



PowerBroker for Unix & Linux

Installation Guide for:

- PowerBroker for Unix & Linux
- PowerBroker for Networks
- PowerBroker for Sudo
- PowerBroker Essentials

Revision/Update Information: May 2018
Software Version: PowerBroker for Unix & Linux 10.0.1
Revision Number: 0

Ready for

IBM Power
Systems Software

redhat.
R E A D Y

CORPORATE HEADQUARTERS
5090 N. 40th Street
Phoenix, AZ 85018
Phone: 1 818-575-4000

COPYRIGHT NOTICE

Copyright © 2018 BeyondTrust Software, Inc. All rights reserved.

The information contained in this document is subject to change without notice.

No part of this document may be photocopied, reproduced or copied or translated in any manner to another language without the prior written consent of BeyondTrust Software.

BeyondTrust Software is not liable for errors contained herein or for any direct, indirect, special, incidental or consequential damages, including lost profit or lost data, whether based on warranty, contract, tort, or any other legal theory in connection with the furnishing, performance, or use of this material.

All brand names and product names used in this document are trademarks, registered trademarks, or trade names of their respective holders. BeyondTrust Software is not associated with any other vendors or products mentioned in this document.

Contents

- Introduction 9**
 - Conventions Used in This Guide 9
 - Font Conventions 9
 - Linespacing Conventions 10
 - Documentation for PowerBroker for Unix & Linux 10
 - Sample Policy Files 10
 - Contacting Support 10

- Installation Considerations 12**
 - Supported Platforms and Operating Systems 12
 - What Is a Flavor? 12
 - Interactive Versus Packaged Installation 12
 - Interactive and Packaged Installations on the Same Computer 13
 - PowerBroker Express 13
 - SELinux Support 14
 - Resource Overhead 14
 - Disk Space Requirements 14
 - Required Utilities 14
 - Installation Directories 15
 - Default Directories 15
 - Prefix and Suffix Installations 17
 - System File Modifications 18
 - Policy Files 18
 - Role Based Policy Database 19
 - Default Policies 19
 - Helpdesk role 19
 - PBTest 19
 - Controlled Shells 19
 - Admin role 19
 - Demo role 20
 - Network and File Encryption 20
 - Third-Party Libraries 20
 - Using PowerBroker for Unix & Linux Third-Party Libraries Instead of Your Own Libraries 20
 - Third-Party Library File Names and Locations 20
 - Shared Library Directory Location for AIX and HP (PA RISC) 21
 - Shared Library File Name for AIX 21
 - Using Your Own Third-Party Libraries 22
 - Installing Third-Party Libraries in Future Installations 22
 - Release Numbers 22

- Installation Preparation 24**
 - Pre-installation checks 24
 - Obtain a License Validation Key 24
 - Obtain root Access 24
 - Plan PowerBroker for Unix & Linux Hosts 24

Select License Servers	25
Select Submit Hosts	25
Select Run Hosts	25
Select Policy Server Hosts	25
Select Log Hosts	26
Enable Log Synchronization Host	26
Enable GUI Host	26
PowerBroker for Unix & Linux Utilities	26
PowerBroker for Unix & Linux Shells	26
Select Port Numbers	26
Select Installation Directories	27
Select syslog	27
Select Encryption	27
Firewalls	27
Use NIS	27
Verify Proper TCP/IP Operation	28
Verify Network Host Information	28
Installation Process	29
pbinstall Installation Script	29
Basic pbinstall Information	30
Navigating the pbinstall Menu and Choosing Option Values	30
pbinstall Installation Menu Conventions	30
Installation Events Using pbinstall	31
Step-by-Step Instructions for a Basic Installation Using pbinstall	32
Running a Basic Installation Using pbinstall	32
PowerBroker for Unix & Linux pbinstall Installation Menu	35
Prefix and Suffix Installation Instructions	40
Running Prefixed and Suffixed Installations	40
Advanced Installation Instructions Using pbinstall	41
Starting pbinstall	41
Using the Menu Options	41
Completing the Installation	71
Example of a pbinstall Execution	72
Installation Example for PowerBroker Express	74
pbmakeremotetar Installation Script	77
pbmakeremotetar Installation Information	77
Remote Installations Using pbmakeremotetar	78
Example of a pbmakeremotetar Execution	78
Make the Tar File Available to the Remote System	81
Untar the Remote Archive and Execute remote_unpack	81
pbpatchinstall Installation Script	83
pbpatchinstall Installation Information	83
Example of a pbpatchinstall Execution	84
Custom Installations	85
Third-Party Libraries	85
Executable Files	85
Settings File	85
Key File	86

superdaemon Configuration	86
Policy Files for Policy Server Hosts	87
Package Installer	88
Solaris Package Installer	89
Prerequisites	89
Planning Your Installation	89
Choosing a Package Administration File	92
Using PowerBroker for Unix & Linux Packages on Solaris Zones	93
Overview of Steps	93
Installation Procedure	94
Removing PowerBroker for Unix & Linux Packages	96
Relocating the Base Directory	96
Updating PowerBroker for Unix & Linux with the Solaris Package Installer	97
Package Update Considerations	97
Package Update Procedure	97
Sample Execution for the Solaris Package Installer	98
Generating the PowerBroker for Unix & Linux Settings Files	98
Creating the PowerBroker for Unix & Linux Configuration Package Using pbcreatesolcfgpkg	101
Installing Component Packages Using the pkgadd Command	103
Installing the Configuration Package Using the pkgadd Command	106
Sample of the Uninstall Process from a Package Installation	107
Linux Package Installer	109
Prerequisites	109
Planning Your Installation	109
Overview of Steps	113
Installation Procedure	113
Removing PowerBroker for Unix & Linux Packages	116
Relocating the Base Directory	116
Upgrading PowerBroker for Unix & Linux with the Linux Package Installer	116
Package Upgrade Considerations	117
Package Upgrade Procedure	117
Reverting to a Previous Version	118
Sample Execution for the Linux Package Installer	118
Generating the PowerBroker for Unix & Linux Settings Files	118
Creating the PowerBroker for Unix & Linux Configuration Package Using pbcreatelincfgpkg	121
Installing Component Packages Using the rpm Command	123
Installing the Configuration Package Using the rpm Command	124
Sample of the Uninstall Process from a Package Installation	124
AIX Package Installer	126
WPARs	126
Prerequisites	126
Planning Your Installation	126
Using PowerBroker for Unix & Linux Packages on AIX WPARs	130
Overview of Steps	131

Installation Procedure	131
Installing PowerBroker for Unix & Linux onto WPARs	133
Removing PowerBroker for Unix & Linux Packages	133
Removing PowerBroker AIX Package from Shared WPARs	134
Updating PowerBroker for Unix & Linux with Update Packages	134
Update Package Considerations	135
Update Package Procedure	135
Rejecting an Update Package	136
Update Packages and WPARs	136
Sample Execution for the AIX Package Installer	136
Generating the PowerBroker for Unix & Linux Settings Files	136
Creating the PowerBroker for Unix & Linux Configuration Package Using pbcreateaixcfgpkg	140
Installing Component Packages Using the installp Command	142
Installing the Configuration Package Using the installp Command	144
Viewing a List of Installed PowerBroker for Unix & Linux Packages	145
Performing a Cursory Test of PowerBroker for Unix & Linux on the AIX Global Environment	145
Viewing a List of WPARs	146
Using syncwpar to Propagate Additional Packages to Shared WPARs	146
Logging into Shared WPARs	147
Running a Cursory Test of PowerBroker on a Shared WPAR System	148
Sample of Removing an AIX Package Installation	148
Example of Using syncwpar to Propagate Package Removal From Shared WPARs	149
Verifying Removal of PowerBroker for Unix & Linux Packages	151
HP-UX Package Installer	152
Prerequisites	152
Planning Your Installation	152
Overview of Steps	155
Installation Procedure	156
Removing PowerBroker for Unix & Linux Filesets	158
Remote Installation	158
Updating PowerBroker for Unix & Linux with Update Depots	159
Update Depot Considerations	159
Update Depot Procedure	159
Reverting to a Previous Version	160
Sample Execution for the HP-UX Package Installer	160
Generating the PowerBroker for Unix & Linux Settings Files	160
Creating the PowerBroker for Unix & Linux Configuration Package Using pbcreatehpuxcfgpkg	165
Copying the PowerBroker for Unix & Linux Depots Using the swcopy Command	167
Installing the PowerBroker for Unix & Linux Filesets Using the swinstall Command	171
Sample of the Uninstall Process from a Package Installation	174
Installing Multiple Copies	178
Remote Installation Using pbmakerremotetar with Prefixes and Suffixes	178
Program Names and Execution	178
Service Names and Port Numbers	178
NIS(+) Netgroup Names	179
Settings File	179

root Policy Filename	179
Policy File Contents	179
Key File Name	179
Log File Names	179
Man Pages	180
Sample Policy Files	180
Installation Verification	181
Installation Environment Variables	182
Installation Programs	184
pbininstall	184
run_pbininstall	188
pbmakeremotetar	189
pbpatchinstall	191
pbcreateaixcfgpkg	192
pbcreatehpuxcfgpkg	194
pbcreatelincfgpkg	196
pbcreatesolcfgpkg	197
pbuninstall	198
Upgrades and Reinstallations	200
Pre-Upgrade Instructions	200
pbininstall Install Upgrades	201
pbmakeremotetar Install Upgrades and Reinstallations	202
Post-Upgrade Instructions	202
Patch Installations	202
Uninstalling PowerBroker for Unix & Linux	203
Example of a pbuninstall Execution	203
Solr Installations	206
Installation Considerations	206
Supported Platforms	206
PowerBroker for Unix & Linux Solr Java Requirements	206
System Requirements	206
Unix/Linux Utilities	206
System File Modifications	207
SSL Certificates and Search Interface	207
Prerequisites when Installing with BeyondInsight	207
Command Line Options	208
Options for Use with BeyondInsight	208
Options for Use with PSMC	208
Command Options	208
Installation	209
Menu Options	210
Post-Install when Installing with BeyondInsight	210

Re-Installation when Installing with BeyondInsight	212
Solr Uninstall	212
PowerBroker Sudo Installation	213
Supported Platforms	213
Unix/Linux Utilities	213
System File Modifications	213
Prerequisites	214
Installation	215
PowerBroker Sudo Uninstall	216

Introduction

This guide provides detailed information regarding the security policy file programming language for the BeyondTrust PowerBroker® for Unix & Linux software. PowerBroker for Unix & Linux includes PowerBroker® for Unix & Linux - UNIX® Edition, PowerBroker® for Unix & Linux - Linux® Edition, PowerBroker® Virtualization, and PowerBroker® Express. This language is used to create security policy files that are used by PowerBroker® for Unix & Linux to control the following:

- Which tasks a user, or group of users, may perform
- Which machines a task may be submitted from
- On which machines a task may be run
- When a specific task may be run (day and time)
- Where a task may be run from
- Whether or not secondary security checks, such as passwords or checksums, are required to run a task
- Whether or not one or more supplemental security programs are run before a task is started

This manual assumes that the reader has a basic understanding of Unix or Linux system administration and some experience with a scripting or other computer language. It is recommended that you have experience in these areas before you attempt to create or modify security policy files.

Conventions Used in This Guide

Specific font and linespacing conventions are used in this book to ensure readability and to highlight important information, such as commands, syntax, and examples.

Font Conventions

The font conventions used for this document are:

- Courier New Font is used for program names, commands, command arguments, directory paths, variable names, text input, text output, configuration file listings, and source code. For example:

```
/etc/powerbroker/product.cfg
```

- Courier New Bold Font is used for information that should be typed into the system exactly as shown. For example:

```
pbcheck -v
```

- Courier New Italics Font is used for input variables that need to be replaced by actual values. In the following example, *variable-name*, must be replaced by an actual environment variable name. For example:

```
result = getenv (variable-name);
```

- Bold is used for Windows buttons. For example:

Click **OK**.

Linespacing Conventions

The linespacing of commands, syntax, examples, and computer code in this manual may vary from actual Windows and Unix/Linux usage because of space limitations. For example, a single line does not fit within the text margins for this book, the text is displayed on two lines with the second line indented, as shown in the following sample:

```
result = sprintf ("System administrator Ids: %s %s %s", "Adm1", "Adm2",  
                "Adm3");
```

Where to Go Next?

For licensing information and installation instructions for PowerBroker for Unix & Linux, see the PowerBroker for Unix & Linux Installation Guide.

Documentation for PowerBroker for Unix & Linux

The complete PowerBroker for Unix & Linux documentation set includes the following:

- PowerBroker for Unix & Linux Installation Guide
- PowerBroker for Unix & Linux System Administration Guide
- PowerBroker for Unix & Linux Policy Language Guide
- PowerBroker for Unix & Linux Browser Interface Guide
- PowerBroker for Unix & Linux Diagnostic Messages Guide
- Man pages (for Unix/Linux)

Sample Policy Files

When you install PowerBroker for Unix & Linux, you can choose to copy sample PowerBroker for Unix & Linux policy files to the installation host. These sample policy files include detailed explanations of what they do. You can use these files to learn how policy files are typically written for various scenarios. The directory that these sample files are copied to is determined by the GUI library directory option that you specify during installation. By default, this directory is `/usr/local/lib/pbbuilder`. A `readme_samples` text file in that directory includes a brief description of each sample file.

Contacting Support

For support, go to our [Customer Portal](#) then follow the link to the product you need assistance with.

The Customer Portal contains information regarding contacting Technical Support by telephone and chat, along with product downloads, product installers, license management, account, latest product releases, product documentation, webcasts and product demos.

Telephone

Privileged Account Management Support

Within Continental United States: 800.234.9072

Outside Continental United States: 818.575.4040

Vulnerability Management Support

North/South America: 866.529.2201 | 949.333.1997

+ enter access code

All other Regions

Standard Support: 949.333.1995

+ enter access code

Platinum Support: 949.333.1996

+ enter access code

Online

<http://www.beyondtrust.com/Resources/Support/>

Installation Considerations

PowerBroker for Unix & Linux is a non-intrusive software program that does not require kernel reconfiguration or a system reboot and does not replace system executable files. The statements that apply to the PowerBroker for Unix & Linux product also apply to the PowerBroker Express and PowerBroker Virtualization products except where noted. The items in this section contain information you should consider when planning your implementation.

For more detailed information about PowerBroker Express and PowerBroker Virtualization products, see “PowerBroker Express” and “PowerBroker Virtualization” in the PowerBroker for Unix & Linux System Administration Guide.

Supported Platforms and Operating Systems

For information on the platforms and operating systems that are supported by PowerBroker for Unix & Linux, refer to the latest PowerBroker for Unix & Linux README file at www.beyondtrust.com.

What Is a Flavor?

Flavor is a BeyondTrust term that defines a build of a BeyondTrust product, such as PowerBroker for Unix & Linux, that is compiled and tested for a certain range of operating system versions and underlying hardware. For instance, when this guide was written, PowerBroker for Unix & Linux was available in several flavors for Linux operating systems. The README file describes which flavor is the right match for specific combinations of hardware and operating system in the Release Identifier column. The release identifier is the flavor plus the version of the PowerBroker for Unix & Linux distribution. For detailed information about version and release numbers, see [Release Numbers](#).

During installation, the flavor of the PowerBroker for Unix & Linux distribution you are using will be compared to the flavor required for the operating system and hardware version combination you are installing on.

If you believe that you are using the correct version of PowerBroker for Unix & Linux for the machine you are installing on, but the installer is returning a flavor mismatch, then please contact BeyondTrust Technical Support for assistance.

Interactive Versus Packaged Installation

For all flavors PowerBroker for Unix & Linux can be installed by using an interactive program that presents you with a series of options. Your choices determine the details of the installation of PowerBroker for Unix & Linux on a particular host.

The Client Registration facility can be used to automate the installation of new clients by downloading the default configuration from the Primary Policy Server. Options will be defaulted within the interactive installation, and shared encryption keys will be copied over.

For certain flavors, PowerBroker for Unix & Linux and PowerBroker Express can also be installed by using package installers. Package installers enable you to choose the options once and then install that configuration of PowerBroker for Unix & Linux or PowerBroker Express noninteractively on multiple identical hosts. Using package installers also takes advantage of the operating system’s installation management system, which tracks the source of installed files and enables their safe removal.

PowerBroker for Unix & Linux has package installers for the following flavors:

- Solaris 8, 9, and 10 on x86 and SPARC.
- Red Hat Enterprise Linux 3, 4, and 5 on x86, x86_64, ia64, and S/390.
- AIX 5.1, 5.2, 5.3, and 6.1 on a POWER 32-bit or POWER 64-bit computer. AIX package installers are compatible with or without WPARs. WPARs exist only in AIX V6.1 and higher.
- HP-UX 11i v1, 11i v2, and 11i v3.

Interactive and Packaged Installations on the Same Computer

Although it is possible to combine interactive and packaged PowerBroker for Unix & Linux installations on the same computer, this practice is not recommended. If both interactive and packaged installations are present, and you remove the packaged installation, the shared libraries will be removed even though they are needed by the interactive installation. This behavior is inherent in all package installations and is not specific to PowerBroker for Unix & Linux.

In the case of SELinux, if you attempt to perform a package installation on a computer that already has an interactive installation present, the package installation is not allowed. The reason for this limitation is that the SELinux PowerBroker for Unix & Linux packages can fail to install because RPM does not have the permissions to change SELinux file types that are already installed.

If you must combine interactive and packaged PowerBroker for Unix & Linux installations on the same computer, follow these recommendations:

- For the interactive installation, use a prefix/suffix installation (see [Prefix and Suffix Installations](#)).
- Install the shared libraries for the interactive and packaged installations in separate directories, by doing one of the following:
 - In the interactive installation, specify an alternative shared library directory with the BeyondTrust built-in third-party library directory menu item.
 - Use the relocatable base directory feature of the package installer.

PowerBroker Express

PowerBroker Express provides full keystroke logging and firewall capabilities with instantly searchable I/O logging of all privileged activities on servers and devices. For a more detailed description of the product, how it works, and a list of features, see “PowerBroker Express” in the PowerBroker Administration Guide.

You may install PowerBroker Express using the interactive menu on all supported platforms or using a package installer on AIX, Red Hat Linux, HPUX, and Solaris platforms. For more details on the supported platforms, see the PowerBroker for Unix & Linux README file.

The differences between standard PowerBroker for Unix & Linux and PowerBroker Express are as follows:

- PowerBroker for Unix & Linux client pbrun is replaced with pbssh. pbssh is contained in the express package. The express package is not compatible with the PowerBroker for Unix & Linux submit host or run host packages.
- PowerBroker for Unix & Linux run host (pblocald) is not provided. It is contained in the PowerBroker for Unix & Linux run host package.
- PowerBroker for Unix & Linux shells (pbsh and pbksh) are not provided. They are contained in the PowerBroker for Unix & Linux submit host package.

- PowerBroker utilities (pbless, pbmg, pbnvi, pbvi, and pbumacs) are not provided. They are contained in the PowerBroker for Unix & Linux run host package.

For detailed installation instructions, see [pbinstall Installation Script](#) or the chapter with the instructions for the package installer flavor for your installation.

SELinux Support

PowerBroker for Unix & Linux SELinux policies are no longer provided. When installing PowerBroker for Unix & Linux on Red Hat Enterprise Linux (RHEL) 5, with SELinux enabled and using the Targeted policy, PowerBroker binaries will run unconfined.

Resource Overhead

There are no startup or shutdown programs associated with PowerBroker for Unix & Linux. From a system resource perspective, a basic PowerBroker for Unix & Linux session uses about the same overhead as a telnet session, plus a bit at the front end for processing the policy security file. I/O logging can add the equivalent of another telnet session.

Instances of the PowerBroker for Unix & Linux daemons (pbmasterd and pblocald) are requested by pbrun, and are actually started by the superdaemon, when a monitored task request is submitted to pbrun. The superdaemon is inetd, xinetd, launchd, or SMF, depending on the platform. Most of this document references inetd, xinetd, launchd, and SMF interchangeably. A distinction is made only when there are substantive differences.

For systems based on RedHat v7 and later, xinetd is no longer installed by default since it has been superceded by systemd, an init system. The installation program of PowerBroker for Unix & Linux will perform a check to see if systemd exists and is functional. If it exists, it will configure the PowerBroker for Unix & Linux daemons to be managed by systemd. If systemd is not present, the installation program will check if xinetd is installed and running, and display a warning message if it isn't.

Having the superdaemon start pbmasterd and pblocald when requested by pbrun is the normal way to initiate the PowerBroker for Unix & Linux daemons. It is also possible to explicitly start the daemon as a persistent daemon. For more information, see the PowerBroker for Unix & Linux System Administration Guide.

Note: The terms monitored task and secured task are interchangeable.

SSL adds some startup overhead for certificate exchange and verification. The encryption overhead is slightly larger than self-contained encryption technologies (such as DES) because of the use of packet checksums by SSL.

Disk Space Requirements

PowerBroker for Unix & Linux requires 10 to 50 MB of disk space, depending on the installation options that are selected.

Required Utilities

The PowerBroker for Unix & Linux installer requires the following Unix/Linux utilities and built-in commands:

awk	cut	getopt	ps	sort	unset
basename	date	grep	pwd	stty	vi

cat	diff	id	read	tar	wc
cd	dirname	kill	rm	tee	xargs
chmod	df	ls	rmdir	touch	
chown	echo	mkdir	sed	tr	
cksum	eval	more	set	trap	
clear	exec	mv	shift	umask	
cp	export	od	sleep	uname	

Installation Directories

PowerBroker for Unix & Linux is not sensitive about the location of its binary files; you can place them in any convenient directory. However, there are a few points to consider when you are selecting PowerBroker for Unix & Linux installation directories:

- It is important to install the PowerBroker for Unix & Linux pbrun and pbssh programs in a directory that is in the user's path.
- Online manuals (such as user man pages and PowerBroker for Unix & Linux documentation) should be accessible from every computer to enable users to get online help for PowerBroker for Unix & Linux programs.

Default Directories

The following table lists the various PowerBroker for Unix & Linux components and their locations. The PowerBroker for Unix & Linux installation script uses these locations by default, but you can change them during installation. Usually /usr/local/bin is used for user programs and /usr/sbin for administrator and daemon programs (depending on the platform).

Table 1. Default Directories for PowerBroker for Unix & Linux Components

Directory	Files	Description
/etc (v9.4.1 and earlier) /opt/pbul/policies (v9.4.3+)	pb.conf	Default Policy. Includes /etc/pb/pbul_policy.conf (v9.4.1 and earlier) /opt/pbul/policies/pbul_policy.conf (v9.4.3+)
/etc/pb (v9.4.1 and earlier) /opt/pbul/policies (v9.4.3+)	pbul_policy.conf	Main policy containing the following roles: - Helpdesk role - PBTest (connectivity test) - Controlled Shells - Admin Role - Demo Role
/etc/pb (v9.4.1 and earlier) /opt/pbul/policies (v9.4.3+)	pbul_functions.conf	Functions and Procedures implementing the roles in pbul_policy.conf
/opt/pbul/policies (v9.4.3+)	pb.key	Encryption key

	pb.settings	PowerBroker for Unix & Linux configuration file
/usr/adm, /var/adm, or /var/log	pb.eventlog	Default event log file
	pbguid.log	GUI diagnostic log file
	pblocald.log	pblocald diagnostic log file. Not applicable for PowerBroker Express.
	pblogd.log	pblogd diagnostic log file
	pbmastard.log	pbmastard diagnostic log file
	pbrun.log	pbrun diagnostic log file. Not applicable for PowerBroker Express.
	pbssh.log	Used for PowerBroker Express
	pbsguid.log	Secure GUI diagnostic log file
	pbsync.log	pbsync diagnostic log file
	pbsyncd.log	pbsyncd diagnostic log file
/usr/local/bin	pbbench	Utility
	pbcall	Utility
	pbksh	Utility Not available for PowerBroker Express.
	pbless	Utility
	pbmg	Utility Not available for PowerBroker Express.
	pbnvi	Utility Not available for PowerBroker Express.
	pbrun	Utility Not available for PowerBroker Express.
	pbssh	Utility Used for PowerBroker Express
	pbumacs	Utility Not available for PowerBroker Express.
	pbsh	Utility
pbvi	Utility Not available for PowerBroker Express.	
/usr/local/lib /pbbuilder		Contains the various GUI and pbguid components. Do not make any changes in this directory.
/usr/sbin	pbcheck	Utility
	pbdbutil	Utility providing Powerbroker database maintenance.

	pbencode	Utility
	pbguid	Daemon
	pbkey	Utility
	pblocald	Daemon Not available for PowerBroker Express.
	pblog	Utility
	pblogd	Daemon
	pbmasterd	Daemon
	pbmerge	Utility
	pbpasswd	Utility
	pbreplay	Utility
	pbsum	Utility
	pbsync	Utility
	pbsyncd	Daemon
	pbversion	Utility
/opt/pbul/dbs	pbsudo.db	Database files generated and used by PowerBroker Unix & Linux
	pbsvc.db	
	pbsvccache.db	
	pbdbsync.db	
	pbrgclnt.db	
	pbrbpolicy.db	
	pbevent.db	
	pbfim.db	
	pbrstkeys.db	
	pblogarchive.db	
	pblogcache.db	

The default log directory varies by platform to match that platform’s conventions. The directories /usr/adm, /var/adm, and /var/log are used interchangeably throughout this document as the default location of the PowerBroker for Unix & Linux log files.

