

- Module executes only if web server instance previously non-existent
- Parse the following directives from each httpd.conf found in VVSnapshot:
  - ServerName
  - Port (Listen)
  - SSL Port (Listen)
  - User
  - Group
  - ProcessSL
  - Chroot
- Calls vvws_createxsec to create the (OUTSIDE or INSIDE) server instance
  - Create the web server instance wp-<servername>
  - Create the web server compartment ows_<servername> or iws_<servername>
- Create the web server chroot environment:
  - /var/jail/wp_internet/ if VV Chroot was /var/opt/vaultTS/outside/app/
  - /var/jail/wp_intranet/ if VV Chroot was /var/opt/vaultTS/inside/app/
  - /var/jail/wp_<servername>/ if non-default Chroot was used
- Log vvws_createxsec web server creation details to vvws_createxsec.log
- Fixup CGI client directives to the correct CGI listening port 11340
- Fixup Webproxy client directives to the correct Webproxy listening port
- Add commented VV Proxy* and Rewrite* directives to new httpd.conf
- Copy VV ssl.key/server.key and ssl.crt/server.crt into new instance
- Add note to new ssl.conf indicating ssl.key and ssl.crt was copied
- VV chroot directory contents are marked for copy later by the VVSCAssist Chroot Configuration module below
- Start web server instance wp-<servername>
- Add entries to the TODO checklist file

Do you want to execute Web Server Configuration tasks?

WEB_SERVER_CONFIGURATION=1     # 1=Yes  0=No

#######################################
## Chroot Configuration              ##
#######################################

VVSnapshot Report contains:
- A note comparing chroot locations on Virtualvault vs. Security Containment

Staging Directory contains:
- Web Server Chroot directories parsed from <ws-instance> httpd.conf:
  - Chroot /var/opt/vaultTS/outside/app
- Webproxy Chroot directories parsed from httpd.conf:
  - Chroot /var/opt/vvproxy
- TGA Chroot directories parsed from tgad.conf:
  - gw_root=/var/opt/vaultTS/inside/app

The VVSCAssist Chroot Configuration module performs the following tasks:

- Copy VVSnapshot chroot directories found in Web Server and Webproxy Configuration modules above to the new chroot locations
- /var/jail/ will typically contain the following chroot environments:
  - wp.cgi/ for wp-cgi server instance
Appendix D: VVSCAssist Configuration

# wp_webproxy/ for wp-webproxy instance
# wp_internet/ for OUTSIDE web servers using /var/opt/vaultTS/outside/app
# wp_intranet/ for INSIDE web servers using /var/opt/vaultTS/inside/app
# wp_<servername>/ for each OUTSIDE or INSIDE server not using VV defaults
- Copy all subdirectories and files found in VVSnapshot chroot except:
  - dev/ SC equivalent already created by mkchroot
  - lib/ SC equivalent already created by mkchroot
  - servlet/ SC implementation is different from VV
  - tmp/ SC equivalent should be empty
  - tomcat/ SC implementation is different from VV
  - usr/ SC equivalent already created by mkchroot
- Following directories on VV copy into different location on SC:
  - html/ copies to <chroot>/opt/hpws/apache/htcdocs/
  - icons/ copies to <chroot>/opt/hpws/apache/icons/
- Create symlinks for html/ and icons/ so VV admins find them easily
  - html/ -> <chroot>/opt/hpws/apache/htcdocs/
  - icons/ -> <chroot>/opt/hpws/apache/icons/
- Add entries to the TODO checklist file

Do you want to execute Chroot Configuration tasks?

CHROOT_CONFIGURATION=1 # 1=Yes 0=No

# Trusted Proxy TGP Configuration
VVSnapshot Report contains:
- tgp.conf contents
Staging Directory contains:
- tgp.conf

The VVSCAssist Trusted Proxy TGP Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute TGP Configuration tasks?

TGP_CONFIGURATION=1 # 1=Yes 0=No

# Trusted IPC Configuration
VVSnapshot Report contains:
- Executables with Trusted IPC privilege:
  - Server privileges: netmultilevelserver, netsetid
  - Client privilege: netprivsession
  - SVIPC Facilities (ipcs)
Staging Directory contains: N/A

The VVSCAssist Trusted IPC Configuration module performs the following tasks:
- Add entries to the TODO checklist file

Do you want to execute Trusted IPC Configuration tasks?

TRUSTED_IPC_CONFIGURATION=1 # 1=Yes 0=No
The following code enforces module dependencies. Do not modify!

```bash
if [ $WEB_SERVER_CONFIGURATION -eq 1 -o $WEBPROXY_CONFIGURATION -eq 1 ]; then
  CHROOT_CONFIGURATION=1
fi
```

The following code enforces module dependencies. Do not modify!
Appendix E: VVSCAssist TODO

The following is a sample listing of a VVSCAssist TODO file, produced by the `vvscassist.sh` script.

```
# Exec /opt/hpvv/vvscassist/vvscassist.sh  Wed Jun 29 18:48:42 EDT 2005

VVSCAssist TODO Checklist. Please check each item as completed:

[ ] Boot Time Configuration
[ ] Kernel Tunables Configuration
[ ] Users Groups Configuration
[ ] Secure Shell SSH Configuration
[ ] Filesystem Configuration
[ ] Network Configuration
[ ] Audit Files Configuration
[ ] Cron Files Configuration
[ ] Openview VPO Client Configuration
[ ] Common Gateway CGI Configuration
[ ] Java Servlet Configuration
[ ] Webproxy Configuration
[ ] Web Server Configuration
[ ] Trusted Proxy TGP Configuration
[ ] Trusted IPC Configuration
```

[ ] The following non-default entries were found in Virtualvault's
/etc/inittab. Please verify their correct functioning and add them to your system's /etc/inittab as appropriate. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/etc/inittab

[ ] userapp1
[ ] userapp2

[ ] The following non-default files were found in Virtualvault's /etc/rc.config.d/. Please verify their correct functioning and add them to your system's /etc/rc.config.d/ as appropriate. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/etc/rc.config.d/*

[ ] jss8009
[ ] opcagt
[ ] userapp1
[ ] userapp2

[ ] The following non-default files were found in Virtualvault's /sbin/init.d/. Please verify their correct functioning and add them to your system's /sbin/init.d/ (including symlinks to other /sbin/rc*.d/ subdirectories) as appropriate. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/sbin/init.d/*
/opt/hpvv/vvsnapshot_staging/sbin/rc[0-4].d/*

[ ] jss8009
[ ] opcagt
[ ] userapp1
[ ] userapp2

########################################################################
## Kernel Tunables Configuration ##
########################################################################

[ ] The following non-default tunable values were found in Virtualvault's /stand/system. Please verify their correct functioning and use them as appropriate. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/stand/system

<table>
<thead>
<tr>
<th>NAME</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>STRMSGSZ</td>
<td>65535</td>
</tr>
<tr>
<td>maxswapchunks</td>
<td>512</td>
</tr>
<tr>
<td>nstrpty</td>
<td>60</td>
</tr>
<tr>
<td>scroll_lines</td>
<td>99</td>
</tr>
</tbody>
</table>

[ ] The system definition of the Virtualvault system as generated by sysdef(1m) has been preserved. Please use this as a reference to configure kernel parameters. Detailed information is available in the VVSsnapshot Report File.

########################################################################
## Users Groups Configuration ##
########################################################################

[ ] Non-default User entries were found in the VVSsnapshot Report File. User related configuration files and home directory files were preserved in the staging archive:
VVSCAssist calls useradd(1M) to create each VV user on the Security Containment system. Additionally, the user's VV home directory files were copied into place from the staging area. Note that dot files such as .login and .profile are copied as .login_vv and .profile_vv to preserve the active SC versions of the files.

NOTE: User configuration attributes differ between the VV and SC systems

Therefore, the new user configuration should be verified before allowing user login to the system. The user config files can be compared with the VV user config in the staging area. Once the user config is verified, use the passwd(1M) command to enable user login.

The following VV users should be verified for login:

[ ] vaultadm
[ ] opc_op
[ ] safeusr

[ ] Non-default Group entries were found in the VVSnapshot Report File. The VV group configuration was preserved in the staging archive:

/opt/hpvv/vvsnapshot_staging/etc/group

VVSCAssist calls groupadd(1M) to create each VV group on the Security Containment system. The following VV groups should be verified for use on the SC system:

[ ] www
[ ] vaultusr
[ ] opcgrp
[ ] safegrp

#######################################
## Secure Shell SSH Configuration    ##
#######################################

[ ] Host-level SSH configuration was found in the VVSnapshot Report File. Virtualvault SSH host config files were preserved in the staging archive:

/opt/hpvv/vvsnapshot_staging/opt/vaultTS/tools/OpenSSH/etc/

On the Security Containment system, HP-UX Secure Shell (SSH) was installed during the initial system installation. Additionally, the default host-level configuration was enabled at that time in /opt/ssh/etc/.

NOTE: SSH configuration attributes may differ between the VV and SC systems

Therefore, the new SSH configuration should be verified before allowing SSH login access to the system. The current SSH config files can be compared with the VV SSH config in the staging area to select the appropriate values.

[ ] User-level SSH configuration was found in the VVSnapshot Report File. Virtualvault SSH user config files were preserved in the staging archive:

/opt/hpvv/vvsnapshot_staging/home/<user>/apphome/{VV_mld_INSIDE}/.ssh/

VVSCAssist activates the VV user SSH config files by copying them from the staging directory to the user's home directory at /home/<user>/.ssh/. The SSH configuration for the following users should be verified before allowing SSH login access to the system:
Appendix E: VVSCAssist TODO

[ ] user1
[ ] user2
[ ] vaultadm

#######################################
## Filesystem Configuration          ##
#######################################

[ ] The following LVM Volume Groups and Logical Volumes were detected. Please verify that they are needed on the Security Containment system and create them accordingly. Detailed information is available in the VVSsnapshot Report File.

/dev/vg00
/dev/vg00/lvol1 /stand
/dev/vg00/lvol3 /
/dev/vg00/lvol4 /tmp
/dev/vg00/lvol5 /home
/dev/vg00/lvol6 /opt
/dev/vg00/lvol7 /usr
/dev/vg00/lvol8 /var

[ ] Additionally, a complete listing of the mode, owner, group, access control lists, privileges, and security classification of each file has been preserved. Please use this information as a reference when configuring the Security Containment system. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/l*.out

[ ] DAC User Group Permissions (ll -R / > ll.out)
[ ] DAC Access Control Lists (lsacl -R / > lsacl.out)
[ ] MAC Sensitivity Levels (lslevel -R / > lslevel.out)
[ ] MAC Privileges (lspriv -R / > lspriv.out)

#######################################
## Network Configuration             ##
#######################################

[ ] The following Virtualvault Network configuration files were found and preserved. Please verify their appropriateness and use them as a reference when configuring the Security Containment system. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/etc/*

[ ] /etc/hosts
[ ] /etc/services
[ ] /etc/nsswitch.conf
[ ] /etc/resolv.conf
[ ] /etc/rc.config.d/netconf
[ ] /etc/rc.config.d/nddconf

[ ] The interface configurations, routing tables, and ports with active TCP listeners were recorded for reference when configuring the Security Containment system. Detailed information is available in the VVSsnapshot Report File.