Prefix and Suffix Installations

PowerBroker for Unix & Linux can be installed with prefixes and/or suffixes to create unique installations for multiple installs or for ease of identification. For instructions about using prefixes and suffixes for a PowerBroker for Unix & Linux installation, see [Prefix and Suffix Installation Instructions](#).

Note: Prefixes and suffixes cannot be used with any of the package installers.

System File Modifications

PowerBroker for Unix & Linux does not replace any Unix/Linux files or binaries during installation, but it does modify the following system files:

- `/etc/inetd.conf` (or `xinetd.conf`, `launchd`, `systemd` or SMF configuration file)
- `/etc/services`

These files are automatically backed up as files with the same name and the following extension:

`.sybak####`

The changes that are made to these files depend on whether a Policy Server host, run host (not available for PowerBroker Express), GUI host, log synchronization host, or log host is being installed. Depending on the selected installation options, each file has lines removed, added, or both.

For `/etc/inetd.conf` (or your `xinetd.conf`, `launchd`, or SMF configuration), the installer tries to precisely determine the superdaemon configuration file that is used on the active system. Most systems use the superdaemon's default configuration file name while the rest of the systems use a switch or command line format which makes it possible to determine the superdaemons configuration files to configure.

`xinetd` uses `/etc/xinetd.conf` and any specified `includedir` file directories.

Prior to version 6.0, PowerBroker for Unix & Linux used `xinetd` for the Mac OS X 10.4 operating system. Starting with version 6.0, PowerBroker for Unix & Linux uses `launchd` for Mac OS X 10.4. (PowerBroker for Unix & Linux has always used `launchd` for Mac OS X 10.5.) For Mac OS X 10.4, if PowerBroker for Unix & Linux is upgraded from an earlier release to version 6.0, the PowerBroker for Unix & Linux `xinetd` configuration is removed and replaced with the PowerBroker for Unix & Linux `launchd` configuration. Removal of earlier releases of PowerBroker for Unix & Linux with PowerBroker for Unix & Linux version 6.0 checks for and removes the PowerBroker for Unix & Linux `xinetd` configuration.

Note: For Mac OS X 10.4.7 and later, starting `launchd` jobs using `launchctl` can result in the following error message:

Workaround Bonjour: Unknown error: 0

This message can be ignored. The problem is fixed in Mac OS X 10.5.

SMF is used on Solaris 10 and later, and uses a configuration database.

Starting with version 7.1.0, if the system PowerBroker for Unix & Linux is being installed on is IPv6-capable and configuration of `inetd`, `xinetd`, SMF (Solaris), is being performed, the super daemon configuration will be set for IPv6 rather than IPv4.

Policy Files

`/opt/pbul/policies/pb.conf` (from v9.4.3+, and `/etc/pb.conf` prior to v9.4.3) is usually the root or entry point to the PowerBroker for Unix & Linux policy tree. Although `pb.conf` can contain actual policy code, it is recommended that you use it strictly as a list of include statements that reference other policy modules. Referencing other policy modules in the `pb.conf` file keeps a large policy tree manageable. For more information about policy files, see the PowerBroker for Unix & Linux Policy Language Guide.

Role Based Policy Database

With the introduction of v9 of PowerBroker for Unix & Linux, there is a new Role Based Policy Database. Role Based Policy has been implemented to simplify the definition of policy for administrators. Policies are kept within structured records in a database, simplifying maintenance, decreasing system load, increasing throughput, and providing a comprehensive REST API to integrate policy management with existing customer systems and procedures, including simplified bulk import/export of data. Once the customers data is held within the Role Based Policy database it is much easier to provide management information, such as user entitlement reports. This can be used instead of Policy Script configuration to quickly and succinctly define, retrieve and report on role-based policy. See "PowerBroker for Unix & Linux Role Based Policy" in the PowerBroker for Unix & Linux Administration Guide.

Default Policies

Starting with version 8.0.0, a default policy will be installed by default if an existing policy does not exist. The files 'pbul_policy.conf' and 'pbul_functions.conf' will be created in a /opt/pbul/policies directory (from v9.4.3+ and in /etc/pb prior to v9.4.3) by default. 'pbul_policy.conf' will then be included in the main policy (by default /opt/pbul/policies/pb.conf from v9.4.3+ and /etc/pb.conf prior to v9.4.3).

This default policy contains the following roles:

Helpdesk role

Enabled by default, when invoking "pbrun helpdesk" it allows any user in HelpdeskUsers (default 'root') to initiate a Helpdesk Menu as 'root' on any host in HelpdeskHosts (default submithost only). Helpdesk Menu of actions contains:

- List of processes (ps -ef)
- Check if a machine is up (ping <host>)
- List current users on this host (who -H)
- Display Host's IP settings (ifconfig -a)

PBTest

Enabled by default, for all users on all hosts, "pbrun pbtest" allows checking connectivity and policy.

Controlled Shells

Enabled by default, allows users in ControlledShellUsers (by default the submituser), for runhosts in ControlledShellHosts (by default only submithost), to enable iologging for pbksh/pbsh. iologs are created by default in "/tmp/pb.<user>.<runhost>.<YYYY-MM-DD>.[pbksh|pbsh].XXXXXX". This role has a list of commands (empty by default) to elevate privileges for, as well as a list of commands (empty by default) to reject.

Admin role

Enabled by default, allows users in AdminUsers (by default 'root') to run any command on runhosts in AdminHosts (by default only submithost).

Demo role

Disabled by default, allows users in DemoUsers (default all users) to run commands in DemoCommands (default 'id' and 'whoami') as 'root' on any host in DemoHosts (default all hosts).

The policy ends by allowing all users to run any command as themselves without any privilege escalation.

Network and File Encryption

PowerBroker for Unix & Linux can encrypt data to guard against attack. Several encryption modes are supported. The installation script uses the pbkey program to create an encryption key in the key file, by default /etc/pb.key. This file must then be placed on all PowerBroker for Unix & Linux machines in a PowerBroker for Unix & Linux installation.

Because the pb.settings file is required to be in the /etc directory, if the pb.settings file is encrypted, then the key file that is used to encrypt it must also be in the /etc directory.

Note: A key file can be added to the installation when using pbinstall. For more information about the key file, see “Installation Process,” page 28. For more information about encryption, see “Network Traffic and File Encryption” in the PowerBroker for Unix & Linux System Administration Guide.

For information about Kerberos and SSL see “Secure Socket Layers and Public Key Infrastructure” and “Kerberos V5” in the *PowerBroker for Unix & Linux System Administration Guide*.

Third-Party Libraries

When PowerBroker for Unix & Linux is configured with Kerberos, SSL, LDAP, or CURL it requires the appropriate third-party libraries. The PowerBroker for Unix & Linux installation provides Kerberos, SSL, LDAP, or CURL libraries that are designed to work with PowerBroker for Unix & Linux. It is recommended that you install the PowerBroker for Unix & Linux third-party libraries. However, you have the option of using your own third-party libraries as discussed in the following section.

Caution! Shared libraries can be adversely affected when both interactive and packaged PowerBroker for Unix & Linux installations are present on the same computer. See [Interactive and Packaged Installations on the Same Computer](#).

Note: PowerBroker for Unix & Linux does not currently support shared libraries for the following operating systems: AIX 4.3, NCR, IRIX, OSF, QNX, and Mac OS X

Using PowerBroker for Unix & Linux Third-Party Libraries Instead of Your Own Libraries

If you have your own Kerberos, SSL, LDAP, or CURL libraries and you do not use them or you would rather use the PowerBroker for Unix & Linux third-party libraries, you should do one of the following:

- Remove your libraries from /usr/lib (or /lib) and point to the PowerBroker for Unix & Linux third-party libraries in /usr/lib/beyondtrust/pb (/usr/lib/symark/pb for version prior to 8.0) in pb.settings.
- Replace your third-party libraries with the PowerBroker for Unix & Linux third-party libraries in /usr/lib (or /lib) and specify this directory in pb.settings.

Third-Party Library File Names and Locations

If you are installing PowerBroker for Unix & Linux shared libraries, then the following files are installed:

For Kerberos:

```
llibcom_err.so.3.0
libk5crypto.so.3.1
libkrb5support.so.0.1
libkrb5.so.3.3
libgssapi_krb5.so.2.2
```

For SSL:

```
libcrypto.so.1.0.0
libssl.so.1.0.0
```

For LDAP:

```
liblber-2.4.so.0.2
libldap-2.4.so.0.2
```

For CURL:

```
libcurl.so.4
```

Shared Library Directory Location for AIX and HP (PA RISC)

For AIX and HP (PA-RISC), the directory for installing third-party libraries must be in one of the following locations:

- /usr/lib/beyondtrust/pb (/usr/lib/symark/pb for version prior to 8.0)
- /usr/lib
- /lib
- /usr/local/lib

If any other directory is specified, then it is rejected with an error message that instructs you to use one of these four directory locations.

Shared Library File Name for AIX

The notation that is used on AIX to specify LDAP libraries is different from other platforms. On AIX, for third-party libraries that are archives, you also need to specify the shared object that is a member of the archive and add it to the file name.

The notation for default LDAP libraries is:

```
/usr/lib/beyondtrust/pb/liblber-2.4.a(liblber-2.4.so.2.10.3)
/usr/lib/beyondtrust/pb/libldap-2.4.a(libldap-2.4.so.2.10.3)
```

For example if libcom_err.a.3.0 is an archive and shr.0.3.0 is the actual shared object, the file specification for the member of the archive is:

```
libcom_err.a.3.0(shr.0.3.0)
```

Note: For SSL and Kerberos, because the library is not an archive, it is not necessary to alter the file name.

Using Your Own Third-Party Libraries

If you have chosen to configure PowerBroker for Unix & Linux with Kerberos, SSL, or LDAP, and do not load PowerBroker for Unix & Linux built-in third-party libraries, you must specify your own shared library file names. If you have Kerberos, SSL, or LDAP libraries of your own in `/usr/lib` or `/lib` and you are using them for other applications, you need to use your libraries for PowerBroker for Unix & Linux as well and not use any of the libraries in `/usr/lib/beyondtrust/pb` (`/usr/lib/symark/pb` for version prior to 8.0). Therefore, during the PowerBroker for Unix & Linux installation, specify `no` for the install option: Install BeyondTrust built-in libraries, and then enter the appropriate shared library directory and filename.

For more information about the installation instructions, see [Advanced Installation Instructions Using `pbinstall`](#).

Installing Third-Party Libraries in Future Installations

If you do not enable the third-party libraries during the PowerBroker for Unix & Linux installation and in the future you decide to enable Kerberos or SSL, or use LDAP in your PowerBroker for Unix & Linux policy, then you must do the following:

1. Install PowerBroker for Unix & Linux third-party libraries or your own third-party libraries.
2. In the `pb.settings` file, do one of the following:
 - If you are using the PowerBroker for Unix & Linux third-party libraries, then specify the directories to install the operating system third-party libraries in by setting the following keywords to specify the full path and library file names:

```
sharedlibkrb5dependencies
sharedlibssldependencies
sharedlibldapdependencies
sharedlibcurldependencies
```
 - If you are using your own third-party libraries, then do the following:
 - Specify the Kerberos library setting and provide the full path and library file names.
 - Specify the SSL library setting and provide the full path and library file names.
 - Specify the LDAP library setting and provide the full path and library file names.
 - Specify the CURL library setting and provide the full path and library file names.
 - Ensure that your libraries are listed in the correct order. For example, if `lib1` is dependent on `lib2`, you must list `lib2` first, followed by `lib1`.

Release Numbers

BeyondTrust product releases are uniquely identified by a string that indicates their hardware and software characteristics. This string contains the following information:

- BeyondTrust product
- Hardware architecture
- Flavor (BeyondTrust term that reflects the architecture and operating system)
- Major version number
- Minor version number

- Release number
- Build number
- Service pack number

An example version number in the extracted tarball directory path is: `pbx86_linuxA-5.1.2-03-sp1`

in which:

`pb` is the BeyondTrust product (PowerBroker for Unix & Linux; `pbx_` indicates PowerBroker Express)

`x86` is the hardware architecture

`linuxA` is the flavor

5 is the major version number 1 is the minor version number 2 is the release number

03 is the build number

`sp1` is the service pack number

The functionality of releases within the same version is the same. Releases within a version denote new ports and fixes. The Release Notes for a release describe the issues that are addressed by the release.

Installation Preparation

This section lists the items that you need to plan for and be aware of before beginning your PowerBroker for Unix & Linux installation.

Pre-installation checks

`pbulpreinstall.sh` performs some basic pre-installation checks such as hostname resolution, DNS and name services resolution, verifying that the default ports are not in use, and checking for sufficient disk space. This also reports technical support related information such as the Operating System, NIC information, gateway, and super daemon status. If PowerBroker for Unix & Linux is already installed, this reports the PowerBroker for Unix & Linux roles (submithost, runhost, Policy Server, logserver, pbx).

This script has an optional `-t <datetime in UTC>` argument, which will initiate a time verification check. This check simply validates that the host's time is within 60 seconds of the time specified. The time specified must be UTC, in the format 20130827154130, such as produced by:

```
date -u '+%Y%m%d%H%M%S'
```

This script has an optional `-f` argument, which will cause `pbulpreinstall.sh` to produce machine readable output intended for the PowerBroker for Unix & Linux Management Console (PBSMC) installation console.

Prior to installation, the `pbulpreinstall.sh` script is located in the PowerBroker for Unix & Linux distribution in the following directory:

```
powerbroker/<version>/<flavor>/install
```

This script is installed in the '\$inst_admin' directory (`/usr/sbin` by default) after the install.

Obtain a License Validation Key

To install PowerBroker for Unix & Linux, you need a license string, which is provided by your BeyondTrust sales representative.

PowerBroker for Unix & Linux Primary License Server hosts perform the license resolution functions for PowerBroker for Unix & Linux and are the only PowerBroker for Unix & Linux host types that require a license key. For a Policy Server host to accept a task, the Primary License Server must have a current valid license key. The distribution includes a temporary license key with a two-month expiration date from the date of the installation.

If installing using `pbinstall`, the license key may be configured during installation using the PowerBroker for Unix & Linux License installation menu item. After the installation is complete, the PowerBroker for Unix & Linux license can also be added using the `"pbadmin --lic -u"` command.

Obtain root Access

Installation of the PowerBroker for Unix & Linux product requires root access.

Plan PowerBroker for Unix & Linux Hosts

A PowerBroker for Unix & Linux installation includes several host types, each of which performs specific functions. Prior to installation, you need to determine which host type needs to be placed on the individual machines in your environment.

Note: PowerBroker for Unix & Linux must be installed separately on each machine that will run any type of PowerBroker for Unix & Linux host.

Select License Servers

Determine which hosts to use as License Servers, the machines that perform the license resolution functions for PowerBroker for Unix & Linux. These hosts are the only types that require a license key. They store and maintain the product license, parameters, and usage information.

The first installation of PowerBroker for Unix & Linux becomes the Primary License Server. Subsequent License Server installations will obtain their data when the Primary License Server performs synchronization.

Select Submit Hosts

Select Submit Hosts determines which machines to use as submit hosts, the machines where pbrun is installed and executed. pbrun is the PowerBroker for Unix & Linux utility used to submit secure tasks that might run on the same or different hosts. At least one submit host must be available to process monitored task requests.

Note: For PowerBroker Express, submit hosts installs and uses pbssh instead of pbrun.

Select Run Hosts

Determine which machines to use as PowerBroker for Unix & Linux run hosts, the machines where pblocald, pbsh, and pbksh are installed and executed. pblocald is the daemon process that executes secure tasks. At least one run host must be available to process accepted task requests.

Multiple PowerBroker for Unix & Linux components can be installed on a single machine. For example, it is possible for a single physical machine to serve as a submit host, Policy Server host, run host, log host, log sync host, and GUI host.

Note: Run hosts are not available for PowerBroker Express.

Select Policy Server Hosts

Determine which machines to use as PowerBroker for Unix & Linux Policy Server hosts, the machines where pbmasterd is installed and executed. pbmasterd is the daemon process that accepts or rejects all tasks that are submitted by submit hosts, and if accepted, authorizes a specific run host to execute each task. The Policy Server host is also the location of the policy file (by default /opt/pbul/policies/pb.conf from v9.4.3+ and /etc/pb.conf prior to v9.4.3) and any other policy files that are referenced by include statements in the policy file.

There must be at least one Policy Server host in a PowerBroker for Unix & Linux installation. It is recommended that a second, failover Policy Server host also be installed (and have the same policy files as the primary Policy Server host) to give redundancy to your PowerBroker for Unix & Linux installation.

Depending on the size of your PowerBroker for Unix & Linux environment, or the volume of tasks that are executed through the PowerBroker for Unix & Linux system, it may be desirable to add additional PowerBroker for Unix & Linux Policy Server hosts to your PowerBroker for Unix & Linux installation. Additional PowerBroker for Unix & Linux Policy Server hosts can be added during the initial installation of PowerBroker for Unix & Linux, or afterward as your load dictates.

Select Log Hosts

Using a log host to record event and I/O logs is optional. To use this feature, determine which machine (or machines) to use as PowerBroker for Unix & Linux log hosts, the machines where `pblogd` is installed and executed. For PowerBroker for Unix & Linux if a log host is not used, then `pbmasterd` and `pblocald` are responsible for logging activities. For PowerBroker Express if a log host is not used, then `pbmasterd` is responsible for logging activities. As with Policy Server hosts, multiple log hosts are recommended to provide redundancy. When there is a log host failover, the log synchronization utilities in PowerBroker for Unix & Linux can be used to resynchronize the log entries.

The load on the log hosts varies with the amount of logging that is performed. I/O logs require greater resources on the log hosts. Additional log hosts can be added to your environment during installation, or afterward as needed.

Enable Log Synchronization Host

Log synchronization enables a log host, or a Policy Server host that is acting as a log host, to participate in log synchronization. Install the log synchronization component on any log host or Policy Server host that may participate in log synchronization. Log synchronization should be installed on each log/Policy Server host if you are installing primary and failover log hosts, or are installing Policy Server hosts that are acting as log hosts.

If log synchronization is used, then one or more machines need to have the ability to initiate log synchronization.

Enable GUI Host

Using a GUI host is optional. The PowerBroker for Unix & Linux GUI is a Web interface for administering the Policy Server hosts and log hosts, and the PowerBroker for Unix & Linux settings file, `/etc/pb.settings`. The GUI host can maintain PowerBroker for Unix & Linux components only on the same machine as where the GUI host is installed. The GUI host can be configured to use the HTTP protocol or the HTTPS protocol. When used with HTTPS, the Web interface is called the secure GUI host.

PowerBroker for Unix & Linux Utilities

Using the PowerBroker for Unix & Linux utilities is optional. The PowerBroker for Unix & Linux utilities are secured versions of `vi`, `nvi`, `mg`, `umacs`, and `less`. The PowerBroker for Unix & Linux utilities can only be installed on a machine where a PowerBroker for Unix & Linux run host is installed.

Note: PowerBroker for Unix & Linux utilities are not available for PowerBroker Express.

PowerBroker for Unix & Linux Shells

Using the PowerBroker for Unix & Linux shells is optional. The PowerBroker for Unix & Linux shells are secured versions of the Korn Shell and the Bourne Shell. The PowerBroker for Unix & Linux shells can be installed only on a machine where a PowerBroker for Unix & Linux submit host is installed.

Note: PowerBroker for Unix & Linux shells are not available for PowerBroker Express.

Select Port Numbers

You need to decide whether to use the PowerBroker for Unix & Linux default port numbers or to specify your own. PowerBroker for Unix & Linux uses the following default port numbers:

```
pbmasterd24345
pblocald 24346
```

```
pblogd 24347
pbguid24348
pbsguid 24349
pbsyncd 24350
```

If you decide to change the port number defaults, be sure to choose port numbers that do not conflict with those already in use. See `/etc/services`. Also, if present and active, review the services NIS map. PowerBroker for Unix & Linux port numbers must use the non-reserved system ports. The allowed port numbers are 1024 to 65535.

Select Installation Directories

Decide whether to use the PowerBroker for Unix & Linux default installation directories or to specify your own. Specifying your own installation directories allows for PowerBroker for Unix & Linux optimization of the local installation.

Select syslog

Use of syslog is optional. Determine if the Policy Server host, run host, submit host, GUI host, log sync host, and/or log host should generate syslog records when system error conditions are encountered.

Select Encryption

Starting in version 8.0, by default, PowerBroker for Unix & Linux installs with aes-256 encryption (prior to 8.0, the default was DES); however, it can support a large number of encryption technologies. In PowerBroker for Unix & Linux V3.0 and earlier, DES and 3DES are supported. Beginning with PowerBroker for Unix & Linux V3.2, many additional encryption modes are supported.

Prior to selecting which encryption technology you plan to use, see “Network Traffic and File Encryption” technologies in the PowerBroker for Unix & Linux Administration Guide.

Firewalls

PowerBroker for Unix & Linux can be used in a firewall environment with special configuration. If you are installing PowerBroker for Unix & Linux into an environment where the PowerBroker for Unix & Linux components need to communicate across firewalls, see “Firewalls” in the PowerBroker for Unix & Linux Administration Guide before installing.

Use NIS

PowerBroker for Unix & Linux can use NIS to provide configuration services for PowerBroker for Unix & Linux settings. Netgroups can be defined for the Accept Policy Servers (pbacceptmaster), Submit Policy Servers (pbsubmitmasters) and Log Host (pblogservers) settings. NIS can also be used to provide port lookup information for the PowerBroker for Unix & Linux components. If NIS is running in your environment, consider using PowerBroker for Unix & Linux netgroups and port definitions.

Verify Proper TCP/IP Operation

PowerBroker for Unix & Linux uses TCP/IP as its communication protocol. Therefore, it is essential that TCP/IP be working correctly before PowerBroker for Unix & Linux installation. Use programs such as ping, netstat, route, or traceroute to verify correct TCP/IP operation among all hosts that will have PowerBroker for Unix & Linux components installed.

Verify Network Host Information

Ensure that each network host knows the names and addresses of all other network hosts. Network host information is generally stored in the `/etc/hosts` file on each network host machine or in the NIS maps or DNS files on a server. Each submit host should resolve all of the Policy Server host names correctly. Each Policy Server host should resolve all submit, run, log, and GUI host names correctly. The resolution must work correctly in both directions: name-to-IP address and IP address-to-name.

After installation, the `pbbench` utility generates warnings for any host name resolution issues on a host where PowerBroker components are installed.

Installation Process

PowerBroker for Unix & Linux supports three interactive installation methods and three package installation methods. Before you choose which installation method to use, it is recommended that you review the indicated section.

- Using `pbinstall` - `pbinstall` is a command-line script that can be used to install (or upgrade) PowerBroker for Unix & Linux. It enables the user to review each setting during the installation process and customize the PowerBroker for Unix & Linux installation on that host. See [pbinstall Installation Script](#).

In addition, `pbinstall` command line script can be also used to install PowerBroker Express. To view an example of the PowerBroker Express installation menu options, see [Installation Example for PowerBroker Express](#).

A wrapper script, `run_pbinstall`, is available to simplify installation of all PBUL components. See [run_pbinstall](#).

- Using `pbmakeremotetar` - `pbmakeremotetar` enables you to clone a PowerBroker for Unix & Linux installation on a host across other hosts. `pbmakeremotetar` is effective when you have multiple systems that are running the same PowerBroker for Unix & Linux flavor and are to be configured identically. See [pbmakeremotetar Installation Script](#).
- Using `pbpatchinstall` - `pbpatchinstall` enables you to install PowerBroker for Unix & Linux patches on installations of PowerBroker for Unix & Linux V4.0 and higher. See [Example of a pbpatchinstall Execution](#).
- Using package installers - For Solaris, Linux, and AIX, you can install PowerBroker for Unix & Linux using package installers. See [Solaris Package Installer](#), [Linux Package Installer](#), and [AIX Package Installer](#). You can also install PowerBroker Express using a package installer.

Note: In addition, you can customize your installation (see [Custom Installations](#)).

pbinstall Installation Script

This section contains the following information about the `pbinstall` script:

- Basic `pbinstall` information. Includes `pbinstall` script location, required privileges, editor requirements, environment variables, and so on. See [Basic pbinstall Information](#).
- `pbinstall` installation menu. See [PowerBroker for Unix & Linux pbinstall Installation Menu](#).
- Navigation tools for `pbinstall`. See [Navigating the pbinstall Menu and Choosing Option Values](#).
- Basic installation. Includes step-by-step instructions for a basic PowerBroker for Unix & Linux installation using `pbinstall`. See [Step-by-Step Instructions for a Basic Installation Using pbinstall](#).
- Client Registration configuration. Documents the procedure for setting up a Client Registration Server, and how to use the `pbinstall` option to automatically configure new client installs. See [Running a Basic Installation using pbinstall](#). Additionally, refer to the PowerBroker for Unix & Linux System Administration Guide for more information on Client Registration.
- Advanced installation. Includes an explanation for each of the options for a PowerBroker installation using `pbinstall`. See [Advanced Installation Instructions Using pbinstall](#).
- PowerBroker Express installation. Includes an example of the installation menu when installing PowerBroker Express. The menu you see may differ depending on the options you select during installation. See [Installation Example for PowerBroker Express](#).