[ ] Network Interface Cards (NICs) configuration
[ ] Network Routing Table configuration
[ ] Network Listening Endpoints

#######################################
## Audit Files Configuration         ##
#######################################
The audit configuration of the Virtualvault system was recorded and the audit configuration on this system has been updated to monitor the set of system calls which were monitored on Virtualvault. Auditing has been enabled if it was enabled on the Virtualvault. Detailed information is available in the VVSsnapshot Report File.

The original Audit configuration file has been preserved as:

/etc/rc.config.d/auditing.orig

The alarm configuration on the Virtualvault system was recorded and should be used as a reference if an alarm-generating system (such as Host IDS) is installed. Detailed information is available in the VVSsnapshot Report File.

The following files and directories related to cron(1m) and/or at(1) on Virtualvault were preserved. Please verify their applicability and install them as appropriate. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/var/*/*/cron/*

/var/adm/cron/at.allow
/var/adm/cron/cron.allow
/var/spool/cron/atjobs/
/var/spool/cron/crontabs/

Below is a list of crontab files preserved, sorted by user and compartment. Note that on the Security Containment system, a given user's crontab will consist of one unified file containing commands for all compartments. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/var/spool/cron/crontabs/*

<table>
<thead>
<tr>
<th>User</th>
<th>Compartment</th>
</tr>
</thead>
<tbody>
<tr>
<td>audit</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>root</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>vaultadm</td>
<td>SYSHI</td>
</tr>
<tr>
<td>vaultadm</td>
<td>SYSTEM/INSIDE</td>
</tr>
<tr>
<td>vaultadm</td>
<td>SYSTEM/OUTSIDE</td>
</tr>
<tr>
<td>user1</td>
<td>SYSTEM/INSIDE</td>
</tr>
<tr>
<td>user1</td>
<td>SYSTEM/OUTSIDE</td>
</tr>
<tr>
<td>user2</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>user2</td>
<td>SYSTEM/INSIDE</td>
</tr>
</tbody>
</table>

The following Openview VPO items were preserved for reference when configuring the Security Containment system. Please verify the appropriateness of the information before using it. Detailed information is available in the VVSsnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/var/opt/OV/*
CGI Directory /var/opt/vaultTS/inside/app/cgibin/
was the default CGI directory in the Virtualvault TGA configuration
file. The directory was successfully copied into the CGI Chroot at:
/var/jail/wp_cgi/cgibin/

This is the new Security Containment system default CGI directory for
all OUTSIDE non-Webproxy web servers. Please verify this configuration
and correct it as appropriate. Detailed information is available
in the VVSsnapshot Report File and:
/opt/hpvv/vvsnapshot_staging/tcb/files/tgad.conf

Non-default CGI Directory /home/user1/cgibin/
was found in the TGA configuration file. The directory
was successfully copied into the CGI Chroot as:
/var/jail/wp_cgi/home_user1_cgibin/

However no Apache web server ScriptAlias directive has been configured
to execute CGIs from this location. Please verify this configuration
and correct it as appropriate. Detailed information is available in
the VVSsnapshot Report File and:
/opt/hpvv/vvsnapshot_staging/tcb/files/tgad.conf

The following Virtualvault Java configuration files were found and
preserved. Please verify their appropriateness and use them as a reference when configuring the Security Containment system. Detailed
information is available in the VVSsnapshot Report File and:

/mod_jk.conf-outsidetomcat
/server.xml
/workers.properties
(/
/httpd.conf

The Security Containment implementation for Java Tomcat Servlets differs
from Virtualvault. The HP-UX 11i Web Server Suite Tomcat-based Servlet
Engine will be used. The following Tomcat configuration directories
and files are used:

/opt/hpws/tomcat/conf/
/opt/hpws/tomcat/webapps/

The Tomcat Servlet Engine runs in the in_tomcat compartment on the
Security Containment system. The compartment was created successfully,
however, no Network Communication Rules are granted by default. If the
Tomcat server connects to the Intranet, appropriate Network Rules
must be granted in the in_tomcat compartment definition file:

/etc/cmpt/in_tomcat.rules

The following Virtualvault Java configuration files were found and
preserved. Please verify their appropriateness and use them as a reference when configuring the Security Containment system. Detailed
information is available in the VVSsnapshot Report File and:

/mod_jk.conf-outsidetomcat
/server.xml
/workers.properties
(/
/httpd.conf

The Security Containment implementation for Java Tomcat Servlets differs
from Virtualvault. The HP-UX 11i Web Server Suite Tomcat-based Servlet
Engine will be used. The following Tomcat configuration directories
and files are used:

/opt/hpws/tomcat/conf/
/opt/hpws/tomcat/webapps/

The Tomcat Servlet Engine runs in the in_tomcat compartment on the
Security Containment system. The compartment was created successfully,
however, no Network Communication Rules are granted by default. If the
Tomcat server connects to the Intranet, appropriate Network Rules
must be granted in the in_tomcat compartment definition file:

/etc/cmpt/in_tomcat.rules
Proxy* and Rewrite* directives were found in Virtualvault's Webproxy /opt/vvproxy instance. Detailed information is available in the VVSnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/opt/vvproxy/conf/httpd.conf

The following comment was added to the new wp-webproxy instance's httpd.conf. Please verify that the directives are correct and uncomment as appropriate to re-activate behavior:

```
# The following Proxy* and Rewrite* directives were found
# in the original Webproxy instance on Virtualvault. Please
# examine and uncomment the directives as appropriate:
#
# ProxyRequests On
# RewriteEngine on
# RewriteRule ^/(.*)$ http://www.myinside.com/$1 [P]
# ProxyPassReverse / http://www.myinside.com/
#
# -------------- End Virtualvault Comment ---------------
```

The Webproxy server runs in the iws_webproxy compartment on the Security Containment system. The compartment was created successfully with a default Network Rule granting access to any port on the Intranet (in_iface compartment); see the 'grant client tcp in_iface' rule in:

```
/etc/cmpt/iws_webproxy.rules
```

However, a tighter Network Rule may be desired. For example, if Webproxy proxies requests only to port 80 of an internal web server, the in_iface compartment grant rule can be strengthened to:

```
grant client tcp peer port 80 in_iface
```

An SSL Certificate was found in Virtualvault's Webproxy /opt/vvproxy instance. Detailed information is available in the VVSnapshot Report File and:

```
/opt/hpvv/vvsnapshot_staging/opt/vvproxy/conf/ssl.crt/server.crt
/opt/hpvv/vvsnapshot_staging/opt/vvproxy/conf/ssl.key/server.key
```

The following comment was added to the new wp-webproxy instance's ssl.conf. Please verify that this is the correct SSL Certificate configuration:

```
#   ---------------- Begin Virtualvault Comment ----------------
#   NOTE: The original Virtualvault SSL Certificate files have
#         been copied into this instance configuration. If you
#         do not want the Virtualvault certificate to be
#         installed here. Please replace the following files
#         with your desired certificate files:
#             wp-webproxy/conf/ssl.crt/server.crt
#             wp-webproxy/conf/ssl.key/server.key
```
Appendix E: VVSC Assist TODO

Virtualvault to Security Containment Transition
Hewlett Packard Company.

# ------------- End Virtualvault Comment -----------

#############################
# Web Server Configuration  
#############################

[ ] An SSL Certificate was found in Virtualvault's Web Server
ws-outside instance. Detailed information is available in
the VVSnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/opt/vaultWS/servers/ws-
outside/conf/ssl.crt/server.crt

The following comment was added to the new wp-outside instance's
ssl.conf. Please verify that this is the correct SSL Certificate
configuration:

[ ] /opt/hpws/apache/webproxy/servers/wp-outside/conf/ssl.conf

# ------------- Begin Virtualvault Comment -------------
# NOTE: The original Virtualvault SSL Certificate files have
# been copied into this instance configuration. If you
do not want the Virtualvault certificate to be
# installed here. Please replace the following files
# with your desired certificate files:
# wp-outside/conf/ssl.crt/server.crt
# wp-outside/conf/ssl.key/server.key
# # ------------- End Virtualvault Comment -------------

[ ] Proxy* and Rewrite* directives were found in Virtualvault's Web Server
ws-outsideproxy instance. Detailed information is available in the
VVSnapshot Report File and:

/opt/hpvv/vvsnapshot_staging/opt/vaultWS/servers/ws-
outsideproxy/conf/httpd.conf

The following comment was added to the new wp-outsideproxy instance's
httpd.conf. Please verify that the directives are correct and uncomment
as appropriate to re-activate behavior:

[ ] /opt/hpws/apache/webproxy/servers/wp-outsideproxy/conf/httpd.conf

# ------------- Begin Virtualvault Comment -------------
# # The following Proxy* and Rewrite* directives were found
# in the original web server instance on Virtualvault. Please
# examine and uncomment the directives as appropriate:
# #   ProxyRequests On
#   RewriteEngine on
#   RewriteRule ^/(.*)$  http://127.0.0.1:8080/$1 [P]
#   ProxyPassReverse  /  http://127.0.0.1:8080/
# # ------------- End Virtualvault Comment -------------

#############################
# Trusted Proxy TGP Configuration 
#############################
The following TGP items were preserved for reference when configuring Security Containment compartment rules. Detailed information is available in the VVSnapshot Report File and:

```
/opt/hpvv/vvsnapshot_staging/tcb/files/tgp.conf
```

- tgp-service1
- tgp-service2

```
```

## Trusted IPC Configuration

The following Trusted IPC items were preserved for reference when configuring Security Containment compartment rules. Detailed information is available in the VVSnapshot Report File and:

```
```

- /usr/safe-tgp/bin/safe.server
- /usr/safe/bin/safe.client
- /usr/safe/bin/safe.server

SVIPC Facilities (ipcs -op) were preserved for reference when configuring the Security Containment system. Please verify the appropriateness of the information before using it. Detailed information is available in the VVSnapshot Report File.
## Appendix F: Complete HP-UX Software Bundles List

The following is the complete listing of the recommended software selections when installing HP-UX 11iv2:

- Software Tab
- Category: All

<table>
<thead>
<tr>
<th>Marked</th>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>10GigEthr-00</td>
<td>PCI-X 10 Gigabit Ethernet; Suppnd HW=AB287</td>
</tr>
<tr>
<td>No</td>
<td>ATM-00</td>
<td>PCI ATM; Suppnd HW=A5513A</td>
</tr>
<tr>
<td>No</td>
<td>B3929DA</td>
<td>HP OnLineJFS (Server)</td>
</tr>
<tr>
<td>No</td>
<td>B3929EA</td>
<td>HP OnLineJFS (Server)</td>
</tr>
<tr>
<td>Yes</td>
<td>B5725AA</td>
<td>HP-UX Installation Utilities (Ignite-UX)</td>
</tr>
<tr>
<td>No</td>
<td>B6848BA</td>
<td>Ximian GNOME 1.4 GTK+ Libraries for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>B6849AA</td>
<td>Bastille Security Hardening Tool</td>
</tr>
<tr>
<td>Yes</td>
<td>B9073BA</td>
<td>HP-UX iCOD (Instant Capacity)</td>
</tr>
<tr>
<td>Yes</td>
<td>B9901AA</td>
<td>HP IPFilter 3.5alpha5</td>
</tr>
<tr>
<td>No</td>
<td>Base-VXFS</td>
<td>VERITAS File System Bundle 4.1 for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>Base-VXVM</td>
<td>Base VERITAS Volume Manager Bundle 4.1 for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>BUNDLE11i</td>
<td>Required Patch Bundle for HP-UX 11i v2</td>
</tr>
<tr>
<td>No</td>
<td>DSAUtilities</td>
<td>HP-UX Distributed Systems Administration Utilities</td>
</tr>
<tr>
<td>----</td>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Yes</td>
<td>EnableVxFS</td>
<td>File-System library, commands enhancements for VxFS4.1</td>
</tr>
<tr>
<td>Yes</td>
<td>FDDI-00</td>
<td>PCI FDDI; Supptd HW=A3739B; SW=J3626AA</td>
</tr>
<tr>
<td>Yes</td>
<td>FEATURE11i</td>
<td>Feature Enablement Patches for HP-UX 11i v2, March 2006</td>
</tr>
<tr>
<td>Yes</td>
<td>FibrChanl-00</td>
<td>PCI FibreChannel; Supptd HW=A6795A, A5158A</td>
</tr>
<tr>
<td>Yes</td>
<td>FibrChanl-01</td>
<td>PCI-X FibreChannel; Supptd HW=A6826A, A9782A, A9784A, AB465A, AB378A, AB379A</td>
</tr>
<tr>
<td>Yes</td>
<td>FileSystems</td>
<td>HP-UX Disks and File Systems Tool Bundle</td>
</tr>
<tr>
<td>Yes</td>
<td>General Patches</td>
<td>Mark to load all patches, unmark for just critical &amp; HW patches.</td>
</tr>
<tr>
<td>Yes</td>
<td>GigEther-00</td>
<td>PCI GigEther; Supptd HW=A4926A/A4929A/A6096A; SW=J1642AA</td>
</tr>
<tr>
<td>Yes</td>
<td>GigEther-01</td>
<td>PCI GigEther; Supptd HW=A6825A/A6794A/A6847A/A8685A/A9782A/A9784A/A7109A/AB465A</td>
</tr>
<tr>
<td>No</td>
<td>HPSIM-HP-UX</td>
<td>HP Systems Insight Manager Server Bundle</td>
</tr>
<tr>
<td>Yes</td>
<td>HPUX-HIDS</td>
<td>HP-UX Host IDS E.04.00</td>
</tr>
<tr>
<td>Yes</td>
<td>HPUXBaseAux</td>
<td>HP-UX Base OS Auxiliary</td>
</tr>
<tr>
<td>No</td>
<td>HPUXMOBILEIP</td>
<td>HP-UX Mobile IP Software</td>
</tr>
<tr>
<td>Yes</td>
<td>hpuxwsApache</td>
<td>HP-UX Apache-based Web Server</td>
</tr>
<tr>
<td>Required</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Yes</td>
<td>hpuxwsTomcat</td>
<td>HP-UX Tomcat-based Servlet Engine</td>
</tr>
<tr>
<td>No</td>
<td>hpuxwsWebmin</td>
<td>HP-UX Webmin-based Admin</td>
</tr>
<tr>
<td>Yes</td>
<td>hpuxwsXml</td>
<td>HP-UX XML Web Server Tools</td>
</tr>
<tr>
<td>Yes</td>
<td>HWEnable11i</td>
<td>Hardware Enablement Patches for HP-UX 11i v2, March 2006</td>
</tr>
<tr>
<td>No</td>
<td>HyprFabrc-00</td>
<td>PCI HyperFabric; Supptd HW=A6386A</td>
</tr>
<tr>
<td>Yes</td>
<td>IEther-00</td>
<td>PCI/PCI-X IEther; Supptd HW=A7011A/A7012A/AB352A/AB290A/AB545A</td>
</tr>
<tr>
<td>No</td>
<td>Ignite-UX-11-00</td>
<td>HP-UX Installation Utilities for Installing 11.00 Systems</td>
</tr>
<tr>
<td>No</td>
<td>Ignite-UX-11-11</td>
<td>HP-UX Installation Utilities for Installing 11.11 Systems</td>
</tr>
<tr>
<td>No</td>
<td>Ignite-UX-11-23</td>
<td>HP-UX Installation Utilities for Installing 11.23 Systems</td>
</tr>
<tr>
<td>No</td>
<td>iSCSI-00</td>
<td>HP-UX iSCSI Software Initiator</td>
</tr>
<tr>
<td>No</td>
<td>ISEEPlatform</td>
<td>ISEE Platform</td>
</tr>
<tr>
<td>No</td>
<td>J4258CA</td>
<td>Netscape Directory Server v6 for HP-UX</td>
</tr>
<tr>
<td>No</td>
<td>J4269AA</td>
<td>LDAP-UX Integration</td>
</tr>
<tr>
<td>Yes</td>
<td>Java15JDK</td>
<td>Java 1.5 JDK for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>Java15JDKadd</td>
<td>Java 1.5 JDK -AA addon for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>Java15JRE</td>
<td>Java 1.5 JRE for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>Java15JREadd</td>
<td>Java 1.5 JRE -AA addon for HP-UX</td>
</tr>
<tr>
<td>No</td>
<td>JAVAOOOB</td>
<td>Java2 Out-of-box for HP-UX</td>
</tr>
<tr>
<td>Opt</td>
<td>Judy</td>
<td>Judy Library and Related files</td>
</tr>
<tr>
<td>Requirement</td>
<td>Product</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Yes</td>
<td>LVMProvider</td>
<td>CIM/WBEM Provider for LVM</td>
</tr>
<tr>
<td>Opt</td>
<td>MOZILLA</td>
<td>Mozilla for HP-UX</td>
</tr>
<tr>
<td>No</td>
<td>MOZILLAsrc</td>
<td>Mozilla Source distribution</td>
</tr>
<tr>
<td>No</td>
<td>NSDirSvr7</td>
<td>Netscape Directory Server v7 for HP-UX</td>
</tr>
<tr>
<td>No</td>
<td>OBSOLESCENCE</td>
<td>OBSOLESCENCE</td>
</tr>
<tr>
<td>Yes</td>
<td>OnlineDiag</td>
<td>HPUX 11.23 Support Tools Bundle, Mar 2006</td>
</tr>
<tr>
<td>No</td>
<td>ParMgr</td>
<td>Partition Manager - HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>perl</td>
<td>Perl Programming Language</td>
</tr>
<tr>
<td>Yes</td>
<td>QPKAPPSS</td>
<td>Applications Quality Pack Bundle for HP-UX 11i v2, March 2006</td>
</tr>
<tr>
<td>Yes</td>
<td>QPKBASE</td>
<td>Base Quality Pack Bundle for HP-UX 11i v2, March 2006</td>
</tr>
<tr>
<td>Yes</td>
<td>RAID-01</td>
<td>RAID SA; Supptd HW=A7143A/A9890A/A9891A</td>
</tr>
<tr>
<td>Yes</td>
<td>scsiU320-00</td>
<td>PCI-X SCSI U320; Supptd HW=A7173A/AB290A</td>
</tr>
<tr>
<td>Yes</td>
<td>Sec00Tools</td>
<td>Install-Time security infrastructure.</td>
</tr>
<tr>
<td>No</td>
<td>Sec10Host</td>
<td>Host-Based Lockdown, without IPFilter configuration</td>
</tr>
<tr>
<td>No</td>
<td>Sec20MngDMZ</td>
<td>Lockdown + block most incoming traffic with IPFilter firewall</td>
</tr>
<tr>
<td>No</td>
<td>Sec30DMZ</td>
<td>Host-Based and IPFilter Network Lockdown</td>
</tr>
<tr>
<td>Yes</td>
<td>SecPatchCk</td>
<td>HP-UX Security Patch Check Tool</td>
</tr>
<tr>
<td>No</td>
<td>SerialSCSI-00</td>
<td>PCI-X SerialSCSI; Supptd HW=AB288A</td>
</tr>
<tr>
<td>No</td>
<td>SwPkgBuilder</td>
<td>Software Package Builder</td>
</tr>
<tr>
<td>Yes</td>
<td>SysFaultMgmt</td>
<td>HPUX System Fault Management</td>
</tr>
<tr>
<td>No</td>
<td>Component</td>
<td>Description</td>
</tr>
<tr>
<td>----</td>
<td>---------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Yes</td>
<td>SysMgmtWeb</td>
<td>HP-UX Web Based System Management User Interfaces</td>
</tr>
<tr>
<td>Yes</td>
<td>T1456AA</td>
<td>Java 2.1.4 SDK for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>T1456AAaddon</td>
<td>Java 2.1.4 SDK - AA addon for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>T1457AA</td>
<td>Java 2.1.4 RTE for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>T1457AAaddon</td>
<td>Java 2.1.4 RTE - AA addon for HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>T1471AA</td>
<td>HP-UX Secure Shell</td>
</tr>
<tr>
<td>No</td>
<td>T2351AA</td>
<td>HP Pay per use (PPU)</td>
</tr>
<tr>
<td>No</td>
<td>T2743AA</td>
<td>HP Global Workload Manager Agent</td>
</tr>
<tr>
<td>No</td>
<td>TermIO-00</td>
<td>PCI MUX; Supptd HW=A6748A/A6749A; SW=J3596A</td>
</tr>
<tr>
<td>No</td>
<td>TokenRing-00</td>
<td>PCI Token Ring; Supptd HW=A5783A; SW=J1644AA</td>
</tr>
<tr>
<td>Yes</td>
<td>USB-00</td>
<td>USB Subsystem and Drivers</td>
</tr>
<tr>
<td>No</td>
<td>UtilProvider</td>
<td>HP-UX Utilization Provider</td>
</tr>
<tr>
<td>No</td>
<td>VMGuestLib</td>
<td>Integrity VM Guest Support Libraries</td>
</tr>
<tr>
<td>No</td>
<td>VMPProvider</td>
<td>WBEM Provider for Integrity VM</td>
</tr>
<tr>
<td>No</td>
<td>vParProvider</td>
<td>vPar Provider - HP-UX</td>
</tr>
<tr>
<td>Yes</td>
<td>WBEMP-LAN-00</td>
<td>LAN Provider for Ethernet LAN interfaces.</td>
</tr>
</tbody>
</table>

1. Mozilla 1.4 may be optionally installed if a web browser client is desired on the system.
Appendix G: Disk Space Requirement Template

The space required for VVSCAssist will vary greatly from system to system. The two possible areas where most space will be required are web server chroot instances and user installed applications.