- Example execution of pbinstall. See [Example of a pbinstall Execution](#).

Basic pbinstall Information

The following list provides basic information about the pbinstall script:

- The pbinstall script is located in the PowerBroker for Unix & Linux distribution in the following directory:
`powerbroker/<version>/<flavor>/install`
- pbinstall can be run from a PowerBroker for Unix & Linux distribution CD or from an unpacked tar file. The pbinstall install script guides you through the installation and enables you to specify which PowerBroker for Unix & Linux components to install.
- Run pbinstall on each machine that needs PowerBroker for Unix & Linux components installed.
- Superuser authority is required to run pbinstall. Before running pbinstall, either log on as root or use the su command to acquire root privileges.
- pbinstall can be run with various options. See [pbinstall](#) for more information about these options.

Navigating the pbinstall Menu and Choosing Option Values

The pbinstall script presents options in a numbered menu. Because of the large number of options, the menu is divided into pages. You use the navigation characters that are listed below to navigate the pages. To use a navigation character, type the character and press Enter.

The navigation character are the following:

- C Continue installation
- N Next menu page
- P Previous menu page
- R Redraw menu (not shown due to space limitations)
- X Exit script without performing any configuration

To set the value of a menu option, type the number for that option and press Enter. Specify the value for the option and press Enter. For Yes/No options, you can specify N, n, Y, or y.

You might also see the following prompts, which are synonymous:

```
Press return to continue
Hit return to continue
```

Review the messages preceding these prompts on the screen and press Return, Enter, <carriage return>, or <line feed> for the installation process to continue.

pbinstall Installation Menu Conventions

Conventions for the pbinstall installation menu include the following:

- Some options are displayed only if other options or the system configuration allow them.
- The item numbers vary with the configuration of the installation target system.
- The step numbers for the basic PowerBroker for Unix & Linux installation instructions do not necessarily match the option numbers in the pbinstall installation script.

- If the current value of an option forces the line to be longer than 80 characters, then the value within the square brackets is truncated and appended with ellipses (...).
- Menu pages are limited to a maximum of 18 items. To view additional options, use the navigation characters: N (for next page) or P (for previous page).
- The values that are shown in the installation menu are examples and not necessarily the defaults or recommended values for your system. Your defaults and existing values (on a re-installation) will appear in the appropriate places when pbinstall executes.
- Yes and No answers are not case-sensitive and may be abbreviated as y and n.
- pbinstall is designed for 24 line by 80 column displays. Using a larger display is also supported. pbinstall does not support smaller displays.
- Although white space, line terminators, and shell (sh) meta characters are usually allowed in file and directory names, PowerBroker for Unix & Linux does not support them. Do not use them in PowerBroker for Unix & Linux file or directory names.
- The sections that provide installation instructions for pbinstall are:
 - For a basic installation, see [Step-by-Step Instructions for a Basic Installation Using pbinstall](#).
 - For more advanced installation options, see [Advanced Installation Instructions Using pbinstall](#).
 - For an example of the installation menu for a PowerBroker Express install, see [Installation Example for PowerBroker Express](#).
- For more detailed information about PowerBroker for Unix & Linux settings, see the *PowerBroker for Unix & Linux System Administration Guide*.

Installation Events Using pbinstall

When pbinstall runs, the following actions occur:

- If Client Registration is used:
 - The /etc/pb.settings file is downloaded from the Primary Policy Server
 - The /etc/pb.key (or equivalent) is downloaded from the Primary Policy Server
 - If SSL is enabled the SSL Server certificates are downloaded from the Primary Policy Server
- REST Services daemon (pbconfigd) is installed and configuration made to the operating system to enable service management through the native operating system service manager.
(**Note:** REST Services are not fully supported on Mac OS X.)
- The /etc/pb.settings file is created. It contains various parameters and settings that PowerBroker for Unix & Linux uses at run time. PowerBroker for Unix & Linux cannot run without this file.
- The installation process also creates a work file, /etc/pb.cfg. The pb.cfg file is used to locate the PowerBroker for Unix & Linux components during upgrades and uninstalls.
- The /etc/pb.key file is created. It stores the encryption key. This step is completed only if the PowerBroker for Unix & Linux encryption option was selected.
- If you have chosen to add entries to /etc/services, then the following two steps are performed:
 - The /etc/services file is backed up to: /etc/services.sybak.####

The installation script backs up files using the name format {original_name}.sybak.####, where #### is a number between 0000 and 9999. By default, up to 10 of these files are kept in the directory. This backup method is especially advantageous when performing multiple installations and uninstalls.

- Entries are added to the /etc/services file for pbmasterd, pblockd, pblogd, pbguid, and pbsguid. (pblockd is not available for PowerBroker Express.)
- If the system uses inetd.conf for superdaemon configuration, then the following three steps are performed. If the system uses xinetd.conf, then similar steps are performed.
 - The /etc/inetd.conf file is backed up to a file called: /etc/inetd.sybak.####.
 - Entries are added to the inetd.conf file. These entries enable inetd to start instances including:
 - pbmasterd - to validate a monitored task request.
 - pblockd - to execute a monitored task request that has been accepted by pbmasterd. pblockd is not available for PowerBroker Express.
 - pblogd - to perform logging.
 - pbguid - to run the PowerBroker for Unix & Linux administration GUI.
 - pblighttpd - to run PowerBroker REST services.
(**Note:** REST Services are not fully supported on Mac OS X)
 - The inetd superdaemon restarts.
- The appropriate PowerBroker for Unix & Linux programs and online man pages are copied to the specified installation directories.
- During the installation, you have the option to view the generated install script. This option is only for troubleshooting by BeyondTrust Technical Support; the generated install script contains thousands of lines of code.

Step-by-Step Instructions for a Basic Installation Using pbinstall

The basic pbinstall procedure assumes that you have successfully downloaded and unarchived the PowerBroker for Unix & Linux distribution or have a PowerBroker for Unix & Linux CD (see [pbinstall Installation Script](#)).

For additional information about PowerBroker for Unix & Linux components and more options for pbinstall, see the following:

- For a description of the basic PowerBroker for Unix & Linux components that can be installed on this host in step 7, see [Installation Preparation](#).
- For details about the additional options available during a pbinstall installation, see [Advanced Installation Instructions Using pbinstall](#).

Running a Basic Installation Using pbinstall

To perform a basic PowerBroker for Unix & Linux installation using the pbinstall script, use the following procedure:

1. If you downloaded PowerBroker for Unix & Linux using the Web or FTP, then do the following. (To install PowerBroker for Unix & Linux from a CD, skip to step [2](#).)
 - a. Create the /opt/beyondtrust directory if it does not already exist.
 - b. Extract the PowerBroker for Unix & Linux installation files by executing the following command:


```
gunzip -c pb<flavor_version>.tar.Z | tar xvf -
```

- To install from a CD, insert it into the CD-ROM drive on your machine. Mount the CD by entering:

```
mount /cdrom <device_name>
```

Your system may require additional command options or have a different mount point. For more information, see the mount man page for your system.

- Navigate to the appropriate install directory on the file system or CD.
- Start the pbinstall script with the following command:

```
./pbinstall
```

- Press Enter after you read the initial messages.

- A prompt will ask if this is the first installation in the enterprise:

```
PowerBroker for Unix & Linux must have a designated Primary Server
to provide control and consistency for all its components/entities.
The Primary Server must be installed and configured first before all other
hosts.
```

```
Is this the first installation in the enterprise (designated Primary
Server) [yes]?
```

- If you are installing a new client you may wish to use the Client Registration facility. When first invoking pbinstall, the following is displayed:

```
Client Registration provides a method of automatic configuration based upon
a profile provided by your Primary Policy Server. To use this
functionality you will need to know specific parameters from your
Primary Policy Server setup. Please see the installation guide for details.
```

```
Do you wish to utilize Client Registration? [no]? yes
```

```
Enter the Application ID generated on the Primary Policy Server: appid
```

```
Enter the Application Key generated on the Primary Policy Server: 0b5e954e-
be38-424d-b7e7-3e0ec91d9301
```

```
Enter the Primary Policy Server address/domain name for registering
clients: master.organization.com
```

```
Enter the Primary Policy Server REST TCP/IP port [24351]: Enter the
Registration Client Profile name [default]:
```

If you wish to enable automatic configuration using Client Registration you will require information, including the REST Application ID, REST Application Key, the network name or IP address of the Primary Policy Server that has been configured to enable Client Registration. You will also require the REST services port and the name of the Client Registration Profile which has been configured by the administrator. Once you have these data and have inputted them into the pbinstall prompts, the configuration of the client will be downloaded and the installation will continue. All defaults during the rest of the installation process will now be defaulted from the information retrieved. For more information see "Client Registration Configuration"

- A prompt will ask if you want to install the Registry Name Services.

```
The Registry Name Service of PowerBroker for Unix & Linux facilitates location of
other services within the PBUL enterprise with the aid of a centralized data
repository.
```

```
IMPORTANT: It is highly recommended to utilize Client Registration if you are using
Registry Name Services. Do you wish to utilize Registry Name Service? [yes]?
```

If you answered "no" to the previous question, "Is this the first installation?", you will be asked to register the host as a Registry Name Service client.

To enable the use of Registry Name Services each client needs to be registered with the Primary Server

Please complete the questions below to enable this registration.

Enter the Application ID generated on the Primary Registry Name Server: appid

Enter the Application Key generated on the Primary Registry Name Server: appkey

Enter the address/domain name for the Primary Registry Name Server: host

Enter the Primary Registry Name Server REST TCP/IP port [24351]:

If RNS is specified, the defaults for submitmasters, acceptmasters, logservers, etc, are changed to "*", and "registrynameserver yes" is added to the prospective pb.settings.

9. The pbinstall menu displays a set of options similar to the following:

Opt	Description	[Value]
1	Install Everything Here (Demo Mode)?	[yes]
2	Install License Server?	[yes]
3	Install Registry Name Services Server?	[no]
4	Install Client Registration Server?	[yes]
5	Install PowerBroker Policy Server Host?	[yes]
6	Install PowerBroker Run Host?	[yes]
7	Install PowerBroker Submit Host?	[yes]
8	Install PowerBroker PBSSH	[yes]
9	Install PBSUDO Policy Server?	[yes]
10	Install PowerBroker Log Host?	[yes]
11	Enable Logfile Tracking and Archiving?	[yes]
12	Is this a Log Archiver Storage Server?	[yes]
13	Is this a Log Archiver Database Server?	[yes]
14	Install File Integrity Monitoring Polic...	[yes]
15	Install PowerBroker REST Services?	[yes]
16	List of PowerBroker License Servers	[hoki.unix.symark.com]
17	Central License	[]
18	Enable License History?	[no]

Note: For PowerBroker Express, there will not be an option for a Run Host. For PowerBroker Express, the submit host option installs pbsssh using the express component. For the other PowerBroker for Unix & Linux products, this option installs pbrun.

The following instructions select the required options to do a basic installation only.

For instructions on performing an advanced installation, see [Advanced Installation Instructions Using pbinstall](#).

10. Choose your options.

11. Use the c navigation command to continue the installation.

12. A prompt asks if the settings are correct. Specify "y".
13. A prompt asks if you want to view the install script. Specify n.
Caution! This option is intended for troubleshooting by BeyondTrust Technical Support. The generated install script contains thousands of lines of code.
14. A prompt asks if you want to install PowerBroker for Unix & Linux now. Specify "y".
 The PowerBroker for Unix & Linux install script now executes and installs PowerBroker for Unix & Linux components on this machine.
15. If a PowerBroker for Unix & Linux policy file exists, it is not modified. Starting with version 8.0, if you do not have a policy file, a default policy will be installed by default. The files '{prefix}pbul_policy.conf{suffix}' and '{prefix}pbul_functions.conf{suffix}' will be created in the default directory /opt/pbul/policies from v9.4.3+ and /etc prior to v9.4.3. '{prefix}pbul_policy.conf{suffix}' will then be included in the main policy (by default /opt/pbul/policies/{prefix}pb.conf {suffix} from v9.4.3+ and /etc/{prefix}pb.conf {suffix} prior to v9.4.3).
Caution! An empty policy file rejects all PowerBroker for Unix & Linux commands. For information about writing policy files, see the PowerBroker for Unix & Linux Policy Language Guide.
16. Change the permissions on the policy file so that it can be read by root only:

```
chmod 600 /opt/pbul/policies/pb.conf
```

 The installation is now complete.

PowerBroker for Unix & Linux pbinstall Installation Menu

The PowerBroker for Unix & Linux installation menu (also used by PowerBroker Express) that is provided by the pbinstall script is a comprehensive list of the menu options and default prompts. The items that are displayed vary depending on your system, options selected, and any settings that are found from a current or removed PowerBroker for Unix & Linux installation. The values that are used here are for demonstration purposes and are not necessarily the defaults or recommended values for a given installation.

If you are installing PowerBroker Express, some of these options are not available. This information is noted in the explanation for the appropriate options in [Using the Menu Options](#). For an example of the PowerBroker Express menu options, see [Installation Example for PowerBroker Express](#).

The following list shows all the menu options. However, the menu option numbers that you see might differ from this list, depending on your PowerBroker for Unix & Linux flavor.

BeyondTrust PowerBroker for Unix & Linux Installation Menu

Opt	Description	[Value]
1	Install Everything Here (Demo Mode)?	[yes]
2	Enter existing 'pb.settings' path	[none]
3	Enter directory path for settings file ...	[/opt/acpkg/powerbroker/v9....]
4	Install License Server?	[yes]
5	Install Registry Name Services Server?	[yes]
6	Install Client Registration Server?	[yes]
7	Install PowerBroker Policy Server Host?	[yes]
8	Install PowerBroker Run Host?	[yes]

```

9      Install PowerBroker Submit Host?          [yes]
10     Install PowerBroker PBSSH                 [yes]
11     Install PBSUDO Policy Server?            [yes]
12     Install PowerBroker Log Host?           [yes]
13     Enable Logfile Tracking and Archiving?  [yes]
14     Is this a Log Archiver Storage Server?  [yes]
15     Is this a Log Archiver Database
      Server?                                  [yes]
16     Install File Integrity Monitoring
      Polic...                               [yes]
17     Install PowerBroker REST Services?      [yes]
18     List of PowerBroker License Servers     [*]
19     Central License                         []
20     Enable License History?                 [no]
21     Path to PowerBroker Password Safe
      'pkru...                               []
25     Install PowerBroker Synchronization?    [yes]
26     Install PowerBroker GUI Host?          [yes]
27     Install PowerBroker Secure GUI Host?   [no]
28     Install PowerBroker Utilities: pbvi,
      pb...                                   [yes]
29     Install PowerBroker pbksh?             [yes]
30     Install PowerBroker pbsh?              [yes]
31     Install PowerBroker man pages?         [yes]
32     Will this host use a Log Host?         [yes]
33     PowerBroker Identity Services
      Integration?                           [no]
38     Integration with BeyondInsight?        [no]
53     Registry Name Service database path?   [/opt/pbul/dbs/pbsvc.db]
54     Client Registry database path?        [/opt/pbul/dbs/pbregclnt.db]
55     sudo policy database file path and
      file...                                 [/opt/pbul/dbs/pbsudo.db]
56     Directory location for sudo policy
      files?                                  [/opt/pbul/sudoersdir]
57     Configure pbsudo IO log?               []
58     PowerBroker synchronization can be
      init...                                 [yes]
59     PowerBroker daemon location            [/usr/sbin]
60     Number of reserved spaces for submit
      pr...                                   [80]
61     Administration programs location      [/usr/sbin]
62     User programs location                [/usr/local/bin]

```

```

63     GUI library directory                [/usr/local/lib/pbbuilder]
64     Policy include (sub) file directory  [/opt/pbul/policies]
65     Policy file name                    [/opt/pbul/policies/pb.conf]
66     User man page location              [/usr/local/man/man1]
67     Admin man page location             [/usr/local/man/man8]
68     Log Archive Storage Server name     [*]
69     Log Archive destination directory?   [/var/log/pblogarchive]
70     Log Archiver Database Server name   [*]
71     Log Tracking Database file path and  [/opt/pbul/dbs/pblogarchive.db]
       fil...
72     Logfile Name Cache Database file path? [/opt/pbul/dbs/pblogcache.db]
73     REST Service installation directory? [/usr/lib/beyondtrust/pb/rest]
74     Install PowerBroker REST API sample  [no]
       code?
76     Pblighttpd user                    [pblight]
80     File Integrity Monitor db path?     [/opt/pbul/dbs/pbfim.db]
81     Configure target system's SuperDaemon? [yes]
82     Command line options for pbmasterd  [-ar]
83     Policy Server Delay                  [500]
84     Policy Server Protocol Timeout      [-1]
85     pbmasterd diagnostic log            [/var/log/pbmasterd.log]
86     Eventlog filename                   [/var/log/pb.eventlog]
87     Configure eventlog rotation via size? []
88     Configure eventlog rotation path?   []
89     Configure eventlog rotation via cron? [no]
90     Validate Submit Host Connections?   [no]
91     List of PowerBroker Policy Servers to  [*]
       s...
92     pbrun diagnostic log?               [none]
93     pbssh diagnostic log?               [none]
94     Allow Local Mode?                   [yes]
95     Additional secured task checks?     [no]
96     Suppress Policy Server host failover  [yes]
       er...
97     List of PowerBroker Policy Servers to  [*]
       a...
98     pblocald diagnostic log             [/var/log/pblocald.log]
99     Command line options for pblocald    []
100    Syslog pblocald sessions?           [no]
101    Record PTY sessions in utmp/utmpx?   [yes]
102    Validate Policy Server Host          [no]
       Connections?

```

103	List of PowerBroker Log Hosts	[*]
104	Command line options for pblogd	[]
105	Log Host Delay	[500]
106	Log Host Protocol Timeout	[-1]
107	pblogd diagnostic log	[/var/log/pblogd.log]
108	List of PowerBroker log reserved filesy...	[none]
109	Number of free blocks per log system fi...	[0]
110	Command line options for pbsyncd	[]
111	Sync Protocol Timeout	[-1]
112	pbsyncd diagnostic log	[/var/log/pbsyncd.log]
113	pbsync diagnostic log	[/var/log/pbsync.log]
114	pbsync sychronization time interval (in...	[15]
115	Add installed shells to /etc/shells	[no]
116	PowerBroker pbksh diagnostic file	[/var/log/pbksh.log]
117	PowerBroker pbsh diagnostic file	[/var/log/pbsh.log]
118	Stand-alone pblogd command	[none]
119	Stand-alone root shell default iolog	[/pbshell.iolog]
120	Command line options for pbguid	[]
122	pbguid diagnostic log	[/var/log/pbguid.log]
123	pbguid site configuration file	[none]
124	Use syslog?	[yes]
125	Syslog facility to use?	[LOG_AUTHPRIV]
126	Base Daemon port number	[24345]
127	pbmasterd port number	[24345]
128	pblocald port number	[24346]
129	pblogd port number	[24347]
130	pbguid port number	[24348]
132	pbsyncd port number	[24350]
133	REST Service port number	[24351]
134	Add entries to '/etc/services'	[yes]
135	Allow non-reserved port connections	[yes]
136	Inbound Port range	[1025-65535]
137	Outbound Port range	[1025-65535]
140	PowerBroker network encryption options	[aes-256:keyfile=/etc/pb.rest.key]
141	PowerBroker event log encryption options	[none]
142	PowerBroker I/O log encryption options	[none]

```

143 PowerBroker report encryption options [none]
144 PowerBroker policy file encryption [none]
    options
145 PowerBroker settings file encryption [none]
    type
146 PowerBroker REST API encryption [aes-
    options 256:keyfile=/etc/pb.rest.key]
147 Configure with Kerberos v5? [no]
153 Enforce High Security Encryption? [yes]
154 Use SSL? [yes]
155 SSL Configuration? [requiresssl]
156 SSL pbrun Certificate Authority [none]
    Directory?
157 SSL pbrun Certificate Authority File? [none]
158 SSL pbrun Cipher List? [HIGH:!SSLv2:!3DES:!MD5:@ST...]
159 SSL pbrun Certificate Directory? [none]
160 SSL pbrun Certificate File? [none]
161 SSL pbrun Private Key Directory? [none]
162 SSL pbrun Private Key File? [none]
163 SSL pbrun Certificate Subject Checks? [none]
164 SSL Server Certificate Authority [none]
    Direct...
165 SSL Server Certificate Authority File? [none]
166 SSL Server Cipher List? [HIGH:!SSLv2:!3DES:!MD5:@ST...]
167 SSL Server Certificate Directory? [none]
168 SSL Server Certificate File? [/etc/pbssl.pem]
169 SSL Server Private Key Directory? [none]
    [none]
170 SSL Server Private Key File? [/etc/pbssl.pem]
171 SSL Server Certificate Subject Checks? [none]
172 PowerBroker SSL Certificate Country [US]
    Code
173 PowerBroker SSL Certificate [AZ]
    State/Province
174 PowerBroker SSL Certificate Location [Phoenix]
    (T...
175 PowerBroker SSL Certificate [Security]
    Organizatio...
176 PowerBroker SSL Certificate [BeyondTrust]
    Organization
177 Configure PowerBroker with LDAP? [no]
178 Install BeyondTrust built-in third- [yes]
    part...

```

179	BeyondTrust built-in third-party librar...	[/usr/lib/beyondtrust/pb]
191	Use PAM?	[no]
199	Allow Remote Jobs?	[yes]
200	PowerBroker UNIX Domain Socket directory	[none]
201	Reject Null Passwords?	[no]
202	Enable TCP keepalives?	[no]
203	Name Resolution Timeout	[0]

Prefix and Suffix Installation Instructions

A prefixed or suffixed installation is performed by specifying the `-p` and/or `-s` arguments to `pbinstall` and `pbuninstall`, respectively (refer to the `pbinstall` and `pbuninstall` sections of this manual). Both options take one argument: the prefix (or suffix) to use.

With a prefix and/or suffix specified, the names of all of the executable programs, services and ports, and default log file names are qualified with that prefix and/or suffix.

Prefixes always are added to the beginning of the name. Suffixes, with the exception of the daemon error logs and man page file names, are added to the end of the name. Daemon error logs are named (for example) `{prefix}pbmasterd{suffix}.log`.

Note: You cannot use a prefixed or suffixed installation with PowerBroker package installations.

If PowerBroker for Unix & Linux is installed with a prefix and/or suffix, execute `pbuninstall` using the same prefix and/or suffix. Failure to correctly specify the prefix and/or suffix to `pbuninstall` results in either `pbuninstall` failing or the uninstall of the incorrect copy of PowerBroker for Unix & Linux.

Note: The `pb.cfg` file is also prefixed and/or suffixed when it is created.

Running Prefixed and Suffixed Installations

To run a prefix installation, type:

```
./pbinstall -p prefix
```

where `prefix` is the prefix you are using.

To run a suffix installation, type:

```
./pbinstall -s suffix
```

where `suffix` is the suffix you are using.

To run a prefix and suffix installation, type:

```
./pbinstall -p prefix -s suffix
```

where `prefix` is the prefix and `suffix` is the suffix you are using.

Note: You cannot use a prefixed or suffixed installation with PowerBroker package installers.

Advanced Installation Instructions Using pbinstall

This section provides step-by-step instructions for using all the installation options that are available using the pbinstall script. These options are discussed in the order that they are used in the PowerBroker for Unix & Linux installation menu. These steps are optional and should be selected after reviewing [Installation Considerations](#) and [Installation Preparation](#).

In addition, some options do not appear unless certain combinations of options are selected. For more information, see [Example of a pbinstall Execution](#).

Starting pbinstall

If you downloaded PowerBroker for Unix & Linux using the Web or FTP, do the following. (To install PowerBroker for Unix & Linux from a CD, see the instructions in [Step-by-Step Instructions for a Basic Installation Using pbinstall](#)).

1. Extract the tarball files into /opt/beyondtrust by executing the following command:

```
gunzip -c pb<flavor_version>.tar.Z | tar xvf -
```
2. Navigate to the installation directory:

```
cd /opt/beyondtrust/powerbroker/<version>/<flavor>/install
```
3. Execute the installation script by typing:

```
./pbinstall
```

Note: If you are using a prefix and/or suffix, see the information in [Prefix and Suffix Installations](#).

4. Press Enter after reading the initial messages.

Using the Menu Options

Note: Depending on your operating system and other factors, the step numbers that are listed here may or may not match the menu option numbers that you see on the screen, and some items might not be available. In these steps, “choose this option” means to type the number that corresponds to the option on the screen and press Enter.