Space for user installed applications

Use regular system administrator tools like du(1M) to determine the space needed by the CGI applications installed on your system. Add that space to the total additional amount of space needed.

Space for web server chroot instances

Web server chroot instances require the duplication of certain directories containing required files, like libraries and executables. For each web server chroot instance there is an approximate space of 300Mb needed. However, not all web server chroot instances require their own separate space. There are instances that share the required directories.

The following sections serve as a guide to determine how many web server chroot instances will be installed by VVSCAssist from the Virtualvault configuration information contained in the Staging Archive.

Directory locations

VVSCAssist creates instances of web servers on the Security Containment system from the VVSnapshot data. All web server chroot instances are created under /var/jail. The following table lists the Virtualvault web server chroot locations and their corresponding Security Containment chroot location:
### Virtualvault chroot location vs. Security Containment chroot location

<table>
<thead>
<tr>
<th>Virtualvault chroot location</th>
<th>Security Containment chroot location</th>
</tr>
</thead>
<tbody>
<tr>
<td>/var/opt/vaultTS/outside/app/</td>
<td>/var/jail/wp.internet/</td>
</tr>
<tr>
<td>/var/opt/vaultTS/inside/app/</td>
<td>/var/jail/wp.intranet/</td>
</tr>
<tr>
<td>All CGI directories</td>
<td>/var/jail/wp.cgi/</td>
</tr>
<tr>
<td>/var/opt/vvproxy/</td>
<td>/var/jail/wp_webproxy/</td>
</tr>
<tr>
<td>/var/opt/vvproxy2/</td>
<td>/var/jail/wp_webproxy/ 1</td>
</tr>
<tr>
<td></td>
<td>Or</td>
</tr>
<tr>
<td></td>
<td>/var/jail/wp_webproxy2/ 2</td>
</tr>
<tr>
<td>Outside web server chroot directories not in</td>
<td>/var/jail/wp_&lt;server name&gt;/</td>
</tr>
<tr>
<td>/var/opt/vaultTS/outside/app/</td>
<td></td>
</tr>
<tr>
<td>Inside web server chroot directories not in</td>
<td>/var/jail/wp_&lt;server name&gt;/</td>
</tr>
<tr>
<td>/var/opt/vaultTS/inside/app/</td>
<td></td>
</tr>
</tbody>
</table>

1. When only Virtualvault WebProxy 2.0 (/var/opt/vvproxy2/) exists on the Virtualvault system.

2. When Virtualvault WebProxy 1.3 (/var/opt/vvproxy/) and Virtualvault WebProxy 2.0 (/var/opt/vvproxy2/) exist on the Virtualvault system.

### Finding chroot web server directories

To determine if a web server is chrooted, look for the Chroot directive in the corresponding httpd.conf. If the Chroot directive is not commented out (i.e. the line does not start with a pound (“#”) sign), the web server is chrooted. The same applies for WebProxy instances.

Look at the chroot directory specified in the Chroot directive line and use that information to fill in the following template.
Determining number of web server chroot instances to be created

To calculate the approximate space used by the web server chroot instances to be created in the Security Containment system, use the following template:

1. If there is at least one non-chrooted Outside web server or one Outside web server chrooted to /var/opt/vaultTS/outside/app/ in the Virtualvault system, place a 1 in this box, otherwise place a 0. ..............................................

2. If there is at least one non-chrooted Inside web server or one Inside web server chrooted to /var/opt/vaultTS/inside/app/ in the Virtualvault system, place a 1 in this box, otherwise place a 0. ..............................................

3. If Virtualvault Webproxy 1.3 exists (under /var/opt/vvproxy/) in the system, place a 1 in this box, otherwise place a 0. .................................

4. If Virtualvault Webproxy 2.0 exists (under /var/opt/vvproxy2/) in the system, place a 1 in this box, otherwise place a 0. ..............................................

5. If there are Outside web servers chrooted to a directory other than /var/opt/vaultTS/outside/app/, place the number of web servers in this box (i.e. if there are five webservers chrooted to directories other than the default, place a 5), otherwise place a 0. ...............  

6. If there are Inside web servers chrooted to a directory other than /var/opt/vaultTS/inside/app/, place the number of web servers in this box (i.e. if there are five webservers chrooted to directories other than the default, place a 5), otherwise place a 0. .................................  

Total chroot instances:  

Calculating required space for web server chroot instances

Use the following formula to get an approximate amount of space required under `/var` for the chroot web server instances to be moved to the Security Containment system:

Total chroot instances x 300Mb = approximate space required
Appendix H: Updating Completed Transitions

This chapter documents steps that should be taken on completed transition systems to address issues that were identified after specific versions of the transition toolkit were released. A transition is considered completed when the procedures documented in Chapters 1 through 7 have been performed. All issues should be carefully reviewed for applicability to the toolkit used and the indicated actions taken to address the issues.