1. Install Everything Here (Demo Mode)?
 Choose this option and specify "y" to install the Policy Server host, run host, submit host, and log host on this computer. This option is useful for testing or demonstrating PowerBroker for Unix & Linux on a single computer in your environment.
2. Enter existing 'pb.settings' path
 For package installations only. Choose this option and specify the path to an existing pb.settings file to use for the package installation.
3. Enter directory path for settings file creation
 For package installations only. Choose this option and specify a directory in which to save the generated PowerBroker for Unix & Linux settings, configuration, and key files. The default directory is:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/install/ settings_files
```
4. Install License Server?
 Specify "y" to install a License Server which provides product license management for PowerBroker for Unix & Linux.
5. Install Registry Name Services Server?
 Specify "y" to install the Registry Name Service which will provide the product with a method of addressing and locating other parts of PowerBroker for Unix & Linux.

6. Install Client Registration Server?
Specify "y" to install the Client Registration Server which provides a repository for customized install profiles. If you already chose to install the Registry Name Service, installing Client Registration Server is mandatory.
7. Install PowerBroker Policy Server Host?
Choose this option and specify "y" to install the Policy Server host component on this host.
8. Install PowerBroker Run Host?
Choose this option and specify "y" to install the run host component on this host.
Note: This option is not available with PowerBroker Express.
9. Install PowerBroker Submit Host?
Choose this option and specify "y" to install the submit host component on this host.
Note: For PowerBroker Express, this option installs pbssh. For the other PowerBroker for Unix & Linux products, this option installs pbrun.
10. Install PBSSH
This item is available only when you specify "y" for the previous item. Using PowerBroker for Unix & Linux policy and the pbssh program, you can control access to, and activities on, SSH-managed devices. The pbssh program uses the SSH protocol (or, optionally, the telnet protocol) to connect to devices that do not have PowerBroker for Unix & Linux installed on them; such devices can include Windows computers and certain network devices.
Choose this option and specify "y" to install the pssh program.
Note: This option is not used with PowerBroker Express because pbssh is installed with the previous option for the PowerBroker submit host.
11. Install PBSUDO Policy Server?
Specify "y" to configure the server to be able to store and process sudo policies.
12. Install PowerBroker Log Host?
Choose this option and specify "y" to install the log host component on this host.
13. Enable Logfile Tracking and Archiving?
If the installation detects that the user is installing the Policy Server Host or the Log Host on the current machine, it will display in the menu the install question "Enable Logfile Tracking and Archiving?" and set it to "yes" by default. When the answer to this question is set to "yes", the installer would then prompt the user for the Log Archive Storage Server name and the Log Archiver Database Server name. Log Tracking and Archiving requires REST services to be installed.
Note: REST Services are not fully supported on Mac OS X.
14. Is this a Log Archiver Storage Server?
If the current machine is the intended Log Archive Database Server, it must have the REST service pre-installed on it. It is also required to have the logarchivedb setting in pb.settings, which specifies where the SQLite database that stores the location of logfiles, as well as the archiving information is located.
If the answer to this question is set to "yes", the install will display the following question:
Configure this host to be a Log Archive Storage Server which receives logfiles to archive and stores them in the appropriate path:

Yes This host will be configured as a Log Archiver Storage Server
No This host will NOT be configured as a LogArchiver Storage Server
Set as a Log Archiver Storage Server? [no]? yes
The Log Archive Storage Server which will accept and place archived logfiles in a designated pathname.

Ensure that it is located in filesystem with ample free space to accommodate incoming logfiles.

Enter the default directory path for archived logfiles []: /pbul/logs

It will also set the "Log Archive Storage Server name" to the hostname of the current machine.

15. Is this a Log Archiver Database Server?

If the current machine is the intended Log Archive Database Server, it must have the REST service pre-installed on it. It is also required to have the logarchivedb setting in pb.settings, which specifies where the SQLite database that stores the location of logfiles, as well as the archiving information is located. If the answer to this question is set to "yes", the install will display the following question:

Configure this host to be a Log Archive Database Server which creates and maintains the log tracking database:

Yes This host will be configured as a Log Archiver Database Server

No This host will NOT be configured as a LogArchiver Database Server

Set as a Log Archiver Database Server? [no]? yes

PowerBroker will create and maintain a SQLite database to track the location of logfiles. Specify the path and filename of the SQLite logfile tracking database file and ensure that the given database file system has ample space for growth.

Enter the path and filename of PowerBroker's SQLite log tracking database file []: /var/log/pbul90_tracking.db

It will also set the "Log Archive Database Server name" to the hostname of the current machine.

16. Install File Integrity Monitoring Policy Server?

Specify "y" to install and configure the centralized repository for FIM policies.

17. Install PowerBroker REST Services?

This option is automatically enabled to install the PowerBroker RESTful web-based API for product settings, policy configuration, and IO log retrieval. When installing server-side components of PowerBroker for Unix & Linux, installing the REST Services is mandatory.

Note: REST Services are not fully supported on Mac OS X.

18. List of PowerBroker License Servers

Enter a space-separated list of hostnames of License Servers within the PowerBroker for Unix & Linux installation. The Primary License Server is first in the list, followed by Secondary License Servers listed in order of failover.

If Registry Name Service is configured this value should be "*" denoting that the value is held within the Service database.

19. Central License

Enter the license provided by BeyondTrust. Type either the JSON-formatted license text or the full pathname of the file containing the license.

20. Enable License History?

Enter "yes" if every license event should be logged to the Primary Log Server for detailed license information.

21. Path to PowerBroker Password Safe 'pkrun' binary

This item is available only if you choose to install PBSSH. Choose this option to specify where the PowerBroker Password Safe pkrun binary resides. The pbssh command can use PowerBroker Password Safe for the user's password acquisition. To do this, PowerBroker for Unix & Linux needs to know where the PowerBroker Password Safe pkrun binary resides.

Choose this option and do one of the following:

- Specify the absolute path where pkrun resides.
 - Specify none to clear the entry (default).
22. `PowerBroker Password Safe certificate file`
This item is available only when you specify the path instead of the blank default for the previous item. This item is required by `PowerBroker Password Safe pkrun` binary.
Choose this option to specify the `PowerBroker Password Safe` certificate file's path and do one of the following:
- Specify the `PowerBroker Password Safe` certificate file's path (include the certificate files absolute path).
 - Specify none to clear the entry (default).
23. `Primary failover PowerBroker Password Safe appliances`
This item is available only when you specify the path instead of the blank default in step 21. Choose this option to specify `PowerBroker Password Safe` appliances names and do one of the following:
- Enter a list of space-separated `PowerBroker Password Safe` appliance names. (This item is required by the `PowerBroker Password Safe pkrun` binary.)
 - Enter none to clear the entry (default).
24. `Support short names in PowerBroker Password Safe certificate?`
This item is available only when you specify the path instead of the blank default in step 21. Choose this option and select `y` to specify that `pbssh` must use short host names when communicating with `PowerBroker Password safe`, rather than using the fully-qualified domain name.
25. `Install PowerBroker Synchronization?`
Choose this option and specify `"y"` to enable this host to participate in log synchronization.
26. `Install PowerBroker GUI Host?`
Choose this option and specify `"y"` to install the GUI host component on this host.
27. `Install PowerBroker Secure GUI Host?`
Choose this option and specify `"y"` to install the secure GUI host component on this host.
28. `Install PowerBroker Utilities: pbvi, ...`
Choose this option and specify `"y"` to install the `PowerBroker for Unix & Linux` utilities on this host.
Note: This option is not available with `PowerBroker Express`.
29. `Install PowerBroker pbksh?`
Choose this option and specify `"y"` to install the `pbksh` component on this host.
30. `Install PowerBroker pbsh?`
Choose this option and specify `"y"` to install the `pbsh` component on this host.
31. `Install PowerBroker man pages?`
Choose this option and specify `"y"` to install the man pages.
Note: This option installs only those man pages applicable for `PowerBroker Express`.
32. `Will this host use a Log Host?`
Choose this option and specify `"y"` to log the components on this host to a log server.
33. `PowerBroker Identity Services Integration?`
The `pbinstall` program does not detect whether `PowerBroker Identity Services` is installed. For more information about `PowerBroker for Unix & Linux` integration with `PowerBroker Identity Services`, see “`PowerBroker for Unix & Linux and PowerBroker Identity Services`” in the *PowerBroker for Unix & Linux System Administration Guide*.
Choose this option and specify one of the following:

- no to disable PowerBroker for Unix & Linux integration with PowerBroker Identity Services. This is the default.
 - yes to enable PowerBroker for Unix & Linux integration with PowerBroker Identity Services.
34. `Enable failover event logging to PowerBroker Identity Services?`
This option is available only if you choose yes for the `PowerBroker Identity Services Integration?` item. This option enables or prevents information about failover events (for Policy Server hosts and log hosts) to be sent to the PowerBroker Identity Services event log mechanism. It also controls the posting of an event to PowerBroker Identity Services when the Policy Server host fails to connect to the run host.
Choose this option and specify one of the following:
- no to prevent the sending of failover event information to the PowerBroker Identity Services event log mechanism. This is the default.
 - yes to send failover event information to the PowerBroker Identity Services event log mechanism.
35. `Enable successful connection event logging to PowerBroker Identity Services?`
If specifying "yes", PowerBroker clients will report successful connection events to PowerBroker Identity Services.
36. `Enable event logging to PowerBroker Identity Services?`
This option is available only if you choose yes for the `PowerBroker Identity Services Integration?` item. This option enables or prevents information about Accept, Reject, Finish, and keystroke action events to be sent to the PowerBroker Identity Services event log mechanism.
Choose this option and specify one of the following:
- no to prevent the sending of event information to the PowerBroker Identity Services event log mechanism. This is the default.
 - yes to send event information to the PowerBroker Identity Services event log mechanism.
37. `PowerBroker Identity Services shared libraries`
This option is available only if you choose yes for the `PowerBroker Identity Services Integration?` item. This option specifies the shared libraries that are required for the integration.
The default value is system-dependent subdirectories of the `/opt/pbis` directory. If you do not modify the default value, PowerBroker for Unix & Linux follows these rules to locate the PowerBroker Identity Services shared libraries:
- If the libraries are found in the default directories, the default directories are used.
 - If the libraries are not found in the default directories, the `/opt/pbis/lib` directory is searched. If the libraries are not found there, then this option is reset to none.
- If the libraries are not found (or if this option is set to none), then PowerBroker for Unix & Linux displays a warning message during installation and the PowerBroker Identity Services -related settings are set to no or none (as applicable).
To modify this option from the default value, choose the option and specify the full paths and file names for the PowerBroker Identity Services `libeventlog_norpc` and `liblwbase_nothr` libraries, in that order.
38. `Integration with BeyondInsight?`
This option is available for logservers and Policy Server hosts.
This option allows the sending of eventlog records to BeyondInsight and indexing of IO logs.
39. `Send event log records to BeyondInsight?`
This option is only available if you choose 'yes' for the 'Integration with BeyondInsight?' item. Default is 'yes'
This option enables the sending of event log records to BeyondInsight.

40. `BeyondInsight hostname`

This option is only available if you choose 'yes' for the 'Integration with BeyondInsight?' item. Enter the name of the BeyondInsight host. If set to 'none', return to the main menu and revert to 'no' for 'Send events log records to BeyondInsight'.

41. `BeyondInsight Workgroup ID`

Choose this option to provide a value for BeyondInsight Workgroup ID. BeyondInsight will use this Workgroup ID as an identifier for asset matching and grouping for all the eventlog records from this Policy Server/Logserver.

This identifier should be a maximum of 128 characters and composed of only letters, numbers, and spaces.

42. `BeyondInsight SSL port number`

This option is only available if you choose 'yes' for the 'Integration with BeyondInsight?' item. Enter the name of the BeyondInsight SSL port.

43. `BeyondInsight SSL Client Certificate`

This option is only available if you choose 'yes' for the 'Integration with BeyondInsight?' item. Enter the SSL Client Certificate filename and location. This cert must be exported from BeyondInsight. If set to 'none', return to the main menu and revert to 'no' for 'Send events log records to BeyondInsight'.

44. `BeyondInsight SSL CA file`

This option is only available if you choose 'yes' for the 'Integration with BeyondInsight?' item. Enter the SSL Certificate Authority filename and location. This CA must be exported from BeyondInsight. If set to 'none', return to the main menu and revert to 'no' for 'Send events log records to BeyondInsight'.

45. `BeyondInsight Event Store file`

This option is only available if you choose 'yes' for the 'Integration with BeyondInsight?' item. Sets the path/file name which will store events that were unsuccessfully forwarded to BeyondInsight. The default directory location is `"/var/log"` (platform dependent, also `/usr/log`, `/usr/adm`, `/var/adm`) and the default filename format is `<prefix>pb.rcs_eventstore<suffix>`. The user may change this pathname.

46. `IOLog index Store file`

This option is only available if you choose 'yes' for the 'Integration with BeyondInsight?' item. Sets the path/file name which will IOLog events that were unsuccessfully forwarded to BeyondInsight. The default directory location is `"/var/log"` (platform dependent, also `/usr/log`, `/usr/adm`, `/var/adm`) and the default filename format is `<prefix>pb.iolog.store<suffix>`. The user may change this pathname.

47. `Index IO Logs using Solr?`

This option is only available if you choose 'yes' for the 'Integration with BeyondInsight?' item. Default is 'no'. This option enables the indexing of IO Logs using Solr.

48. `Solr hostname`

This option is only available if you choose 'yes' for the 'Index IO Logs using Solr?' item. Host where the Solr Server is installed. If set to 'none', returns to the main menu and reverts to 'no' for 'Index IO logs using Solr'.

49. `Solr port number`

This option is only available if you choose 'yes' for the 'Index IO Logs using Solr?' item. Port number for Solr server. Default is 8983.

50. `Solr SSL CA file`

This option is only available if you choose 'yes' for the 'Index IO Logs using Solr?' item. Enter the Solr SSL CA filename and location. If set to 'none', return to the main menu and revert to 'no' for 'Solr SSL Client key file' and 'Solr SSL Client Certificate File'.

51. Solr SSL Client key file

This option is only available if you choose 'yes' for the 'Index IO Logs using Solr?' item. Enter the Solr SSL Client key filename and location. If set to 'none', return to the main menu and revert to 'no' for Solr SSL CA file ' and 'Solr SSL Client Certificate File'.

52. Solr SSL Client Certificate file

This option is only available if you choose 'yes' for the 'Index IO Logs using Solr?' item. Enter the Solr SSL Client Certificate filename and location. If set to 'none', return to the main menu and revert to 'no' for Solr SSL CA file ' and Solr SSL CA file '.

53. Registry Name Service database path?

Specify the path of the Registry Name Service database file.

54. Client Registry database path

Specify the path for the Client Registry profile database file.

55. sudo policy database file path and filename

When the answer to "Integrate PowerBroker with sudo" is set to yes, this menu option will be displayed. PowerBroker for Unix & Linux will create and maintain a SQLite database to track and store sudo policy files. Provide a secure path with ample space for growth and the name of the database where Sudo policies will be stored. The path and file name will be created the first time a sudoers policy is added to the database from a Sudo client. There is no default value for this option.

56. Directory location for sudo policy files?

Specify a secure directory where PowerBroker for Unix & Linux will store the sudo policy files centrally on PowerBroker for Unix & Linux policy server host. If this directory doesn't exist, it will be created. There is no default value for this option.

57. Configure pbsudo IO log?

The PowerBroker for Unix & Linux Log Server can process sudo I/O via the PowerBroker sudo plugin Configure the I/O log path and filename for sudo I/O logs. Specify a path and filename to store these I/O log files. This path can include PowerBroker for Unix & Linux variables surrounded by '%'. Refer to the description of 'pbsudo_iolog' in the *PowerBroker for Unix & Linux Administration Guide* for a list of valid variables.

Example: /var/log/pbsudo/iologs/%host%/pbsudo.io.XXXXXX

The directory specified will be created during the installation if it doesn't exist. There is no default value for this option.

58. PowerBroker synchronization can be initiated from this host?

Choose this option and specify "y" to install pbsync to enable this host to start log synchronization.

59. PowerBroker daemon location

Choose this option and specify a location for it. It is recommended that you use the default location, but you can choose to specify a different location. However, do not use system directories for this purpose.

60. Number of reserved spaces for submit pr...[80]

Available in version 8.0 and later, and only on Linux, AIX and Mac OSX platforms, this feature will modify the pbmasterd, pblocald and pblogd command line arguments (viewable via ps) to include information about the originating pbrun request. This will allow administrators to determine which pbrun/pbmasterd/pblocald/pblogd processes are related to a given request.

Choose this option and specify the number of space to reserve in the process list of pbmasterd, pblocald and pblogd processes by adding a "-i" to the daemon startup files. This new command line option will be used to reserve space in the process list so that the command line argument space can be updated with information about the originating request (submituser, submithost, runcommand, and the pbrun pid).

61. `Administration programs location`
Choose this option and specify a location for them. It is recommended that you use the default location, but you can choose to specify a different location. However, do not use system directories for this purpose.
62. `User programs location`
Choose this option and specify a location for them. It is recommended that you use the default location, but you may choose to specify a different location. However, do not use system directories for this purpose.
63. `GUI library directory`
Choose this option and specify a location for it. This option creates a directory under `/usr/local/lib` to contain the help files for the PowerBroker for Unix & Linux Browser Interface, as well as the sample policy files. It is recommended that you use the default location, but you can specify a different location. However, do not use system directories for this purpose.
64. `Policy include (sub) file directory`
Choose this option and specify a directory for the policy files. It is recommended that you use the default location, but you can specify a different location. However, do not use system directories for this purpose.
65. `Policy file name`
Enter the PowerBroker for Unix & Linux policy file name.
66. `User man page location`
Choose this option and specify a location for them. This option enables you to specify where the user man pages are located. It is recommended that you use the default location, but you may specify a different location. However, do not use system directories for this purpose.
67. `Admin man page location`
Choose this option and specify a location for them. This option enables you to specify where the admin man pages are located. It is recommended that you use the default location, but you can specify a different location. However, do not use system directories for this purpose.
68. `Log Archive Storage Server name`
The Log Archive Storage Server is the destination host where the logfiles are archived. It is required to have the PBUL REST service pre-installed on that machine. There is no default value for this field, but the user is not allowed to proceed without specifying the appropriate server name. The value is saved in the `logarchivehost` PowerBroker for Unix & Linux setting.
69. `Log Archive destination directory?`
This is the destination directory on the Log Archive Storage Server where archived logfiles are stored in a designated pathname. Ensure that it is located in filesystem with ample free space to accommodate incoming logfiles.
70. `Log Archiver Database Server name`
The Log Archive Database Server is the destination host where the logfile tracking database resides. It is required to have the PowerBroker for Unix & Linux REST service pre-installed on that machine. There is no default value for this field, but the user is not allowed to proceed without specifying the appropriate server name. The value is saved in the `logarchivedbhost` PowerBroker for Unix & Linux setting.
71. `Log Tracking Database file pathname?`
This is the path and filename of the PowerBroker will create and maintain the SQLite database to track the location of logfiles. Specify the pathname of the SQLite logfile tracking database file and ensure that the given database file system has ample space for growth.
72. `Logfile Name Cache Database file path?`
Enter the path of the database file to cache the location of event and IO logfiles. It will be used when integrating PowerBroker Server Management Console with PowerBroker for Unix & Linux. Enter "none" to disable the feature.

73. `REST Service installation directory?`
 This menu item is enabled only if REST Services are to be installed.
Note: REST services are not fully supported on Mac OS X.
74. `Install PowerBroker REST API sample code?`
 This menu item is enabled only if REST Services are to be installed.
Note: REST services are not fully supported on Mac OS X.
75. `REST API sample code directory?`
 This menu item is enabled only if REST Services are to be installed.
Note: REST services are not fully supported on Mac OS X.
76. `Pblighttpd user`
 The username used to run the REST services as. This user is created if necessary.
 This menu item is enabled only if REST Services are to be installed.
Note: REST services are not fully supported on Mac OS X.
77. `Create Pblighttpd user?`
 If the `pblighttpd` user specified in the previous step does not exist, answering “yes” to this question will create it.
78. `Pblighttpd user UID`
 Specify the user id for the new `pblighttpd` user.
79. `Pblighttpd user GID`
 Specify the group id for the new `pblighttpd` user.
80. `File Integrity Monitor db path?`
 Enter the path of PowerBroker's File Integrity Monitor Policy database file.
81. `Configure target system's SuperDaemon?`
 Choose this option and specify "y" if you want to configure the file.
 PowerBroker for Unix & Linux can be configured into the `inetd`, `xinetd`, `launchd`, or `SMF` superdaemons, which are OS-dependent. These superdaemons are used by PowerBroker for Unix & Linux to listen on a TCP/IP port for inbound connections requesting PowerBroker for Unix & Linux daemon services.
 When the superdaemon detects a connection request, it forks a copy of the PowerBroker for Unix & Linux daemon to serve the request. For more information about superdaemons, consult the documentation for your operating system.
Note: If no is specified, any existing PowerBroker for Unix & Linux installation that is configured with the specified prefix and/or suffix is removed from the superdaemon configuration.

 Starting with version 7.1.0, if the system PowerBroker for Unix & Linux is being installed on is IPv6-capable and configuration of `inetd`, `xinetd`, `SMF` (Solaris), is being performed, the super daemon configuration will be set for IPv6 rather than IPv4.
82. `Command line options for pbmasterd`
 Choose this option and specify the command line options that you want. Available syntax and command line options for `pbmasterd` are:
 Syntax: `[-arsV] [-e logfile]`

 `[--disable_optimized_runmode]`
 -a Send the job acceptance messages to syslog.
 -e Use the log file as the `pbmasterd` diagnostic log file. The `-e` command line option will override the syslog setting in the `pb.settings` file. You must specify the file name if you use the `-e` option.
 -r Send the job rejection messages to syslog.

-s Send the error messages to syslog. The -s command line option will override the syslog setting in the pb.settings file, if you want to change it in the future.

-V Print the version number mismatch messages.

none Erase all options

--disable_optimized_runmode

Suppresses optimized run mode for any tasks that are authorized by this Policy Server host.

Note: The installation is currently set to use the syslog in the PowerBroker for Unix & Linux pb.settings file. This setting is the default.

83. Policy Server Delay

Choose this option and specify the length of time (in milliseconds) that a pbrun command should wait for an initial connection to a Policy Server host. If a connection does not occur within a specified number of milliseconds, then the command uses another host that is specified in the pb.settings file for submitmasters.

84. Policy Server Protocol Timeout

Choose this option and specify the length of time the daemon should wait for a response from a Policy Server host or the time a Policy Server host should wait for a response from another PowerBroker for Unix & Linux program.

85. pbmasterd diagnostic log

Choose this option and specify a location for it. This option enables you to specify where the pbmasterd diagnostic log is located.

86. Eventlog filename

Choose this option and specify a location for it. This option enables you to specify where the event log file is located.

87. Configure eventlog rotation via size

Choose this option and specify a size for event log rotation.

88. Configure eventlog rotation path

Choose this option and specify a path where the event log will be moved to.

89. Configure eventlog rotation via cron

Choose this option add a cron job to rotate the eventlog, and specify the cron minute, hour, days-of-the-month, month, and days-of-the-week fields.

90. Validate Submit Host Connections?

Choose this option and specify one of the following settings. The PowerBroker for Unix & Linux Policy Server daemon (pbmasterd) can use name resolution to validate the host name and IP address of the submit host connection to a Policy Server host.

- Specify y to validate submit host connections. If you decide to use this facility, then you must do the following:
 - a. Ensure that name resolution works correctly on all machines.
 - b. Ensure all Policy Server hosts and submit hosts are upgraded to PowerBroker for Unix & Linux V3.5.7 or higher before enabling this feature.
 - c. Ensure that each submit host connection's host name and IP address match those that are listed in the Policy Server host's name resolution services.
- Specify n to disable this checking. This setting is the default value.

91. List of PowerBroker Policy Server Hosts to submit to

Choose this option and do the following:

- a. If submitmasters already has a value, specify "y" at the prompt.
Do you wish to make changes to this list?
 - b. At the prompt, specify a host name, or a list of space-delimited host names, to serve as Policy Servers to submit secured tasks to (a fully-qualified domain name may be required):
Enter Policy Server list (submitmasters):

The host names should now appear in the List of PowerBroker Policy Server Hosts to submit to line of the pbininstall menu.
92. `pbrun diagnostic log?`
Choose this option and specify a location for it. This option is typically used only when requested by BeyondTrust Technical Support.
Note: This option is not available with PowerBroker Express.
93. `pbssh diagnostic log?`
BeyondTrust PowerBroker for Unix & Linux' pbssh program can maintain a separate, individual host diagnostic log file. This log file is typically only used when requested by BeyondTrust Technical Support.
Specify a full path specification for the pbssh diagnostic log file or none for none.
94. `Allow Local Mode?`
Choose this option and specify y to allow Local Mode. This option allows the requested secured task to replace the executing copy of pbrun. Local Mode executes secured tasks on the submit host only.
Note: This option is not available with PowerBroker Express.
95. `Additional secured task checks?`
Choose this option and specify whether to enable additional secured task checks.
This option determines whether the run host or submit host performs an additional check on the security of the requested command. This check helps to ensure that the command cannot be compromised by a user other than root or the user running the PowerBroker for Unix & Linux command (for example, sys, oracle).
This setting is used on run hosts or submit hosts using Local Mode. The policy language variable `runsecurecommand` can be set by the configuration policy on the Policy Server host for the same effect.
- Specify "y" to check the `runcommand` and all directories above it to see if anyone other than root or the `runuser` has write permission. If the command file or any of the directories above it are writable by anyone other than root or the `runuser`, then the run host refuses to run the command.
 - Specify "n" to disable this feature.
96. `Suppress Policy Server host failover error messages?`
When a connection to Policy Server host fails, PowerBroker for Unix & Linux will failover to another available Policy Server host (if configured), and generate an error message regarding the event.
Choose this option and do one of the following:
- Specify n to enable the Policy Server host failover error messages (default).
 - Specify y to suppress the Policy Server host failover error messages.
97. `List of PowerBroker Policy Servers to accept from`
Choose this option and then do the following:
- a. If acceptmasters already has a value, specify "y" at the prompt.
Do you wish to make changes to this list?
 - b. At the prompt, specify a host name, or a list of space-delimited host names, to serve as Policy Servers to accept secured tasks from (a fully-qualified domain name may be required):
Enter Incoming Policy Server list (acceptmasters):

The accept Policy Server host name should now display in the List of PowerBroker Policy Server Hosts to accept from ... line of the pbinstall menu.

98. pblockd diagnostic log

Choose this option and specify a directory and file name for it.

Note: This option is not available with PowerBroker Express.

99. Command line options for pblockd

Choose this option and specify the command line options that you want. Available syntax and command line options for pblockd are:

```
[-sV] [-e logfile] [-m master_host]
```

-s Send error messages to syslog. The -s command line option overrides the syslog setting in the pb.settings file if you decide to change it in the future.

-e Use logfile as the pblockd diagnostic log file. The -e command line option overrides the settings file.

-m Accept pbmasterd connections from master_host only. Multiple -m options can be used to specify more than one host.

-V Print version number mismatch messages.

none Erase all options.

The installation is currently set to use the syslog in the PowerBroker for Unix & Linux pb.settings file. This setting is the default.

Note: This menu option is not available with PowerBroker Express.

100. Syslog pblockd sessions

Choose this option and specify y to log pblockd accepted and rejected requests to syslog.

Note: This option is not available with PowerBroker Express.

101. Record PTY sessions in utmp/utmpx?

Choose this option and specify y to record PowerBroker for Unix & Linux terminal sessions in the utmp (or utmpx) file.

102. Validate Policy Server Host Connections?

Choose this option and specify one of the following settings.

The PowerBroker for Unix & Linux local daemon (pblockd) can use name resolution to validate the host name and IP address of the Policy Server host connection to a run host.

- Specify y to validate Policy Server host connections. This validation requires that each Policy Server connection's host name and internet address match those that are retrieved from name resolution services.

Note: If you decide to use this facility, then you must ensure that name resolution works correctly on all machines before enabling this feature. You must also ensure that all Policy Server hosts and run hosts are upgraded to PowerBroker for Unix & Linux V3.5.7 or later before enabling this feature.

- Specify n to disable this checking. This setting is the default value.

103. List of PowerBroker Log Hosts

Choose this option and specify which machines are to be log hosts.

PowerBroker for Unix & Linux needs to know which machines you have selected as log hosts. Log hosts are the hosts that Policy Server hosts select to perform event and I/O logging. To accomplish this task, pbmasterd looks at the setting for logservers in the pb.settings file. This logservers setting contains the names of the log host machines or a netgroup.

You can add, modify, or remove machine names by doing the following:

- a. If logservers already has a value, specify "y" at the prompt.
Do you wish to make changes to this list?
- b. At the prompt, specify a host name, or a list of space-delimited host names, to serve as Log Hosts:
Enter Log Server list (logservers):

The log host names should now appear in the List of PowerBroker Log Hosts line of the pbinstall menu.

104. Command line options for pblogd

Choose this option and specify the command line options that you want.

The available syntax and command line options for pblogd are:

```
[-ars] [-e logfile]
```

-a Record accept events on syslog.

-e Use logfile as the pblogd diagnostic log file. If you previously specified the pblogd log file as /var/log/pblogd.log, the -e command line option overrides the pblogd setting in the pb.settings file.

-r Record reject events on syslog.

-s Send error messages to syslog. If you have previously specified to use the syslog setting in the pb.settings file, the -s command line option overrides the settings file if you decide to change it in the future.

none Erase all options.

105. Log Host Delay

Choose this option and specify the length of time (in milliseconds) that a daemon should wait for an initial connection to a log host. If a connection does not occur within a specified number of milliseconds, then it tries another server that is specified in the logservers setting in the pb.settings file.

106. Log Host Protocol Timeout

Choose this option and specify the length of time a daemon should wait for a response from a log host or the time a log host should wait for a response from another PowerBroker for Unix & Linux program. Enter the value of the log host protocol timeout (-1 to 1200000). 0 or -1 disables this timeout. -1 is the default.

107. pblogd diagnostic log

Choose this option and specify a location for it. This option enables you to specify the directory and file name for the pblogd diagnostic log. Enter none for no error reporting.

108. List of PowerBroker log reserved file systems

Choose this option to specify reserved file systems. PowerBroker for Unix & Linux allows the log host to control the file system space and enables the immediate failover to the next log host.

- Enter none to specify no reserved file systems.
- To specify reserved file systems, type the names of the reserved file systems that you want to failover. Use spaces to separate multiple file system names.

When a file system is specified in this option, you also should use the next option to specify the minimum number of free blocks that the log system file must have available. If that number of free blocks is not available, then the logging will be done on the next log host.

109. `Number of free blocks per log system file`

Choose this option and specify the minimum number of free blocks or enter 0 to have no minimum number of free blocks allowed for the file systems specified in the previous option. The valid values for the minimum number of free blocks are 0 to 2048000.

110. `Command line options for pbsyncd`

Choose this option and specify the command line options that you want. The available command line options for pbsyncd are:

`[-s] [-e logfile]`

`-e` Use logfile as the pbsyncd diagnostic log file.

`-s` Use the syslog facilities.

111. `Sync Protocol Timeout`

Choose this option and specify the length of time a synchronization client or server should wait for protocol checks to be completed. Enter the value of the synchronization protocol timeout (-1 to 1200000). 0 or -1 disables this timeout. -1 is the default.

112. `pbsyncd diagnostic log`

Choose this option and specify a location for it. This option enables you to specify the directory and file name for the pbsyncd diagnostic log.

113. `pbsync diagnostic log`

Choose this option and specify a location for the pbsync diagnostic log. This option enables you to specify the directory and file name for the pbsync diagnostic log.

114. `pbsync synchronization time interval (in minutes)`

Choose this option and specify the time interval in minutes between synchronizations.

115. `Add installed shells to /etc/shells`

Choose this option and specify whether to add installed shells. The operating system can validate your PowerBroker for Unix & Linux shells and then add them to `/etc/shells`.

`yes` Add installed shells to `/etc/shells`.

`no` Do not add installed shells to `/etc/shells`.

116. `PowerBroker pbksh diagnostic log`

Choose this option and specify a location for it. This option enables you to specify the directory and file name for the pbksh diagnostic log.

Note: This option is not available with PowerBroker Express.

117. `PowerBroker pbsh diagnostic log`

Choose this option and specify a location for it. This option enables you to specify the directory and file name for the pbsh diagnostic log.

Note: This option is not available with PowerBroker Express.

118. Stand-alone pblocald command

Choose this option and indicate whether to specify a stand-alone pblocald command. When a PowerBroker for Unix & Linux shell executes with the system in Single-User Mode, it is necessary to know which command to execute for some secured task requests that are handled by pblocald. This setting provides the PowerBroker for Unix & Linux shell, running in Single-User Mode, with the pblocald command to execute.

Specify the full command for the local daemon. For example:

```
/usr/sbin/[prefix]pblocald[suffix] -s
```

Note: When you specify the command, any installation prefix or suffix must be included. Specify none to specify no command for the local daemon in Single-User Mode.

This option is not available with PowerBroker Express.

119. Stand-alone root shell default iolog

Choose this option and specify a location for it. This option enables you to specify the directory and file name for the stand-alone root shell default I/O log.

120. Command line options for pbguid

Choose this option and specify the command line option that you want to use. The available syntax and command line option for pbguid are:

```
[-e logfile]
```

-e Use logfile as the pbguid log file.

none Erase all options.

122. pbguid and pbsguid diagnostic log

Choose this option and specify a location for it. This option enables you to specify the directory and file name for the pbguid diagnostic log

123. pbguid and pbsguid site configuration file

Choose this option and specify the location for the GUI site configuration file. The PowerBroker for Unix & Linux pbguid daemon uses a site file to store system-wide defaults for the PowerBroker for Unix & Linux GUI. If this file is not specified, then the system-wide GUI defaults are not used. Enter the full path for the directory and file name for the pbguid site configuration file.

124. Use syslog?

Choose this option and specify whether to use the system syslog facility.

The PowerBroker for Unix & Linux programs can send errors reported by the Policy Server and local daemons to the syslog. If you decide to use the system's syslog facility, then you must ensure that the facility selected for use by PowerBroker for Unix & Linux is enabled according to your system's documentation.

- Specify "y" to use the system syslog facility.
- Specify "n" to not use the system syslog facility.

125. Syslog facility to use?

Choose this option and specify the syslog facility to use. For PowerBroker for Unix & Linux to use the syslog facility, it must be specified. The facilities that can be specified are:

- LOG_AUTH security/authorization messages
- LOG_AUTHPRIV security/authorization messages (Linux and Mac OS). Only supported in PowerBroker for Unix & Linux 7.1.0 and later.
- LOG_DAEMON daemon messages

- LOG_LOCAL0 local messages
- LOG_LOCAL1 local messages
- LOG_LOCAL2 local messages
- LOG_LOCAL3 local messages
- LOG_LOCAL4 local messages
- LOG_LOCAL5 local messages
- LOG_LOCAL6 local messages
- LOG_LOCAL7 local messages
- LOG_USER user messages

Note: The default [LOG_AUTH] is usually sufficient. The message severity level that is used by PowerBroker for Unix & Linux is LOG_INFO.

For more information, see your system's man pages about syslog() and syslog.conf.

126. Base daemon port number

Caution! Unlike individual daemon ports, the base port may not be a Unix/Linux domain socket or a program name. Any daemon port that is already set to either a Unix/Linux domain socket or program name will not be changed. However, the used port number will be skipped. For more information about assigning ports, see [Select Port Numbers](#).

Choose this option and do one of the following:

- If ports 24345 to 24350 are available for all of the PowerBroker for Unix & Linux daemon ports, then accept these ports and continue the installation.
- If those ports are not available, then do one of the following:
 - Specify an available port number that also has the next five sequential port numbers available to set all of the PowerBroker for Unix & Linux daemon ports. The specified value must be numeric and must fall within the range from 1024 to 65530 (inclusive).
 - The pbmasterd port is set to the specified value.
 - The pblocald port is set to the specified value + 1.
 - The pblogd port is set to the specified value + 2.
 - The pbguid port is set to the specified value + 3.
 - The pbsguid port is set to the specified value + 4.
 - The pbsyncd port is set to the specified value +5.
 - Use the following five options to set the port numbers individually for pbmasterd, pblocald, pblogd, pbguid, and pbsyncd.

127. pbmasterd port number

Choose this option and specify the port number for pbmasterd.

The PowerBroker for Unix & Linux Policy Server Host daemon (pbmasterd) requires a dedicated port number or Unix/Linux domain socket name to receive inbound secured task requests from submit hosts. See Caution in step [126](#).

128. `pblocald` port number

Choose this option and specify the port number for `pblocald`.

The PowerBroker for Unix & Linux run host daemon (`pblocald`) requires a dedicated port number or Unix/Linux domain socket name to receive inbound secured task requests from Policy Server hosts. See Caution in step [126](#).

129. `pblogd` port number

Choose this option and specify the port number for `pblogd`.

The PowerBroker for Unix & Linux log host daemon (`pblogd`) requires a dedicated port number or Unix/Linux domain socket name to receive inbound secured task requests from Policy Server and local daemons. See Caution in step [126](#).

130. `pbguid` port number

Choose this option and specify the port number for `pbguid`.

The PowerBroker for Unix & Linux GUI daemon (`pbguid`) requires a dedicated port number or Unix/Linux domain socket name to receive inbound requests from Web browsers. See Caution in step [126](#).

131. `Secure pbsguid` port number

Choose this option and specify the port number for `pbguid`.

The PowerBroker for Unix & Linux

GUI secure daemon (`pbsguid`) service requires a dedicated port number or Unix/Linux domain socket name to receive inbound requests from Web browsers. See Caution in step [126](#).

132. `pbsyncd` port number

Choose this option and specify the port number for `pbsyncd`.

The PowerBroker for Unix & Linux log synchronization daemon (`pbsyncd`) requires a dedicated port number or Unix/Linux domain socket name to receive inbound requests. See Caution in step [126](#).

133. `REST Service` port number

Choose the TCP/IP port number on which the REST service is listening, on the Primary Policy Manager.

134. Add entries to `'/etc/services'`

Choose this option and specify `y` to have the services entries added to `/etc/services`. PowerBroker for Unix & Linux must be able to look up the port numbers to be used by the various PowerBroker for Unix & Linux services. The port number lookup can be done from NIS after you manually create the appropriate NIS entries (see your operating system's documentation for instructions). Otherwise, these services should be listed in `/etc/services`.

Only ports that are specified by number for the PowerBroker for Unix & Linux daemons can have services added to `/etc/services`. Unix/Linux domain sockets and ports that are specified by name are not added to `/etc/services` by this installation procedure.

Note: On some systems you must put entries into your NIS services map (or reboot) because `inetd` ignores `/etc/services` after boot time.

135. Allow non-reserved port connections

Choose this option and choose one of the following:

- Specify `y` to allow non-reserved port connections.
- Specify `n` to disallow connections from non-reserved port connections.

136. Inbound port range

The `MinListeningPort` setting in the `pb.settings` file determines the lower bound on the originating port range that may be used to make PowerBroker for Unix & Linux connections on the listening side. The `MaxListeningPort` setting determines the upper bound on the originating port range that may be used to make PowerBroker for Unix & Linux connections on the listening side.

Choose this option and do the following:

- a. Specify the value of the minimum port number to listen on. The value of this setting must be between 1 and the current value of the `MaxListeningPort` setting (65535).
- b. Specify the value of the maximum port number to listen on. The value of this setting must be between the current value of the `MinListeningPort` setting (1025) and 65535.

137. Outbound port range

The `MinOutgoingPort` setting in the `pb.settings` file determines the lower bound on the originating port range that may be used to make PowerBroker for Unix & Linux connections on the originating side. The `MaxOutgoingPort` setting determines the upper bound on the originating port range that may be used to make PowerBroker for Unix & Linux connections on the originating side.

Choose this option and do the following:

- a. Specify the value of the minimum outbound port number to originate from. The value of this setting must be between 1 and 65535.
- b. Specify the value of the maximum outbound port number to originate from. The value of this setting must be between the current value of the `MinOutgoingPort` setting (600) and 65535.

Starting with version 8.0, the new default in `pbinstall` for the minimum value of the outbound port range was changed from 600 to 1025. However, if you don't set this value during the install and the keyword `'minoutgoingport'` is commented out in the `pb.settings`, the default used by the binaries will still be 600. This is in order to keep backward compatibility with older releases of PowerBroker for Unix & Linux.

140. PowerBroker network encryption options

Note: Before specifying which (or if any) file types are to be encrypted, see “Network Traffic and File Encryption” in the PowerBroker for Unix & Linux System Administration Guide.

Choose this option and do one of the following:

- Specify `none` to not use any network encryption. Optionally, you can type the start date and/or end date for not using any network encryption in the format: `yyyy/mm/dd`. Dates are evaluated in Universal Coordinated Time (UTC).
- To add a new network encryption option, do the following:

- a. Specify a to add a new network encryption option.
- b. Specify the encryption type from the list in the following table. The default for version 8.0 and later is aes-256, and for versions prior to 8.0 is des. The default (AES-256 or DES) is used if end dates are specified for the listed network encryption algorithm and they have all expired. If you do not want the default to be used, then specify a network encryption or none with no end date.

Table 2. Encryption Type

Algorithm	Encryption Type
none	none
DES	des 3des tripledes
AES	aes-16-16 (or aes-128) aes-16-24 (or aes-192) aes-16-32 (or aes-256) aes-24-16 aes-24-24 aes-24-32 aes-32-16 aes-32-24 aes-32-32
Blowfish	blowfish
Cast128	cast128
Gost	gost
Loki97	loki97
Saferplus	saferplus-16 saferplus-24 saferplus-32
Serpent	serpent-16 serpent-24 serpent-32
Threeway	threeway
Tiny	tiny
Twofish	twofish-16 twofish-24 twofish-32

- c. Type the full path and file name where PowerBroker for Unix & Linux is to place the encryption key file. The default is /etc/pb.key. PowerBroker for Unix & Linux requires a key file to use encryption. It is recommended that you specify the /etc directory for the encryption key file.
- d. Optional. Type the start date and/or end date for the encryption pair in the format: yyyy/mm/dd. Dates are evaluated in Universal Coordinated Time (UTC).

Warning! Administrators must ensure that all hosts are using the same encryption pair; otherwise, the hosts cannot communicate with each other.

 - Specify "e" to edit an existing network encryption option and specify the number of the network encryption option. You can edit any of the following items for the selected option:
 - Network [encryption type](#)
 - Location and file name for the encryption file
 - Start date for the encryption pair to take effect
 - End date for the encryption pair

- Specify "d" to delete an existing network encryption option and specify the number of the network encryption option to delete it.
- Specify "x" to exit this option.

141. PowerBroker event log encryption options

Choose this option and do one of the following:

- Specify none to not use any event log encryption. Optionally you may type the start date and/or end date for not using any event log encryption in the format: yyyy/mm/dd. Dates are evaluated in Universal Coordinated Time (UTC).
- To add a new event log encryption option, do the following:
 - a. Specify a to add a new event log encryption option.
 - b. Set the [encryption type](#). The default for version 8.0 and later is aes-256, and for versions prior to 8.0 is des.
 - c. Specify the full path and file name where PowerBroker for Unix & Linux is to place the encryption key file. The default is /etc/pb.key. PowerBroker for Unix & Linux requires a key file to use encryption. It is recommended that you specify the /etc directory for the encryption key file.
 - d. Optional. Type the start date and/or end date for the encryption pair in the format: yyyy/mm/dd. Dates are evaluated in Universal Coordinated Time (UTC).
- Specify e to edit an existing event log encryption option and specify the number of the event log encryption option. You can edit any of the following items for the selected option:
 - Event log [encryption type](#).
 - Location and file name for the encryption file
 - Start date for the encryption pair to take effect
 - End date for the encryption pair
- Specify "d" to delete an existing event log encryption option and specify the number of the event log encryption option to delete it.
- Specify x to exit this option.

142. PowerBroker I/O log encryption options

Choose this option and do one of the following:

- Specify none to not use any I/O log encryption. Optionally you may type the start date and/or end date for not using any I/O log encryption in the format: yyyy/mm/dd. Dates are evaluated in Universal Coordinated Time (UTC).
- To add a new I/O log encryption option, do the following:
 - a. Specify a to add a new I/O log encryption option.
 - b. Set the [encryption type](#). The default for version 8.0 and later is aes-256, and for versions prior to 8.0 is des.
 - c. Specify the full path and file name where PowerBroker for Unix & Linux is to place the encryption key file. The default is /etc/pb.key. PowerBroker for Unix & Linux requires a key file to use encryption. It is recommended that you specify the /etc directory for the encryption key file.
 - d. Optional. Type the start date and/or end date for the encryption pair in the format: yyyy/mm/dd. Dates are evaluated in Universal Coordinated Time (UTC).

Warning! Administrators must ensure that all hosts are using the same encryption pair; otherwise, the hosts cannot communicate with each other.

- Specify `e` to edit an existing I/O log encryption option and specify the number of the I/O log encryption option. You can edit any of the following items for the selected option:
 - I/O log [encryption type](#)
 - Location and file name for the encryption file
 - Start date for the encryption pair to take effect
 - End date for the encryption pair
- Specify `d` to delete an existing I/O log encryption option and specify the number of the I/O log encryption option to delete it.
- Specify `x` to exit this option.

143. `PowerBroker report encryption options`

Choose this option and do one of the following:

- Specify `none` to not use any report encryption. Optionally you may type the start date and/or end date for not using any report encryption in the format: `yyyy/mm/dd`. Dates are evaluated in Universal Coordinated Time (UTC).
- To add a new report encryption option, do the following:
 - a. Specify `a` to add a new report encryption option.
 - b. Set the [encryption type](#). The default for version 8.0 and later is `aes-256`, and for versions prior to 8.0 is `des`.
 - c. Specify the full path and file name where PowerBroker for Unix & Linux is to place the encryption key file. The default is `/etc/pb.key`. PowerBroker for Unix & Linux requires a key file to use encryption. It is recommended that you specify the `/etc` directory for the encryption key file.
 - d. Optional. Type the start date and/or end date for the encryption pair in the format: `yyyy/mm/dd`. Dates are evaluated in Universal Coordinated Time (UTC).

Warning! Administrators must ensure that all hosts are using the same encryption pair; otherwise, the hosts cannot communicate with each other.

- Specify `"e"` to edit an existing report encryption option and specify the number of the report encryption option. You can edit any of the following items for the selected option:
 - Report [encryption type](#)
 - Location and file name for the encryption file
 - Start date for the encryption pair to take effect
 - End date for the encryption pair
- Specify `"d"` to delete an existing report encryption option and specify the number of the report encryption option to delete it.
- Type `"x"` to exit this option.

144. `PowerBroker policy file encryption options`

Choose this option and do the following:

- Enter none to not use any policy file encryption.
- To use the policy file encryption options, do the following:
 - a. Set the [encryption type](#). The default for version 8.0 and later is aes-256, and for versions prior to 8.0 is des.
 - b. Specify the full path and file name where PowerBroker for Unix & Linux is to place the encryption key file. The default is /etc/pb.key. PowerBroker for Unix & Linux requires a key file to use encryption. It is recommended that you specify the /etc directory for the encryption key file.

145. PowerBroker settings file encryption type

Choose this option and do one of the following:

- Specify none to not use any settings file encryption.
- Specify one of the [encryption types](#)

146. PowerBroker REST API encryption options

Configure encryption for the REST service Application Key database.

Choose this option and do one of the following:

- Specify "none" to not use encryption for the REST keystore. Optionally you may type the start date and/or end date for not using any REST keystore encryption in the format: yyyy/mm/dd. Dates are evaluated in Universal Coordinated Time (UTC).
- To add a new REST keystore encryption option, do the following:
 - a. Specify "a" to add a new REST keystore encryption option.
 - b. Set the encryption type. The default for version 8.0 and later is aes-256, and for versions prior to 8.0 is des.
 - c. Specify the full path and file name where PowerBroker for Unix & Linux is to place the encryption key file. The default is /etc/pb.rest.key. PowerBroker for Unix & Linux requires a key file to use encryption. It is recommended that you specify the /etc directory for the encryption key file.
 - d. Optional. Type the start date and/or end date for the encryption pair in the format: yyyy/mm/dd. Dates are evaluated in Universal Coordinated Time (UTC).
- Specify "e" to edit an existing REST keystore encryption option and specify the entry number of the encryption option to change. You can edit any of the following items for the selected option:
 - REST keystore encryption type.
 - Location and file name for the encryption file
 - Start date for the encryption pair to take effect
 - End date for the encryption pair
- Specify "d" to delete an existing REST keystore encryption option and specify the entry number of the encryption option to delete.
- Specify "x" to exit this option.

147. Configure with Kerberos v5?

Choose this option and do one of the following:

- Specify n if Kerberos V5 is not used.

- Specify "y" to configure using Kerberos V5. You will need also to perform steps [148](#) through [152](#).

148. PowerBroker Policy Server Daemon Kerberos Principal

Choose this option and specify the pbmasterd Kerberos principal. The default is pbmasterd.

149. PowerBroker Local Daemon Kerberos Principal

Choose this option and specify the pblockd Kerberos principal. The default is pblockd.

150. PowerBroker Log Daemon Kerberos Principal

Choose this option and specify the pblogd Kerberos principal. The default is pblogd.

151. PowerBroker Sync Daemon Kerberos Principal

Choose this option and specify the pbsyncd Kerberos principal. The default is pbsyncd.

152. Kerberos Keytab File

Choose this option and specify the directory and file name for the Kerberos keytab file.

153. Enforce High Security Encryption

Enabling High Security will enforce configuration to adhere to FIPS 140-2 security. Non-FIPS compatible encryption and hashing algorithms will be disabled. SSL running in strict FIPS mode will be enabled, enhancing the security of the installation.

154. Use SSL?

Choose this option and do one of the following:

- Enter "y" to use SSL. When using SSL, you need to perform steps [155](#) to [176](#).
- Enter "n" to not use SSL and skip steps [155](#) to [176](#).

For more information about SSL, see "Secure Socket Layers and Public Key Infrastructure" in the *PowerBroker for Unix & Linux System Administration Guide*.

155. SSL Configuration?

Choose this option and do one of the following:

- Specify allownonssl to allow connections to and from non-SSL hosts.
- Specify clientcertificates to require client certificates.
- Specify requiressl to allow communication among PowerBroker for Unix & Linux components without requiring PowerBroker for Unix & Linux client certificates. This option is not compatible with the AllowNonSSL option.
- Specify none to clear all existing parameters.

156. SSL pbrun Certificate Authority Directory?

Choose this option and do one of the following:

- Specify the directory location for the SSL pbrun certificate authority files.
- Specify none to not specify a directory for the SSL pbrun certificate authority file. If you do not specify a directory, then you must specify the full path and file name for the SSL pbrun certificate authority file in the next step.

157. SSL pbrun Certificate Authority File?

Choose this option and do one of the following:

- Specify the file name for the SSL pbrun certificate authority file. If you did not specify a directory in the previous step, then you need to provide the full path and file name.
- Specify none to not specify a filename for the SSL pbrun certificate authority file.

Caution! Failure to specify this file name will result in failed communication negotiation.

158. `SSL pbrun Cipher List?`

SSL provides a variety of algorithms that can be used for encryption. This option enables you to restrict the set of encryption algorithms that are used by pbrun for server communication to a subset of those ciphers that are available to SSL.

Choose this option and do one of the following:

- Specify ALL to allow all ciphers to be used from the list in the following table:

Table 3. List of SSL pbrun Ciphers

NULL-MD5	NULL-SHA
EXP-RC4-MD5	RC4-MD5
RC4-SHA	EXP-RC2-CBC-MD5
EXP-DES-CBC-SHA	DES-CBC-SHA
DES-CBC3-SHA	EXP-EDH-DSS-DES-CBC-SHA
EDH-DSS-CBC-SHA	EDH-DSS-DFS-CBC3-SHA
EXP-EDH-RSA-DES-CBC-SHA	EDH-RSA-DES-CBC-SHA
EDH-RSA-DES-CBC3-SHA	

- Specify one or more of the ciphers. If more than one cipher is specified, then type a space between the ciphers.

159. `SSL pbrun Certificate Directory?`

Choose this option and do one of the following:

- Specify the directory location for the SSL pbrun certificate file.
- Specify none to not specify a directory for the SSL pbrun certificate file. If you do not specify a directory, then you must specify the full path and file name for the SSL pbrun certificate file in the next step.

160. `SSL pbrun Certificate File?`

Choose this option and do one of the following:

- Specify the file name for the SSL pbrun certificate file. If you did not specify a directory in the previous step, you need to provide the full path and file name.
- Specify none to not specify a file name for the SSL pbrun certificate file.

Caution! Failure to specify this file name will result in failed communication negotiation.

161. `SSL pbrun Private Key Directory?`

Choose this option and do one of the following:

- Specify the directory for the SSL pbrun private key file.
- Specify none to not specify a directory for the SSL pbrun private key file. If you do not specify a directory, you need to provide the full path and file name in the next step.

162. SSL pbrun Private Key File?

Choose this option and do one of the following:

- Specify the file name for the SSL pbrun private key file. This is the PEM-formatted private key for the client certificate file. If you did not specify a directory in the previous step, then you need to provide the full path and file name.
- Specify none to not specify a filename for the SSL pbrun private key file.
Caution! Failure to specify this file name will result in failed communication negotiation.

163. SSL pbrun Certificate Subject Checks?

The `sslpbrunverifysubject` setting enables strings or substrings of the subjects of SSL certificates to be checked and accepted by pbrun from pbmasterd.

Choose this option and do one of the following:

- Specify the string or substring to check in the SSL pbrun certificate subject. If the specified string or substring finds a match in the certificate subject, then the connection proceeds; otherwise, the connection fails.
- Specify none to remove all checks.

164. SSL Server Certificate Authority Directory?

Choose this option and do one of the following:

- Specify the directory for the SSL server certificate authority file.
- Specify none to not specify a directory for the SSL server certificate file. If you do not specify a directory, then you need to provide the full path and file name for the SSL server certificate authority directory in the next step.

165. SSL Server Certificate Authority File?

Choose this option and do one of the following:

- Specify the file name for the SSL server certificate authority file. If you did not specify a directory in the previous step, then you need to provide the full path and file name.
- Specify none to not specify a SSL server certificate authority file.

Caution! Failure to specify this file name will result in failed communication negotiation.

166. SSL Server Cipher List?

OpenSSL provides a variety of algorithms which can be used for encryption. This option enables you to restrict the set of encryption algorithms that are used by the SSL server for communication to a subset of those ciphers that are available to OpenSSL.

Choose this option and do one of the following:

- Specifying ALL allows all ciphers in the following table to be used

Table 4. List of SSL pbrun Ciphers

NULL-MD5	NULL-SHA
EXP-RC4-MD5	RC4-MD5
RC4-SHA	EXP-RC2-CBC-MD5
EXP-DES-CBC-SHA	DES-CBC-SHA
DES-CBC3-SHA	EXP-EDH-DSS-DES-CBC-SHA
EDH-DSS-CBC-SHA	EDH-DSS-DFS-CBC3-SHA
EXP-EDH-RSA-DES-CBC-SHA	EDH-RSA-DES-CBC-SHA
EDH-RSA-DES-CBC3-SHA	

- Specify one or more of the ciphers. If more than one cipher is specified, type a space between the ciphers.

167. SSL Server Certificate Directory?

Choose this option and do one of the following:

- Specify the directory for the SSL server certificate file.
- Specify none to not specify a directory for the SSL server certificate file. If you do not specify a directory, then you need to provide the full path and file name for the SSL server certificate file in the next step.

168. SSL Server Certificate File?

Choose this option and do one of the following:

- Specify the file name for the SSL server certificate file. If you did not specify a directory in the previous step, you need to provide the full path and file name.
- Specify none to not specify a SSL server certificate file name.

Caution! Failure to specify this file name will result in failed communication negotiation.

169. SSL Server Private Key Directory?

Choose this option and do one of the following:

- Specify the directory for the SSL server private key file.
- Specify none to not specify a directory for the SSL server private key file. If you do not specify a directory, then you need to provide the full path and file name for the SSL server private key file in the next step.

170. SSL Server Private Key File?

Choose this option and do one of the following:

- Specify the file name for the SSL server private key file. If you did not specify a directory in the previous step, then you need to provide the full path and file name.
- Specify none to not specify the SSL server private key file name.

Caution! Failure to specify this file name will result in failed communication negotiation.

171. SSL Server Certificate Subject Checks?

Choose this option and do one of the following:

- Specify the string or substring to check in the SSL server certificate subject. If the specified string or substring finds a match in the certificate subject, then the connection proceeds; otherwise, the connection fails.
- Specify none to remove all checks.

172. PowerBroker SSL Certificate Country Code

The Country Code used when creating client x509 certificates.

173. PowerBroker SSL Certificate State/Province

The State/Province used when creating client x509 certificates.

174. PowerBroker SSL Certificate Location/Town

The general location or town used when creating client x509 certificates.

175. PowerBroker SSL Certificate Organizational Unit

The organizational unit used when creating client x509 certificates.

176. PowerBroker SSL Certificate Organization

The organization used when creating client x509 certificates.

177. Configure PowerBroker with LDAP?

Choose this option and do one of the following:

- Specify `n` to not enable PowerBroker for Unix & Linux to use LDAP
- Specify `y` to enable PowerBroker for Unix & Linux to use LDAP.

178. Install BeyondTrust built-in third-party libraries?

Choose this option and do one of the following:

- Specify `y` to install the BeyondTrust built-in third-party libraries.
- Specify `n` to not install BeyondTrust built-in third party libraries.

Note: If you are using LDAP, Kerberos, or SSL, then you need to install third-party libraries. You can install the BeyondTrust third-party libraries or your own. It is recommended that you use the BeyondTrust third-party libraries. For more information, see [Third-Party Libraries](#).

179. BeyondTrust built-in third-party library directory?

Choose this option and specify the directory for the BeyondTrust built-in third-party libraries. You also need to specify a directory for your own built-in libraries in step [188](#).

Caution! For HP (PA-RISC) and AIX, see [Third-Party Library File Names and Locations](#).

180. Kerberos shared library directory

Choose this option and specify the directory for the Kerberos shared libraries. If you do not specify a directory, then you need to provide the full path and file name in steps [181](#) through [184](#).

181. Kerberos libkrb5 shared library file name

Choose this option and specify the file name for the Kerberos libkrb5 shared library. If you did not specify a directory in step [180](#), you need to provide the full path and file name.

182. Kerberos libgssapi_krb5 shared library file name

Choose this option and specify the file name for the Kerberos libgssapi_krb5 shared library. If you did not specify a directory in step [180](#), you need to provide the full path and file name.

183. Kerberos libcom_err shared library file name

Choose this option and enter the file name for the Kerberos libcom_err shared library. If you did not specify a directory in step [180](#), you need to provide the full path and file name.

184. Kerberos libk5crypto shared library file name

Choose this option and specify the file name for the Kerberos libk5crypto shared library. If you did not specify a directory in step [180](#), you need to provide the full path and file name.

185. SSL shared library default directory

Choose this option and specify the file name for the SSL shared libraries. If you do not specify a directory, you need to provide the full path and file name in steps [186](#) through [187](#).

186. SSL libssl shared library file name

Choose this option and specify the file name for the SSL libssl shared library. If you did not specify a directory in step [185](#), you need to provide the full path and file name.

187. SSL libcrypto shared library file name

Choose this option and specify the file name for the SSL libcrypto shared library. If you did not specify a directory in step [185](#), you need to provide the full path and file name.

188. LDAP shared library default directory

Choose this option and specify the directory for the LDAP shared libraries. If you do not specify a directory, you need to provide the full path and file name in the next steps.

189. LDAP libldap shared library file name

Choose this option and specify the file name for the LDAP libldap shared library. If you did not specify a directory in step [188](#), you need to provide the full path and file name.

190. LDAP liblber shared library file name

Choose this option and specify the file name for the LDAP liblber shared library. If you did not specify a directory in step [188](#), you need to provide the full path and file name.

191. Use PAM?

PowerBroker for Unix & Linux enables the use of PAM (Pluggable Authentication Modules) when PowerBroker for Unix & Linux asks for password confirmation.

The authentication and account management portions of this service are invoked whenever PowerBroker for Unix & Linux verifies a password.

Note: For Mac OS X, PAM must be configured. Otherwise, the PowerBroker for Unix & Linux user and password policy functions will not work. These functions are listed in “User and Password Functions” in the *PowerBroker for Unix & Linux Policy Language Guide*.

- PAM is used on a Policy Server host when the `getuserpasswd()` and `getgrouppasswd()` policy functions are invoked and this setting is set to `y`.
- PAM is used on a submit host when the policy calls the `submitconfirmuser()` policy language function and this setting is set to `y`.

- PAM is used on a run host when the policy sets the `runconfirmuser` policy language variable to `TRUE` and this setting is set to `y`.

Choose this option and do one of the following:

- Specify `y` to use PAM PowerBroker for Unix & Linux processing on this machine. You will need also to perform the next PAM-related steps.
- Specify `n` to not use PAM PowerBroker for Unix & Linux processing on this machine.

192. `PAM service for password verification`

When PowerBroker for Unix & Linux uses PAM for password verification, it is necessary to specify a PAM service name to use for the password verification.

Note: PAM uses its own password prompts, but these are often not as informative as the prompts that are provided by the PowerBroker for Unix & Linux policy language. When PAM is in use, the PowerBroker for Unix & Linux prompt is presented to the user, and PAM displays its own prompts as needed.

Choose this option and do one of the following:

- Specify the PAM service name to use PAM for password verification.
- Specify `none` to use the PowerBroker for Unix & Linux traditional internal password verification.

193. `PAM session service`

When PAM is active, this setting defines the PAM service to invoke when PowerBroker for Unix & Linux starts or ends a terminal session. The `session open`, `session close`, and `account management` portions of the service are invoked when PowerBroker for Unix & Linux starts a terminal session using `pblocald`.

Choose this option and do one of the following:

- Specify the name of the PAM session service.
- Specify `none` to not use the PAM session service.

194. `PAM suppress PowerBroker password prompt`

When PAM is active, this setting defines whether or not to suppress PowerBroker for Unix & Linux password prompting.

Choose this option and do one of the following:

- Specify `y` to suppress PowerBroker for Unix & Linux password prompting (default).
- Specify `n` to not suppress PowerBroker for Unix & Linux password prompting.

195. `PAM library file name`

Choose this option and specify the PAM `libpam` library file name, including the directory path. The default file name varies according to the operating system. The notation that is used on AIX to specify the OS-provided library is the following: `libpam.a(shr.o)`

196. `Call pam_setcred?`

This option is only available if set Run Hosts to `y` in step [8](#) and you enable PAM in step [191](#).

The `pam_setcred` function is used to establish possible additional credentials of a user. This is useful for Solaris Projects and other scenarios. This is only used in conjunction with `pamsessionsservice`. This is currently not used to delete credentials.

Choose this option and do one of the following:

- Specify y to enable additional user credentials.
- Specify n to disable this feature (default).

Note: This option is not available with PowerBroker Express.

197. Enable non-PAM Solaris Projects?

This item is available only on Solaris 9 and above systems and if you set Run Hosts to y in step [8](#).

For Solaris 9 and 10, when Solaris Projects are to be used without using the PAM support (pam, pamsessionsservice, and pam_setcred), enable the non-PAM support by setting the enablesolarisprojects keyword to yes. For Solaris 9, PAM is not able to set a project other than the default project. For Solaris 9, with a project specified on the command line or the PowerBroker for Unix & Linux Policy, set the enablesolarisprojects keyword to yes (regardless of the PAM settings).

Choose this option and do one of the following:

- Specify y to enable non-PAM Solaris projects.
- Specify n to disable non-PAM Solaris projects (default).

Note: This option is not available with PowerBroker Express.

198. Solaris Projects library file name

This item is available only on Solaris 9 and above systems. The default Solaris Projects libproject library file name for this system is: /usr/lib/libproject.so

The following is a sample listing of libproject* files in the /usr/lib directory: (your system may be different):