Each toolkit release is referred to by the date of release and consists of specific versions of each depot and of the Virtualvault Transition to HP-UX with Security Containment Administrator’s Guide. The Administrator’s Guide has the version number listed on its first page. To find the version of an uninstalled depot, use the following command:

```
swlist -a revision -l product @ <host>:<full path of depot file>
```

If the depot is already installed, use the command:

```
swlist -a revision -l product <product>
```

Where <product> is:

- `hp_vvsnapshot` for VVSnapshot.depot
- `hp_hpxharden` for HPUXHarden.depot
- `hp_vv_sc` for VVTransition.depot

April 17, 2006 Release

The April 17, 2006 Release consists of the following depots and documentation:

<table>
<thead>
<tr>
<th>Depot Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPUXHarden.depot</td>
<td>1.2</td>
</tr>
<tr>
<td>VVSnapshot.depot</td>
<td>1.0</td>
</tr>
<tr>
<td>VVTransition.depot</td>
<td>1.2</td>
</tr>
<tr>
<td>VVSCAdminGuide.pdf</td>
<td>1.2.2</td>
</tr>
</tbody>
</table>
March 15, 2006 Release

The March 15, 2006 Release consists of the following depots and documentation:

<table>
<thead>
<tr>
<th>Depot Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPUXHarden.depot</td>
<td>1.0</td>
</tr>
<tr>
<td>VVSnapshot.depot</td>
<td>1.0</td>
</tr>
<tr>
<td>VVTransition.depot</td>
<td>1.01</td>
</tr>
<tr>
<td>VVSCAdminGuide.pdf</td>
<td>1.2.1</td>
</tr>
</tbody>
</table>

January 27, 2006 Release

The January 27, 2006 Release consists of the following depots and documentation:

<table>
<thead>
<tr>
<th>Depot Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPUXHarden.depot</td>
<td>1.0</td>
</tr>
<tr>
<td>VVSnapshot.depot</td>
<td>1.0</td>
</tr>
<tr>
<td>VVTransition.depot</td>
<td>1.0</td>
</tr>
<tr>
<td>VVSCAdminGuide.pdf</td>
<td>1.2</td>
</tr>
</tbody>
</table>

Transitions completed prior to the March 15, 2006 Release

The following issues affect transitions completed prior to the March 17, 2006 Release of the transition toolkit.

IPFilter Related Changes

Earlier versions of the transition toolkit install a version of IPFilter that is not compatible with Security Containment. To ensure correct functioning of IPFilter, complete the following:

1. Download version A.03.05.11 or later IPFilter. Go to http://software/hp.com and search for “IPFilter” to locate the “HP-UX IPFilter” page. Follow the instructions on that page to download depot.

2. Install the IPFilter depot using the following swinstall(1M) command:

   # swinstall -s /download/B9901AA_A.03.05.12_HP-UX_B.11.23_IA+PA.depot */
3. Add IPFilter Rules for each existing web server. For each web server listening to connections from the Internet or Intranet, IPFilter rules must be added to pass the inbound tcp (HTTP) network communication from the out_iface or in_iface interface to the web server HTTP_PORT and SSL_PORT. The IPFilter rules should be added to Bastille’s /etc/opt/sec_mgmt/bastille/ipf.customrules configuration file. Add a separate rule for each listening port. The following example shows IPFilter rules that enable inbound tcp connections on lan1 (out_iface) for ports 80 and 443:

```
pass in quick on lan1 proto tcp from any to any port = 80  keep state  # ows_<ServerID>
pass in quick on lan1 proto tcp from any to any port = 443 keep state  # ows_<ServerID>
```

NOTE: Review the compartment rules file (/etc/cmpt/out_iface.rules or in_iface.rules) to determine which lan interface to use.

4. Run Bastille to activate the new IPFilter rules. For example:

```
# bastille -b
```

5. Download the VVTransition Depot (Version 1.01 or later). The new depot contains updated versions of vvws_createxsec and vvws_removexsec utilities which automate the IPFilter configuration for additional web server creations and removals.


Download the file VVTransition.depot. Optionally, download the file, Readme.txt, which contains a description of the depots and their ckmun(1M) value to verify the integrity of the downloaded file.

6. Install the VVTransition Depot Install the hp_vv_sc.hpvv fileset of the VVTransition depot.

```
# swinstall -s /download/VVTransition.depot \ hp_vv_sc.hpvv
```

This document has been modified to reflect the changes needed to properly support IPFilter. **Chapter 2. Installing the Security Containment System** has been updated to provide instructions for downloading and installing the appropriate version of IPFilter. **Chapter 11. Web Server Transition** has also been modified to
detail the IPFilter rules that are created by the `vvws_createxsec` and removed by `vvws_removexsec`.

Client Certificate Related Changes

Chapter 11. Web Server Transition has been updated to include information on enabling the passing of client certificates to a CGI in the section entitled “Enabling the passing of client certificates to a CGI”.

Transitions completed prior to the April 17, 2006 Release

Updates to the transition toolkit and documentation reflect the changes required for using HP-UX 11i Version 2 Mission Critical Operating Environment March 2006 and the HP-UX 11iV2 March 2006 Software Pack that contain updated packages including Apache 2.0.54, Java 1.5 and Tomcat 5.5.9. No changes to a previously transitioned system are required.