```
lrwxrwxrwx 1 root root 17 Mar 23 2007 /usr/lib/libproject.so ->
./libproject.so.1
-rwxr-xr-x 1 root bin 53416 Jan 22 2005 /usr/lib/libproject.so.1
```

Choose this option and do one of the following:

- Enter the Solaris Projects libproject library file name including its path.
- Enter none.

Note: This option is not available with PowerBroker Express.

199. Allow Remote Jobs?

When this option is set to n, PowerBroker for Unix & Linux prohibits the control of remotely executed jobs as follows:

- On a Policy Server host, requests that have different submit host and run host names are automatically rejected. The runhost policy variable is set to read only.
- On a submit host, the -h option for the pbrun command is disabled, and the runhost variable of the request is set to the IP address of the submit host.
- On a run host, all requests that do not originate from the Run Host are rejected. Choose this option and do one of the following:
 - Specify y to allow remote jobs. This setting is the default.
 - Specify n to not allow remote jobs.

200. PowerBroker UNIX Domain Socket directory

When PowerBroker for Unix & Linux determines that communication may occur using Unix/Linux domain sockets, there must be a protected directory that contains the sockets used for reconnects and backconnects. Using Unix/Linux domain sockets for communication between daemons on the same machine should be more efficient than TCP socket communications.

The directory that is specified for PowerBroker for Unix & Linux Unix/Linux domain sockets must be protected from non-root read and write access, and each of the parent directories must be protected from non-root write access.

Choose this option and specify the directory for the PowerBroker for Unix & Linux Unix/Linux domain socket.

201. Reject Null Passwords?

Choose this option and do one of the following:

- Specify `n` to match an entered null password to any existing password.
- Specify `y` to require the user to exactly match the password.

202. Enable TCP keepalives?

PowerBroker for Unix & Linux enables the communication TCP connections to use the TCP stack's keepalive feature. TCP keepalives can be useful in cases where a firewall keeps track of idle TCP connections and terminates the sessions prematurely.

Choose this option and do one of the following:

- Specify `n` to disable TCP keepalive signals.
- Specify `y` to enable TCP keepalive signals.

203. Name Resolution Timeout

PowerBroker for Unix & Linux attempts to obtain fully qualified domain names when a `pblocald`, `pblogd`, `pbbmasterd`, or `pbrun` session is started. This setting defines the timeout period (in seconds) to be used for the request to expire.

Choose this option and do one of the following:

- Set the value to 0 to disable this feature (default).
- Set the value from 1 to 7200 to define the number of seconds to use for the timeout period.

Completing the Installation

After you finish making menu choices, do the following to complete the installation:

1. Use the `c` command to continue the installation.
2. A prompt asks if all of the installation settings are correct. If they are correct, then specify `y`. If they are not correct, then specify `n`, make the necessary changes, and continue the previous step.
3. A prompt asks if you want to view the installation script. Specify `n`.
Caution! This option is intended for troubleshooting by BeyondTrust Technical Support; the generated installation script contains thousands of lines of code.
4. A prompt asks if you want to install PowerBroker for Unix & Linux now. Press Enter to accept the default of `y`.
5. The PowerBroker for Unix & Linux installation script now executes and installs PowerBroker for Unix & Linux components on this machine.

6. If a PowerBroker for Unix & Linux policy file exists, it is not modified. If you do not have a policy file, then create a policy file using the following command:

```
touch /opt/pbul/policies/{prefix}pb.conf{suffix}
```

Caution! An empty policy file rejects all PowerBroker for Unix & Linux commands. For information about writing policy files, see the PowerBroker for Unix & Linux Policy Language Guide.

7. Change the permissions on the policy file so that it can be read by root only:

```
chmod 600 /opt/pbul/policies/pb.conf
```

The installation is now complete.

Example of a pbininstall Execution

The following is an example of a pbininstall execution:

```
/usr/local/lib/pbbuilder will be created as part of the installation
/etc/pb.key exists.. taking a copy...
```

```
Checking disk space...
```

```
... mountpoints are
/ /dev /net/build/build /net/nethome/nethome/tmp
/net/nethome/nethome/user /pbis
```

```
... local mount points are
/ /dev
```

```
Mount Point Needed Available Flag
/ 27117 359448716 works
```

```
Disk Free space on selected mountpoints appears to be okay.
```

```
Are all the installation settings correct [yes]? Creating the installation script:
'/opt/symark/powerbroker/v8.0/pbx86_64_linuxA-8.0.0-06/install/PowerBroker_Install'
An install script has been made that will install BeyondTrust PowerBroker
```

```
according to your settings. View the install script [no]?
Install BeyondTrust PowerBroker now [yes]?
```

```
Executing '/opt/symark/powerbroker/v8.0/pbx86_64_linuxA-8.0.0-06/install/PowerBroker_Install'
Creating settings file /etc/pb.settings
Removing PowerBroker service definitions (if any) from /etc/services. Adding PowerBroker service
definitions to /etc/services.
Removing any PowerBroker definitions from SuperDaemon xinetd file
/etc/xinetd.conf
Adding PowerBroker definitions to SuperDaemon configurations /etc/xinetd.conf. Installed
/usr/lib/beyondtrust/pb/libcom_err.so.3.0
```



```

Installed /usr/lib/beyondtrust/pb/libgssapi_krb5.so.2.2 Installed
/usr/lib/beyondtrust/pb/libk5crypto.so.3.0 Installed /usr/lib/beyondtrust/pb/libkrb5.so.3.2 Installed
/usr/lib/beyondtrust/pb/libcrypto.so.0.9.8 Installed /usr/lib/beyondtrust/pb/libssl.so.0.9.8 Installed
/usr/lib/beyondtrust/pb/liblber-2.3.so.0.2.12 Installed /usr/lib/beyondtrust/pb/libldap-2.3.so.0.2.12
Installed /usr/lib/beyondtrust/pb/libcurl.so.4.3.0
Created symbolic link /usr/lib/beyondtrust/pb/libcom_err.so.3 Created symbolic link
/usr/lib/beyondtrust/pb/libcom_err.so Created symbolic link /usr/lib/beyondtrust/pb/libgssapi_krb5.so.2
Created symbolic link /usr/lib/beyondtrust/pb/libgssapi_krb5.so Created symbolic link
/usr/lib/beyondtrust/pb/libk5crypto.so.3 Created symbolic link /usr/lib/beyondtrust/pb/libk5crypto.so
Created symbolic link /usr/lib/beyondtrust/pb/libkrb5.so.3
Created symbolic link /usr/lib/beyondtrust/pb/libkrb5.so Created symbolic link
/usr/lib/beyondtrust/pb/libcrypto.so.0 Created symbolic link /usr/lib/beyondtrust/pb/libcrypto.so
Created symbolic link /usr/lib/beyondtrust/pb/libssl.so.0 Created symbolic link
/usr/lib/beyondtrust/pb/libssl.so
Created symbolic link /usr/lib/beyondtrust/pb/liblber-2.3.so.0 Created symbolic link
/usr/lib/beyondtrust/pb/liblber-2.3.so Created symbolic link /usr/lib/beyondtrust/pb/libldap-2.3.so.0
Created symbolic link /usr/lib/beyondtrust/pb/libldap-2.3.so Created symbolic link
/usr/lib/beyondtrust/pb/libcurl.so.4 Created symbolic link /usr/lib/beyondtrust/pb/libcurl.so Installed
pbrun as /usr/local/bin/pbrun
Installed /usr/local/man/man1/pbrun.1 Installed pbssh as /usr/local/bin/pbssh Installed
/usr/local/man/man1/pbssh.1 Installed pbrunssh as /usr/local/bin/pbrunssh Installed pbmasterd as
/usr/sbin/pbmasterd Installed /usr/local/man/man8/pbmasterd.8
Installed pbfwdevents as /usr/sbin/pbfwdevents Installed /usr/local/man/man8/pbfwdevents.8 Installed
pblocald as /usr/sbin/pblocald Installed /usr/local/man/man8/pblocald.8 Installed pblogd as
/usr/sbin/pblogd

Installed /usr/local/man/man8/pblogd.8 Installed pbguid as /usr/sbin/pbguid Installed
/usr/local/man/man8/pbguid.8 Installed pbsyncd as /usr/sbin/pbsyncd Installed
/usr/local/man/man8/pbsyncd.8 Installed pbencode as /usr/sbin/pbencode Installed
/usr/local/man/man8/pbencode.8 Installed pbhostid as /usr/sbin/pbhostid Installed
/usr/local/man/man8/pbhostid.8 Installed pblicense as /usr/sbin/pblicense Installed
/usr/local/man/man8/pblicense.8 Installed pbpasswd as /usr/sbin/pbpasswd Installed
/usr/local/man/man8/pbpasswd.8 Installed pbsum as /usr/sbin/pbsum Installed /usr/local/man/man8/pbsum.8
Installed pbbench as /usr/local/bin/pbbench Installed /usr/local/man/man1/pbbench.1 Installed pbcheck as
/usr/sbin/pbcheck Installed /usr/local/man/man8/pbcheck.8 Installed pbcall as /usr/local/bin/pbcall
Installed pbless as /usr/local/bin/pbless Installed /usr/local/man/man1/pbless.1 Installed pbmg as
/usr/local/bin/pbmg Installed /usr/local/man/man1/pbmg.1 Installed pbnvi as /usr/local/bin/pbnvi
Installed /usr/local/man/man1/pbnvi.1 Installed pbumacs as /usr/local/bin/pbumacs Installed
/usr/local/man/man1/pbumacs.1 Installed pbvi as /usr/local/bin/pbvi Installed /usr/local/man/man1/pbvi.1
Installed pbkey as /usr/sbin/pbkey
Installed /usr/local/man/man8/pbkey.8 Installed pblog as /usr/sbin/pblog Installed
/usr/local/man/man8/pblog.8 Installed pbreplay as /usr/sbin/pbreplay Installed
/usr/local/man/man8/pbreplay.8 Installed pbmerge as /usr/sbin/pbmerge Installed
/usr/local/man/man8/pbmerge.8 Installed pbsync as /usr/sbin/pbsync Installed
/usr/local/man/man8/pbsync.8 Installed pbping as /usr/sbin/pbping Installed /usr/local/man/man8/pbping.8
Installed pbprint as /usr/sbin/pbprint Installed /usr/local/man/man8/pbprint.8 Installed pbksh as
/usr/local/bin/pbksh Installed pbsh as /usr/local/bin/pbsh Installed pbreport as /usr/sbin/pbreport
Installed /usr/local/man/man8/pbreport.8 Installed pbuvqrpq as /usr/sbin/pbuvqrpq Installed
/usr/local/man/man8/pbuvqrpq.8 Installed pbversion as /usr/sbin/pbversion Installed
/usr/local/man/man8/pbversion.8 Installed /usr/local/man/man8/pbinstall.8 Installed
/usr/local/man/man8/pbuninstall.8

Installed /usr/local/man/man8/pbmakeremotetar.8 Installed /usr/local/man/man8/pbpatchinstall.8
Placing policy examples in '/usr/local/lib/pbbuilder'
Placing pbguid html help files in '/usr/local/lib/pbbuilder' Installing /etc/pb.key

```

```
Reloading SuperDaemon Configurations...
Done Reloading SuperDaemon Configurations...
```

```
-----
-----
Installing default role-based policy pbul_policy.conf and pbul_functions.conf in /opt/pbul/policies
The main policy pbul_policy.conf will be included in /opt/pbul/policies/pb.conf
```

```
Installed pbul_policy.conf as /opt/pbul/policies/pbul_policy.conf
-----
```

```
-----
You will have to edit the /opt/pbul/policies/pb.conf file now.
```

```
Installed pblighttpd as /usr/lib/beyondtrust/pb/rest/sbin/pblighttpd Installed pblighttpd-svc as
/usr/lib/beyondtrust/pb/rest/sbin/pblighttpd-svc Installed /usr/lib/beyondtrust/pb/rest/lib/mod_
access.so
```

```
Installed /usr/lib/beyondtrust/pb/rest/lib/mod_dirlisting.so Installed
/usr/lib/beyondtrust/pb/rest/lib/mod_fastcgi.so Installed /usr/lib/beyondtrust/pb/rest/lib/mod_
indexfile.so Installed /usr/lib/beyondtrust/pb/rest/lib/mod_staticfile.so Installed
/usr/lib/beyondtrust/pb/rest/././pbsudoers_server.so
```

```
Installed pbconfigd as /usr/lib/beyondtrust/pb/rest/sbin/pb900pbconfigd Installed pbrestcall as
/usr/sbin/pbrestcall
```

```
Starting pblighttpd-svc service.
```

```
BeyondTrust PowerBroker Installation terminated successfully.
```

Installation Example for PowerBroker Express

This section provides an example of the installation menu when installing PowerBroker Express. The menu you see may vary depending on the options you select during your installation. For further instructions and descriptions of each option, see [Using the Menu Options](#).

```
BeyondTrust PowerBroker Installation Menu
```

```
Opt Description [Value]
```

1. Install Everything Here (Demo Mode)? [no]
2. Install PowerBroker Express Policy Server Host? [yes]
3. Install PowerBroker Express Submit Host? [yes]
4. Path to PowerBroker Password Safe 'pkru... []
5. PowerBroker Password Safe certificate file [] (appears if #4 completed)
6. Primary failover PowerBroker Password S... [] (appears if #4 completed)
7. Install PowerBroker Express Log Host? [yes]
8. Install PowerBroker Express Synchroniza... [yes]
9. Install PowerBroker Express GUI Host? [yes]
10. Install PowerBroker Express Secure GUI ... [no]
11. Install PowerBroker Express man pages? [yes]
12. Will this host use a Log Host? [yes]
13. PowerBroker Express synchronization can... [yes]
14. PowerBroker Express daemon location [/usr/sbin]

15. Number of reserved spaces for submit pr... [80]¹
16. Administration programs location [/usr/sbin]
17. User programs location [/usr/local/bin]
18. GUI library directory [/usr/local/lib/pbbuilder]
19. Policy include (sub) file directory [/opt/pbul/policies]
20. User man page location [/usr/local/man/man1]
21. Admin man page location [/usr/local/man/man8]
22. Configure xinetd /etc/xinetd.conf? [yes]
23. PowerBroker Express license [Expires 2010/06/17]
24. Command line options for pbmasterd [-ar]
25. Policy Server Delay [500]
26. Policy Server Protocol Timeout [-1]
27. pbmasterd diagnostic log [/var/log/pbmasterd.log]
28. Eventlog filename [/var/log/pb.eventlog]
29. Policy file name [opt/pbul/policies/pb.conf]
30. Validate Submit Host Connections? [no]
31. List of PowerBroker Express Policy Server to ... [nodename]
32. pbssh diagnostic log? [none]
33. Suppress master host failover error mes... [yes]
34. List of PowerBroker Express Policy Server to ... [nodename]
35. List of PowerBroker Express Log Hosts [nodename]
36. Command line options for pblogd []
37. Log Host Delay [500]
38. Log Host Protocol Timeout [-1]
39. pblogd diagnostic log [/var/log/pblogd.log]
40. List of PowerBroker Express log reserve... [none]
41. Number of free blocks per log system fi... [0]
42. Command line options for pbsyncd []
43. Sync Protocol Timeout [-1]
44. pbsyncd diagnostic log [/var/log/pbsyncd.log]
45. pbsync diagnostic log [/var/log/pbsync.log]
46. pbsync synchronization time interval (in... [15]
47. Command line options for pbguid []
48. Command line options for secure pbsguid []
49. pbguid and pbsguid diagnostic log [/var/log/pbguid.log]
50. pbguid and pbsguid site configuration file [none]
51. Use syslog? [yes]
52. Syslog facility to use? [LOG_AUTH]
53. Base Daemon port number [24345]
54. pbmasterd port number [24345]
55. pblocald port number [24346]
56. pblogd port number [24347]
57. pbguid port number [24348]

¹This option is only available on Linux, AIX and Mac OSX and is not available on Solaris and HPUX.

58. Secure pbsguid port number [24349]
59. pbsyncd port number [24350]
60. Add entries to '/etc/services' [yes]
61. Allow non-reserved port connections [no]
62. Inbound Port range [1025-65535]
63. Outbound Port range [1025-65535]
64. unused
65. unused
66. PowerBroker Express network encryption ... [aes-256:keyfile=/etc/pb.key]
67. PowerBroker Express event log encryptio... [none]
68. PowerBroker Express I/O log encryption ... [none]
69. PowerBroker Express report encryption o... [none]
70. PowerBroker Express policy file encrypt... [none]
71. PowerBroker Express settings file encry... [none]
72. Configure with Kerberos v5? [yes]
73. PowerBroker Express Policy Server Daemon Kerbe... [pbmasterd]
74. PowerBroker Local Daemon Kerberos Princ... [pblocald]
75. PowerBroker Express Log Daemon Kerberos... [pblogd]
76. PowerBroker Express Sync Daemon Kerbero... [pbsyncd]
77. Kerberos Keytab File [/etc/krb5.keytab]
78. Use SSL? [yes]
79. SSL Configuration? [none]
80. SSL pbrun Certificate Authority Directory? [none]
81. SSL pbrun Certificate Authority File? [none]
82. SSL pbrun Cipher List? [ALL]
83. SSL pbrun Certificate Directory? [none]
84. SSL pbrun Certificate File? [none]
85. SSL pbrun Private Key Directory? [none]
86. SSL pbrun Private Key File? [none]
87. SSL pbrun Certificate Subject Checks? [none]
88. SSL Server Certificate Authority Direct... [none]
89. SSL Server Certificate Authority File? [none]
90. SSL Server Cipher List? [ALL]
91. SSL Server Certificate Directory? [none]
92. SSL Server Certificate File? [none]
93. SSL Server Private Key Directory? [none]
94. SSL Server Private Key File? [none]
95. SSL Server Certificate Subject Checks? [none]
96. Configure PowerBroker Express with LDAP? [yes]
97. Install BeyondTrust built-in third-part... [yes]
98. BeyondTrust built-in third-party librar... [/usr/lib/beyondtrust/pb]
99. Kerberos shared library default directory [none]
100. Kerberos libkrb5 shared library filename [none]
101. Kerberos libgssapi_krb5 shared library ... [none]
102. Kerberos libcom_err shared library file... [none]

```

103.Kerberos libk5crypto shared library fil... [none]
104.SSL shared library default directory [none]
105.SSL libssl shared library filename [none]
106.SSL libcrypto shared library filename [none]
107.LDAP shared library default directory [none]
108.LDAP libldap shared library filename [none]
109.LDAP liblber shared library filename [none]
110.Use PAM? [yes]
111.PAM service for password verification [none]
112.PAM session service [none]
113.PAM suppress PowerBroker Express passwo... [yes]
114.PAM library file name [/lib/libpam.so.0]
115.Allow Remote Jobs? [yes]
116.PowerBroker Express UNIX Domain Socket ... [none]
117.Reject Null Passwords? [no]
118.Enable TCP keepalives? [no]
119.Name Resolution Timeout [0]
120.PowerBroker Identity Services Integration? [yes]
121.Enable failover event logging to PowerBroker Identity Services? [yes]
122.Enable successful connection event log ... [yes]
123.Enable event logging to PowerBroker Id ... [yes]
124.PowerBroker Identity Services shared lib .. [/opt/pbis/lib/libeventlog_...]
P for the previous menu page, C to continue, X to exit
Please enter a menu option [For technical support call 1-800-234-9072]

```

pbmakeremotetar Installation Script

Deployment of PowerBroker for Unix & Linux across multiple machines of the same platform type can be simplified by cloning the installations. Installation cloning is done by making a remote tarball using pbmakeremotetar, a menu-driven, interactive installation script.

pbmakeremotetar Installation Information

The section contains information about running an example pbmakeremotetar installation.

- pbmakeremotetar is used to clone an installed copy of PowerBroker for Unix & Linux so it can be quickly installed on other hosts that use the same PowerBroker for Unix & Linux flavor. The directory structure on the target systems must also be the same as on the host that is running pbmakeremotetar.
- pbmakeremotetar properly configures (as appropriate) /etc/services and the superdaemon configuration files (/etc/inetd.conf, /etc/xinetd.conf, or SMF).
- For Policy Server target installations, an initial installation (not a remote installation) must be done before any target remote installation. Doing so ensures that all licensing issues are handled properly.
- Different target system installation working directories should be used for different prefix and/or suffix versions of cloned installations.

- pbmakeremotetar scans the main policy file (by default /opt/pbul/policies/pb.conf from v9.4.3+ and /etc/pb.conf prior to v9.4.3) for included policy files and includes them in the tarball. If the main policy file is encrypted, pbmakeremotetar is not able to scan it for included policy files. Therefore, if the main policy file is encrypted, you must do one of the following:
 - Restore the unencrypted policy file before running the pbmakeremotetar installation script.
 - Specify each encrypted policy file in the editor session after answering y to the following prompt:


```
Do you wish to make changes to this list?
```
 - Manually move the encrypted files to the target systems.
 - For pbmakeremotetar/pbremoteinstall installations where integration with PowerBroker Identity Services is desired, if PowerBroker Identity Services is configured on the system where the PowerBroker for Unix & Linux instance is cloned, when the cloned instance is installed, if the PowerBroker Identity Services libraries are missing, then a warning message is displayed.

Remote Installations Using pbmakeremotetar

Remote installations using pbmakeremotetar perform the following three basic steps:

1. Execute pbmakeremotetar.
2. Make the created tar file available to the target system.
3. Unarchive the tar file and execute remote_unpack from that tar file.

Example of a pbmakeremotetar Execution

The following is an example of a pbmakeremotetar execution:

```
# ./pbmakeremotetar -a /opt/beyondtrust/pb.tar
```

```
Starting pbmakeremotetar main() from /opt/beyondtrust/powerbroker/v6.0/pbx86_
linuxB-6.0.0-01/install/.
```

```
pbmakeremotetar
```

```
This command is used to duplicate the current system's installation of
BeyondTrust PowerBroker to allow this duplication to be installed on one or
more identically configured systems.
```

```
x86_linuxB
```

```
Hit return or enter to continue...
```

```
Checking tar command for needed switches...
```

```
Done checking tar command for needed switches...
```

```
Making file /opt/beyondtrust/pb.tar for architecture x86_linuxB Reading
/etc/pb.cfg
```

```
Current additional files for deployment: [displays list of files]
```

```
Do you wish to make changes to this list [no]?
```

```
Building encapsulated tarball
```

```
/etc/pb.cfg
/etc/pb.conf
/etc/pb.key
/etc/pb.settings
/etc/pb.key
/opt/beyondtrust/powerbroker/v6.0/pbx86_linuxB-6.0.0-01/install/./pb.keyfiles
/opt/beyondtrust/powerbroker/v6.0/pbx86_linuxB-6.0.0-
  01/install/./pbremoteinstall
/opt/beyondtrust/powerbroker/v6.0/pbx86_linuxB-6.0.0-01/install/./pb_install_
support
/opt/beyondtrust/powerbroker/v6.0/pbx86_linuxB-6.0.0-
  01/install/./pbmakeremotetar
/opt/beyondtrust/powerbroker/v6.0/pbx86_linuxB-6.0.0-01/install/./pbuninstall
/opt/beyondtrust/powerbroker/v6.0/pbx86_linuxB-6.0.0-01/install/./sy_install_
support
/usr/lib/symark/pb/.BeyondTrustCreated
/usr/lib/symark/pb/.pbinstalls
/usr/lib/symark/pb/libcom_err.so
/usr/lib/symark/pb/libcom_err.so.3
/usr/lib/symark/pb/libcom_err.so.3.0
/usr/lib/symark/pb/libcrypto.so

/usr/lib/symark/pb/libcrypto.so.0
/usr/lib/symark/pb/libcrypto.so.0.9.7
/usr/lib/symark/pb/libgssapi_krb5.so
/usr/lib/symark/pb/libgssapi_krb5.so.2
/usr/lib/symark/pb/libgssapi_krb5.so.2.2
/usr/lib/symark/pb/libk5crypto.so
/usr/lib/symark/pb/libk5crypto.so.3
/usr/lib/symark/pb/libk5crypto.so.3.0
/usr/lib/symark/pb/libkrb5.so
/usr/lib/symark/pb/libkrb5.so.3
/usr/lib/symark/pb/libkrb5.so.3.2
/usr/lib/symark/pb/liblber-2.3.so
/usr/lib/symark/pb/liblber-2.3.so.0
/usr/lib/symark/pb/liblber-2.3.so.0.2.12
/usr/lib/symark/pb/libldap-2.3.so
/usr/lib/symark/pb/libldap-2.3.so.0
/usr/lib/symark/pb/libldap-2.3.so.0.2.12
/usr/lib/symark/pb/libssl.so
/usr/lib/symark/pb/libssl.so.0
/usr/lib/symark/pb/libssl.so.0.9.7
```

```
/usr/local/bin/pbbench
/usr/local/bin/pbcall
/usr/local/bin/pbksh
/usr/local/bin/pbless
/usr/local/bin/pbmg
/usr/local/bin/pbnvi
/usr/local/bin/pbrun
/usr/local/bin/pbsh
/usr/local/bin/pbumacs
/usr/local/bin/pbvi
/usr/local/man/man1/pbbench.1
/usr/local/man/man1/pbless.1
/usr/local/man/man1/pbmg.1
/usr/local/man/man1/pbnvi.1
/usr/local/man/man1/pbrun.1
/usr/local/man/man1/pbumacs.1
/usr/local/man/man1/pbvi.1
/usr/local/man/man8/pbcheck.8
/usr/local/man/man8/pbencode.8
/usr/local/man/man8/pbguid.8
/usr/local/man/man8/pbhostid.8
/usr/local/man/man8/pbkey.8
/usr/local/man/man8/pblicense.8
/usr/local/man/man8/pblocald.8
/usr/local/man/man8/pblog.8
/usr/local/man/man8/pblogd.8
/usr/local/man/man8/pbmasterd.8
/usr/local/man/man8/pbmerge.8
/usr/local/man/man8/pbpasswd.8
/usr/local/man/man8/pbprint.8
/usr/local/man/man8/pbreplay.8
/usr/local/man/man8/pbreport.8
/usr/local/man/man8/pbsum.8
/usr/local/man/man8/pbsync.8

/usr/local/man/man8/pbsyncd.8
/usr/local/man/man8/pbuvqrpq.8
/usr/sbin/pbcheck
/usr/sbin/pbencode
/usr/sbin/pbguid
/usr/sbin/pbhostid
/usr/sbin/pbkey
```



```

/usr/sbin/pblocald
/usr/sbin/pblog
/usr/sbin/pblogd
/usr/sbin/pbmasterd
/usr/sbin/pbmerge
/usr/sbin/pbpasswd
/usr/sbin/pbprint
/usr/sbin/pbreplay
/usr/sbin/pbreport
/usr/sbin/pbsum
/usr/sbin/pbsync
/usr/sbin/pbsyncd
/usr/sbin/pbuvqrpq
Building encapsulating tarball remote_unpack
pb.tar.tar

```

```
/opt/beyondtrust/pb.tar has been built
```

Make the Tar File Available to the Remote System

To make the tar file available to the remote system, you can use FTP (image mode), NFS, or any other mechanism as long as the security and integrity of the binary tar file is maintained.

If tar `-x` warns about a directory checksum error, then the tar file archive may be corrupt because it was copied in ASCII, not binary (or image) mode.

Untar the Remote Archive and Execute remote_unpack

When the tar file is made available to the remote system, a temporary working directory must be selected to unarchive the remote archive. An installation work directory other than `/tmp` should be selected (for the same reasons as with `pbinstall`). Unpacking the archive makes the encapsulated tar archive and a script called `remote_unpack` visible.

The `remote_unpack` script then executes. This script unpacks the encapsulated tar file (putting the files in their required places) and reconfigures the system files (`/etc/services` and the superdaemon configuration) for PowerBroker for Unix & Linux.

The following listing shows an example execution of the `remote_unpack` script:

```

# cd {workingdirectory}
# tar -xvf {tarfilename}.tar
x remote_unpack, 1250 bytes, 3 tape blocks
x tarfilename.tar.tar, 48152576 bytes, 94048 tape blocks
# ./remote_unpack

```

```
Deploying executable files...
```

```

x /usr/local/bin/pbrun, 4282296 bytes, 8364 tape blocks
x /usr/local/man/man1/pbrun.1, 2852 bytes, 6
tape blocks

```

```

x /usr/local/bin/pbbench, 3414416 bytes, 6669 tape blocks x /usr/local/man/man1/pbbench.1, 1152 bytes, 3
tape blocks x /usr/local/bin/pbless, 178964 bytes, 350 tape blocks
x /usr/local/man/man1/pbless.1, 743 bytes, 2 tape blocks x /usr/local/bin/pbmg, 52 bytes, 1 tape blocks
x /usr/local/man/man1/pbmg.1, 809 bytes, 2 tape blocks x /usr/local/bin/pbumacs, 52 bytes, 1 tape blocks
x /usr/local/man/man1/pbumacs.1, 832 bytes, 2 tape blocks x /usr/local/bin/pbvi, 212000 bytes, 415 tape
blocks
x /usr/local/man/man1/pbvi.1, 1107 bytes, 3 tape blocks x /usr/local/bin/pbcall, 3585880 bytes, 7004
tape blocks x /usr/sbin/pblocald, 4714020 bytes, 9208 tape blocks
x /usr/local/man/man8/pblocald.8, 1525 bytes, 3 tape blocks x /usr/sbin/pbcheck, 4202964 bytes, 8209
tape blocks
x /usr/local/man/man8/pbcheck.8, 2824 bytes, 6 tape blocks x /usr/sbin/pbhostid, 191596 bytes, 375 tape
blocks
x /usr/local/man/man8/pbhostid.8, 815 bytes, 2 tape blocks x /usr/sbin/pbkey, 187548 bytes, 367 tape
blocks
x /usr/local/man/man8/pbkey.8, 1113 bytes, 3 tape blocks x /usr/sbin/pblog, 3836692 bytes, 7494 tape
blocks
x /usr/local/man/man8/pblog.8, 5346 bytes, 11 tape blocks x /usr/sbin/pbpasswd, 186536 bytes, 365 tape
blocks
x /usr/local/man/man8/pbpasswd.8, 1413 bytes, 3 tape blocks x /usr/sbin/pbreplay, 3550320 bytes, 6935
tape blocks
x /usr/local/man/man8/pbreplay.8, 3522 bytes, 7 tape blocks x /usr/sbin/pbsum, 77872 bytes, 153 tape
blocks
x /usr/local/man/man8/pbsum.8, 853 bytes, 2 tape blocks x /usr/sbin/pbencode, 3163940 bytes, 6180 tape
blocks
x /usr/local/man/man8/pbencode.8, 927 bytes, 2 tape blocks x /usr/sbin/pbmasterd, 5505740 bytes, 10754
tape blocks
x /usr/local/man/man8/pbmasterd.8, 1578 bytes, 4 tape blocks x /usr/sbin/pblogd, 3956552 bytes, 7728
tape blocks
x /usr/local/man/man8/pblogd.8, 1015 bytes, 2 tape blocks x /usr/sbin/pbguid, 6537648 bytes, 12769 tape
blocks
x /usr/local/man/man8/pbguid.8, 2147 bytes, 5 tape blocks x /usr/local/lib/pbbuilder, 0 bytes, 0 tape
blocks
x /usr/local/lib/pbbuilder/.BeyondTrustCreated, 29 bytes, 1 tape blocks x
/usr/local/lib/pbbuilder/fileselect.html, 1075 bytes, 3 tape blocks
x /usr/local/lib/pbbuilder/iolog.html, 2346 bytes, 5 tape blocks x /usr/local/lib/pbbuilder/log.html,
1139 bytes, 3 tape blocks
x /usr/local/lib/pbbuilder/settings.html, 23014 bytes, 45 tape blocks x
/usr/local/lib/pbbuilder/variables.html, 34964 bytes, 69 tape blocks
x /usr/local/lib/pbbuilder/.BeyondTrustCreated, 29 bytes, 1 tape blocks x
/usr/local/lib/pbbuilder/fileselect.html, 1075 bytes, 3 tape blocks
x /usr/local/lib/pbbuilder/iolog.html, 2346 bytes, 5 tape blocks x /usr/local/lib/pbbuilder/log.html,
1139 bytes, 3 tape blocks
x /usr/local/lib/pbbuilder/settings.html, 23014 bytes, 45 tape blocks x
/usr/local/lib/pbbuilder/variables.html, 34964 bytes, 69 tape blocks
x /opt/beyondtrust/pb/install/pbremoteinstall, 3362 bytes, 7 tape blocks
x /opt/beyondtrust/pb/install/pbmakeremotetar, 14650 bytes, 29 tape blocks x
/opt/beyondtrust/pb/install/pbuninstall, 11565 bytes, 23 tape blocks
x /opt/beyondtrust/pb/install/pb_install_support, 13212 bytes, 26 tape blocks
x /opt/beyondtrust/pb/install/sy_install_support, 93560 bytes, 183 tape blocks
x /opt/beyondtrust/pb/install/platform, 5971 bytes, 12 tape blocks x /etc/pb.key, 1026 bytes, 3 tape
blocks
x /opt/beyondtrust/pb/install/pb.cfg, 1161 bytes, 3 tape blocks

```

```
x /opt/beyondtrust/pb/install/pb.cfg.sparc_solaris7, 2 bytes, 1 tape blocks x
/opt/beyondtrust/pb/install/pb.cfg.default, 2 bytes, 1 tape blocks
x /etc/pb.settings, 1915 bytes, 4 tape blocks
x /usr/local/man/man8/pbinstall.8, 6047 bytes, 12 tape blocks x /usr/local/man/man8/pbuninstall.8, 2569
bytes, 6 tape blocks
x /usr/local/man/man8/pbmakeremotetar.8, 4239 bytes, 9 tape blocks x /etc/pb.conf, 202 bytes, 1 tape
blocks
Configure System now? [yes]
Starting pbremoteinstall main() from /opt/beyondtrust//pb_xyzzy/pb/install Reading
/opt/beyondtrust/pb/install/pb.cfg
Reading /opt/beyondtrust/pb/install/pb.cfg.sparc_solaris7 Reading
/opt/beyondtrust/pb/install/pb.cfg.default
Removing PowerBroker service definitions (if any) from /etc/services. Removing PowerBroker service
definitions (if any) from /etc/services. Adding PowerBroker service definitions to /etc/services.
Looking for SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
Removing any PowerBroker definitions from SuperDaemon inetd file
/etc/inetd.conf
Adding PowerBroker definitions to SuperDaemon configurations
/etc/inetd.conf .
Reloading SuperDaemon Configurations...
Done Reloading SuperDaemon Configurations...
/opt/beyondtrust/pb/install/pbremoteinstall ... Done
```

pbpatchinstall Installation Script

BeyondTrust occasionally releases patches to the PowerBroker for Unix & Linux product that improve performance and fix problems. You install these patches with the pbpatchinstall installation script.

pbpatchinstall Installation Information

This section contains information about installing a PowerBroker for Unix & Linux patch with the pbpatchinstall script.

pbpatchinstall determines the current release of PowerBroker for Unix & Linux that is installed on the machine and whether the release is compatible with the current patch. Multiple patches can be installed.

Based on the type of PowerBroker for Unix & Linux host that is installed on the machine (Policy Server host, submit host, log host, and so forth), pbpatchinstall copies only the appropriate files to the appropriate directories to replace the existing files. pbpatchinstall makes a backup copy of all replaced files. These backup files are then available to restore the original files if the patch needs to be removed.

Note: All PowerBroker for Unix & Linux daemons running a process during the patch installation should be stopped before using pbpatchinstall and restarted after using pbpatchinstall.

After you extract a PowerBroker for Unix & Linux patch tarball file, the patch version becomes part of the directory path. For example, in the patch directory:

```
/opt/beyondtrust/powerbroker/v5.1/ pbx86_linuxA-5.1.2-03-sp1/install
```

the patch version is pbx86_linuxA-5.1.2-03-sp1.

The pbpatchinstall installation process performs the following:

- Inventories the PowerBroker for Unix & Linux installation, using prefixes and/or suffixes (if any). Use the -p and/or -s arguments if you want pbpatchinstall to use prefixes and/or suffixes.
- Validates the existence and version of the PowerBroker for Unix & Linux binary files that should be present for each component.
- Lists the PowerBroker for Unix & Linux components that are currently installed.

The PowerBroker for Unix & Linux patch release number must match the installed PowerBroker for Unix & Linux release number. If the release numbers do not match, a prompt is displayed, stating that the patch release does not match the existing PowerBroker for Unix & Linux release and asks if you want to install the patch release over the existing PowerBroker for Unix & Linux release. To complete the installation, type y.

Note: To run the patch installation without this prompt, use the -f argument.

Example of a pbpatchinstall Execution

The following is an example of a pbpatchinstall execution:

```
#pwd
/opt/beyondtrust/powerbroker/v5.1/pbx86_linuxB-5.1.1-03-spl/install
# ./pbpatchinstall

Starting pbpatchinstall from /opt/beyondtrust/powerbroker/v5.1/pbx86_linuxB
-5.1.1-03-spl/install/.x86_linuxB BeyondTrust PowerBroker Patch Installation
Checking MANIFEST against release directory Trying /etc/pb.settings
Settings are from file='/etc/pb.settings'
Reading /etc/pb.cfg
PowerBroker version 5.1.0-08 established from /etc/pb.cfg PowerBroker components currently installed:
run_host submit_host log_synchronization secure_gui_host utilities
pbksh log_sync_initiator

All installed binaries match PowerBroker version 5.1.0-08 Version is not evaluated for binaries pbuvqrpq
and pbnvi.

Patch release 5.1.1 does not match PowerBroker release 5.1.0
Install PowerBroker patch release 5.1.1 over PowerBroker release 5.1.0? [no] y Checking disk space...
... mountpoints are
/ /boot /data /dev /net/nethome/nethome/user

... local mount points are
/ /boot /data /dev

Mount Point Needed Available Flag
/ 1024 2921852 works
/data 2590 126953328 works
Disk Free space on selected mountpoints appears to be okay. Patched /usr/sbin/pbencode installed.
Patched /usr/local/bin/pbbench installed. Patched /usr/local/bin/pbrun installed. Patched
/usr/sbin/pbreport installed.
Patched /usr/local/man/man8/pbguid.8 installed. Patched /usr/local/bin/pbksh installed.
6 files patched, replaced files moved to /opt/beyondtrust/powerbroker/v5.1/pbx86_linuxB-5.1.1-03-
spl/bin_patchbkp
```

NOTE: In order to remove patch, directory `/opt/beyondtrust/powerbroker/v5.1/pbx86_linuxB-5.1.1-03-sp1/bin_patchbkgp` must be left in place.
`/etc/pb.cfg` updated with patch information. 5.1.1-03-sp1 patches installed.

Custom Installations

The preferred methods for installing PowerBroker for Unix & Linux are to use the command line `pbinstall` or `pbmakeremotetar`. In some instances, however, customer requirements may dictate some custom installation methods. This section covers several topics you should be aware of when planning a custom installation.

Before performing a custom installation of PowerBroker for Unix & Linux, several issues need to be taken into consideration:

- Third-party libraries
- Executable files
- `pb.settings` file
- `pb.key` file
- Superdaemon configuration update
- Policy files for Policy Server hosts

There are some concerns about file system accessibility when using remotely mounted file systems. If an installation initially references files on a system with a different name (due to network and/or NIC configurations), the target system may have problems referencing the files correctly on the original host.

Third-Party Libraries

The appropriate third-party libraries are required when PowerBroker for Unix & Linux is configured with Kerberos, SSL, or LDAP. For more information about third-party libraries, see [Third-Party Libraries](#).

Executable Files

Regardless of how PowerBroker for Unix & Linux is placed on multiple systems, the proper executable and supporting files for the flavor and functions of the system must be visible and executable on that system.

It is possible to place the target of the administration, user, daemon, and/or utility programs on a remotely mounted file system. If this is done, the following issues must be addressed:

- The correct flavor for a system must be visible in the path for the given system.
- The superuser owner and `suid` setting of `pbrun` must be handled properly.
- The remotely mounted file system must be very reliable.
- PowerBroker for Unix & Linux event, I/O, and daemon error logs are not supported when written to remotely mounted file systems.

Settings File

The `/etc/pb.settings` file must be properly configured for the functions that the new host is to perform, and the install scripts do this. When performing a custom install, each machine needs a correctly configured `/etc/pb.settings` file.

Key File

If encryption is used, then the pb.key file must be the same across all cooperating PowerBroker for Unix & Linux installations. This is typically a manual distribution (because the pb.key file can be compromised if it is not handled properly) except when performing a remote installation using the archive from pbmakeremotetar.

superdaemon Configuration

The superdaemons on the system must be configured for the PowerBroker for Unix & Linux daemon configuration. The PowerBroker for Unix & Linux installation performs this configuration automatically.

Policy Files for Policy Server Hosts

Policy files and their subfiles must be copied between Policy Server hosts so that all of the Policy Servers use the same policies.

PowerBroker for Unix & Linux, being an authentication tool and not a software distribution tool, does not automatically propagate policy files between Policy Server hosts. It is possible, and left as an exercise, to write procedures and policies that allow a central Policy Server host to propagate policy files to other Policy Server hosts.

Policy subfiles are copied if their name is specified as a constant. If the name is specified as a variable or string concatenation in the parent policy, then that policy is not copied by pbmakeremotetar and must be manually propagated to the target machines.

The policy subfile directory tree and directories referenced by the policies should be created to insure the multiple Policy Server hosts have the same directory tree.

Package Installer

The following sections on how to install PowerBroker for Unix & Linux on Solaris, Linux, HPUX and AIX using the system native package installer.

PowerBroker for Unix & Linux has several separate component packages for each Log Server, Run host, Policy Server, etc.

Starting with v9.0, the "Shared library" component package as well as the "REST API" component package need to be installed prior to installation of Policy Server, Log Server, GUI, Run Host and Submit host.

Solaris Package Installer

This section describes how to install PowerBroker for Unix & Linux using a package installer for Solaris 9 or 10 on an x86 or SPARC computer. Use the Solaris package installer if you want to do any of the following:

- Install PowerBroker for Unix & Linux using the Solaris Package Manager.
- Install PowerBroker Express using the Solaris Package Manager.
- Make the PowerBroker for Unix & Linux installation packages available on a JumpStart server to automate the installation of Solaris computers.

If you have a previous installation of PowerBroker for Unix & Linux, you must first uninstall it before installing PowerBroker Express. Likewise, if you have an installation of PowerBroker Express, you must uninstall it before installing PowerBroker for Unix & Linux.

The PowerBroker for Unix & Linux Solaris package installer that is described here is not compatible with the BeyondTrust PowerBroker V5.x packages. If the Symark PowerBroker Packages V5.x packages are installed, you must remove them before installing the PowerBroker for Unix & Linux Solaris packages.

Prerequisites

To use the Solaris package installer, you must have the following:

- Package tarball file for the appropriate PowerBroker for Unix & Linux flavor
 - Note:** For the Solaris package installer, the tarball files are cumulative. That is, an update tarball file contains a complete PowerBroker for Unix & Linux or PowerBroker Express installation. It is not necessary to install a baseline version of PowerBroker for Unix & Linux or PowerBroker Express before installing an update.
- Root access or superuser privileges
 - Note:** The Solaris package installer does not support prefix/suffix installations.

Planning Your Installation

When preparing to use the Solaris package installer, you should be familiar with the following concepts and restrictions:

Component packages - A PowerBroker for Unix & Linux component package is a Solaris datastream (.ds) file that installs a portion of the PowerBroker for Unix & Linux application. There are six PowerBroker for Unix & Linux component packages:

- BTPBmsth.ds - Contains the Policy Server host, pbsync, and pbsyncd.
- BTPBsbmh.ds - Contains the submit host and PowerBroker for Unix & Linux shells. This component package is not available for PowerBroker Express.
- BTPBrunh.ds - Contains the run host and PowerBroker for Unix & Linux utilities. This component package is not available for PowerBroker Express.
- BTPBlogh.ds - Contains the log host, pbsync, and pbsyncd.
- BTPBguih.ds - Contains the GUI host and secure GUI host.
- BTPBlibs.ds - Contains the shared libraries.

- BTPBexpr.ds - Contains pbssh. This package is only used for PowerBroker Express to install pbssh instead of pbrun. This package is not compatible with PowerBroker Server BTPBsbmh.ds and BTPBrunh.ds packages
- BTPBrest.ds - Contains the REST API files.
- BTPBrnsh.ds - Contains Registry Name Service files.
- BTPBlich.ds - Contains the License Server files.

Which component packages are required depends on the type of PowerBroker for Unix & Linux host you are creating, such as Policy Server host, log host, and so forth. You can select the types of PowerBroker for Unix & Linux hosts in the pbininstall installation menu, as shown in the following table.

Note: For PowerBroker Express, there will not be an option for a run host. For PowerBroker Express, the submit host option installs pbssh. using the PowerBroker BTPBexpr component package. For the other PowerBroker for Unix & Linux products, this option installs pbrun

Table 2. Required Components

Menu Selection	Required Components
Install everything here (demo mode)? = Yes	BTPBmstr
	BTPBrunh
	BTPBsbmh
	BTPBlogh
	BTPBguih
	BTPBlibs
Install PowerBroker Policy Server Host? = Yes	BTPBmstr
Install PowerBroker Run Host? = Yes	BTPBrunh PowerBroker Express does not use the run host package.
Install PowerBroker Submit Host? = Yes	BTPBsbmh For PowerBroker Express, the BTPBexpr package is installed.
Install PowerBroker Log Host? = Yes	BTPBlogh
Install PowerBroker GUI Host? = Yes	BTPBguih
Install PowerBroker Secure GUI Host? = Yes	BTPBguih
Install BeyondTrust built-in third-party libraries? = Yes	BTPBlibs
Install Registry Name Services Server? [yes]	BTPBrnsh.ds
Install License Server? [yes]	BTPBlich.ds

Configuration package - Solaris installation package that is used to install the following files:

- pb.settings
- pb.cfg

- pb.key (if applicable)
- pb.conf (for Policy Server hosts)
- Man pages for the pbinstall and pbcreatesolcfgpkg programs

The PowerBroker for Unix & Linux configuration package is created by the pbcreatesolcfgpkg program. The component packages must be installed before you install the configuration package.

Package name - Name of the installation package stored in the Solaris package manager database. For PowerBroker for Unix & Linux package installations, this name is the same as the package file name without the .ds extension.

Package administration file - Contains alternative settings that control how Solaris packages are installed.

Relocated base directory - The directory where the PowerBroker for Unix & Linux binary files and log files are installed. You can choose an alternative directory in which to install these files. For more information, see [Relocating the Base Directory](#).

pbinstall program - To create the PowerBroker for Unix & Linux settings files, you use the pbinstall program with the -z ("settings only") option. pbinstall -z only creates the settings files and is incompatible with the following command line options:

Table 3. Options Incompatible with pbinstall -z

Option	Description
-b	Runs pbinstall in batch mode.
-c	Skip the steps that process or update the PowerBroker for Unix & Linux settings file.
-e	Runs install script automatically by bypassing the menu step of pbinstall.
-i	Ignores previous pb.settings and pb.cfg files.
-p	Sets the pb installation prefix.
-s	Sets the pb installation suffix.
-u	Install the utility programs.
-x	Creates a log synchronization host (that is, installs pbsyncd).

When you execute pbinstall with the -z option, you can see two menu items that are not otherwise available:

Enter existing pb.settings path

Enables you to specify your own pb.settings file. pbinstall reads this settings file and populates the remaining menu choices. You can override some menu choices.

If set to none, then pbinstall does not read a settings file. The remaining menu choices are populated with default values.

Enter directory path for settings file creation

Enables you to specify an alternative output directory for the settings files. The default directory is:

`/unzip-dir/powerbroker/<version>/<flavor>/install/settings_files`

where unzip-dir is the directory where the package tarball file was unzipped.

The behavior of pbinstall -z depends on whether certain additional command line options are specified:

- If no other command line options are specified, pbinstall initially presents a short version of the installation menu (items 1–8 only). Depending on the choices you make in these items, further menu items become available.
- If command line options -g, -l, -m, -o, -r, or -w are specified, pbinstall presents an expanded version of the installation menu that reflects the host types that you are configuring.

For complete information about the pbinstall command-line options, see [pbinstall](#).

When running pbinstall with the -z option, the following menu items are pre-programmed and cannot be changed:

```
Install man pages? PowerBroker daemon location
Administration programs location User programs location
GUI library location
Policy include (sub) file directory User man page location
Admin man page location Policy filename
BeyondTrust built-in third-party library directory
```

In addition, the values of the following menu items determine the values of other menu items:

Table 4. Options Preset When Running pbinstall -z

Setting this menu option to Yes	Sets these values to Yes
Install PowerBroker Policy Server Host?	Install PowerBroker Synchronization? PowerBroker synchronization can be initiated from this host?
Install PowerBroker Run Host?	Install PowerBroker Utilities? Note: This is not applicable for PowerBroker Express.
Setting this menu option to Yes	Sets these values to Yes
Install PowerBroker Submit Host?	Install PBSSH? Install PowerBroker pbksh? Install PowerBroker pbsh? Will this host use a Log Host? Note: The pbksh and pbsh shell options are not available for PowerBroker Express. For PowerBroker Express, the submit host option installs pbssh. For the other PowerBroker for Unix & Linux products, this option installs pbrun.
Install PowerBroker Log Host?	Install PowerBroker Synchronization? PowerBroker synchronization can be initiated from this host?

Note: If you are using the package installer to install PowerBroker for Unix & Linux on a computer that already has an interactive PowerBroker for Unix & Linux installation on it, see [Interactive and Packaged Installations on the Same Computer](#) for additional considerations.

Choosing a Package Administration File

It is recommended that you use the package administration files that are provided by BeyondTrust (BTPBadmin and BTPBadmin<suffix>). These package administration files are configured to eliminate interactive prompts during package installation. If you want to use the Solaris default package administration file or other package administration file for your environment, you may be required to respond to prompts to install the packages.

Note: When installing a package using custom JumpStart, the installation process is required to be noninteractive.

Using PowerBroker for Unix & Linux Packages on Solaris Zones

The PowerBroker for Unix & Linux Solaris package installer supports Solaris Zones in Solaris release 10. The primary operating system instance is referred to as the “global zone.” All zones that are not the global zone are referred to as “non-global zones.”

Note: Solaris release 10 is required. The use of Solaris Zones is not supported on earlier releases. There are three types of zones:

- sparse root — A sparse zone is the default zone configuration and is configurable. It shares the read-only global zone’s /usr /lib /platform and /sbin partitions.
- whole root — A whole root zone does not share global zone partitions, which increases configuration flexibility.
- branded — A branded zone allows virtualization of Solaris 8, 9, or Linux and shares no partitions from the global zone. Branded zones are available as of Solaris 10 release 08/07 update 4.

Note: PowerBroker for Unix & Linux Solaris Packages will not JumpStart to non-global zones. Using Custom JumpStart to install packages on Solaris 10 Zoned systems will result in errors as the zones are not running during JumpStart execution.

Installing PowerBroker for Unix & Linux Solaris Packages on Zones is very similar to installing these packages on Solaris systems without zones. However, keep the following considerations in mind:

- PowerBroker for Unix & Linux Solaris packages are designed to be installed from the global zone. Packages will be propagated to the sparse and whole root zones upon global zone pkgadd and upon zone creation.
- PowerBroker for Unix & Linux Solaris packages are designed to be uninstalled from the global zone. Packages will be removed from sparse and whole root zones upon the global zone pkgrm.
- PowerBroker for Unix & Linux Solaris packages can be installed in the global zone only (where packages are not installed in non-global zones) with the pkgadd -G command. PowerBroker for Unix & Linux Solaris packages cannot be installed in sparse zones (with read-only partitions) and should instead be installed in the global zone. Although PowerBroker for Unix & Linux Solaris packages could be installed into a whole-root zone, PowerBroker for Unix & Linux Solaris packages are designed to be installed from the Global Zone. Packages installed on a whole-root zone are subject to overwriting by packages installed in the global zone.
- As Solaris branded zones are fully contained instances of Solaris 8 or 9, PowerBroker for Unix & Linux packages should be installed as with non-zoned Solaris instances. Loading packages to the Global Zone will not update a branded zone. PowerBroker for Unix & Linux Solaris packages for Solaris branded zones running Linux are not supported.
- The PowerBroker for Unix & Linux Solaris configuration package must be removed before removing any PowerBroker for Unix & Linux component packages and must be removed individually. PowerBroker for Unix & Linux Solaris component packages may be removed simultaneously.

Overview of Steps

Using the PowerBroker for Unix & Linux Solaris package installer involves the following steps, which are described in more detail in the next section [Installation Procedure](#):

1. Unpack the PowerBroker for Unix & Linux package tarball file.
2. Use the pbinstall program to create PowerBroker for Unix & Linux settings files.
3. Use the pbcreatesolcfgpkg program to create the PowerBroker for Unix & Linux configuration package.
4. Perform a package installation using the Solaris pkgadd command for any required components.
5. Perform a package installation using the Solaris pkgadd command for the PowerBroker for Unix & Linux configuration package.

Installation Procedure

Note: Before installing Solaris Packages, if the directories where files are installed, /usr/local, /usr/bin etc., are symbolic links to other directories, then set the environment variable PKG_NONABI_SYMLINKS to true:

```
# PKG_NONABI_SYMLINKS=true
# export PKG_NONABI_SYMLINKS
```

This will prevent the symbolic links to be removed by 'pkgadd' command on Solaris.

To install PowerBroker for Unix & Linux using the Solaris Package Manager, do the following:

1. Extract the package tarball files into the /opt/beyondtrust/ directory by executing the following command:

```
gunzip -c pb<flavor_version>_pkg.tar.Z | tar xvf -
```

2. Navigate to the directory: /opt/beyondtrust/powerbroker/<version>/<flavor>/install/

3. Execute the following command:

```
./pbinstall -z
```

You can include other options with the -z option (see [Planning Your Installation](#)). Use the -R option if you want to specify an alternate base directory for installing the component packages.

You will be asked if you want to use Client Registration. If you are planning to enable Registry Name Service, and are installing on a host that is not designated as a Primary Server, you must run Client Registration.

pbinstall will then ask if you want to enable Registry Name Service.

pbinstall displays the PowerBroker for Unix & Linux installation menu (see [pbinstall Installation Script](#), for details).

4. Make your menu selections. Note that the Enter existing pb.settings path menu option enables you to specify your own pb.settings file to use. Also, the Enter directory path for settings file creation menu option enables you to specify where to save the generated settings files. These menu options are available only when running pbinstall with the -z option.

When the menu selection process is complete, pbinstall creates the following files in the specified location:

- pb.settings
- pb.cfg
- pb.key (if encryption is enabled)
- pb.conf (for Policy Server host)

5. Optional. For a PowerBroker for Unix & Linux client, if client-server communications are to be encrypted, replace the generated pb.key file with pb.key file from the Policy Server host. Also, copy any other required key files into the same directory.

6. Optional. For a Policy Server host, write a policy file (pb.conf) and place it in the directory with the other generated files. If you do not provide a pb.conf file, a pb.conf file with the single command “reject;” will be generated and packaged.

Starting with v8.0, pbinstall -z can optionally install the default role-based policies and will ask: Installing default role-based policy pbul_policy.conf and pbul_functions.conf in <install_dir>/settings_files

Would you like to use the default role-based policy in the configuration package?

- Answer Yes for new installs only.
- If you are upgrading an existing configuration package, to avoid overwriting your existing policy, answer No.

Use the default role-based policy [Y]?

- If you answer Yes, the default pb.conf, pbul_policy.conf and pbul_functions.conf will be created and installed on the Policy Server.
- If you are installing over an existing installation, and have an existing policy in place, answer No.

7. Navigate to the directory:
/opt/beyondtrust/powerbroker/<version>/<flavor>/install/

8. Run the pbcreatesolcfgpkg utility by typing:
pbcreatesolcfgpkg -p suffix -s directory

where:

- suffix is appended to the filenames of the configuration package datastream file and the package administration file; length can be up to 26 characters (3 characters for unpatched Solaris 8).
- directory contains the PowerBroker for Unix & Linux settings and configuration files to include in the package.

The pbcreatesolcfgpkg utility creates the following files:

- Configuration package file BTPBcf<suffix>.ds
- Package administration file BTPBadmin<suffix>

For more information about the pbcreatesolcfgpkg utility, see [pbcreatesolcfgpkg](#).

9. Navigate to the directory:
/opt/beyondtrust/powerbroker/<version>/<flavor>/package/
10. Optional. To install PowerBroker for Unix & Linux in an alternative base directory, edit the provided BTPBadmin file and change the basedir=default entry as follows: basedir=target_base_directory where target_base_directory is the absolute path of the target base directory.

11. For each required component package, run the Solaris pkgadd utility to install the component package by typing:

```
pkgadd -a BTPBadmin -d pkg-datastream-file pkg-name
```

where pkg-datastream-file is the name of the component package datastream (.ds) file, and pkg-name is the name of the package. For PowerBroker for Unix & Linux packages, the package name is the same as the datastream file name without the .ds extension.

For example:

```
pkgadd -a BTPBadmin -d BTPBrunh.ds BTPBrunh
```

- Run the Solaris pkgadd utility to install the PowerBroker for Unix & Linux configuration package by typing:

```
pkgadd -a BTPBadmin<suffix> -d BTPBcf<suffix>.ds BTPBcf<suffix>
```

where <suffix> is the suffix that you specified when you created the PowerBroker for Unix & Linux configuration package in step [8](#).

- Verify the installation of the packages with the Solaris pkginfo utility by typing:

```
pkginfo | grep BTPB
```

Note: If you are installing PowerBroker for Unix & Linux using a custom JumpStart session, the PowerBroker for Unix & Linux configuration package should be added or removed only once per session to avoid installing conflicting rc scripts.

Removing PowerBroker for Unix & Linux Packages

Removing the PowerBroker for Unix & Linux packages completely uninstalls PowerBroker for Unix & Linux from a computer. To remove the PowerBroker for Unix & Linux packages, do the following:

- Navigate to the directory:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/install/
```

- Remove the PowerBroker for Unix & Linux packages by typing:

```
pkgrm -na ./BTPBadmin config-package-name component-package-1 ...
      component-package-n
```

where:

- BTPBadmin is the package administration file that is supplied by BeyondTrust. You can specify a different package administration file, or leave out the -a option to use the default package administration file. The BTPBadmin package administration file is designed to make the package installation and removal processes run noninteractively.
- config-package-name is the name of the package that you specified when you installed the configuration package. Because of the dependency relationship between the configuration package and the component packages, this package name must come first in the list.
- component-package-1 through component-package-n are the names of the packages that you specified when you installed the component packages.

Relocating the Base Directory

The Solaris package management system enables you specify an alternative base directory for installing packages. With this feature, you can specify a directory to install the PowerBroker for Unix & Linux binary files and log files in. Certain files (such as pb.settings, pb.cfg, and PowerBroker for Unix & Linux key files) must be located in the /etc directory for PowerBroker for Unix & Linux to run. These files are not relocatable. To relocate the base directory from the default / (root) directory, do the following:

- On the target machine, create the target base directory if it does not already exist.
- When you run pbinstall, use the -R option and specify the new base directory.
- Before installing the PowerBroker for Unix & Linux component packages, edit the provided BTPBadmin package administration file and change the basedir entry to refer to the new base directory. See step [10](#) in the “Installation Procedure.”
- When installing the component packages, execute pkgadd with the -a option and use the BTPBadmin package administration file. See step [11](#) in the “Installation Procedure.”

Updating PowerBroker for Unix & Linux with the Solaris Package Installer

The PowerBroker for Unix & Linux Solaris package installer can be used to update an existing PowerBroker for Unix & Linux installation to a new version. The existing PowerBroker for Unix & Linux version should have been installed with the PowerBroker for Unix & Linux package installer.

Note: It is possible to use the Solaris package installer to install PowerBroker for Unix & Linux over an existing version that was installed with `pbinstall`. However, doing so is not recommended because it can result in unused files from the existing version remaining in the file system.

Package Update Considerations

Installing a PowerBroker for Unix & Linux update with the Solaris package installer is similar to using the Solaris package installer to install PowerBroker for Unix & Linux for the first time. Keep these considerations in mind when you prepare to update PowerBroker for Unix & Linux:

- Technically, the PowerBroker for Unix & Linux Solaris packages are update packages, as opposed to upgrade packages. An update package overwrites the existing files before registering the new version number in the Solaris Package Manager database.
- A PowerBroker for Unix & Linux Solaris update package contains a complete PowerBroker for Unix & Linux installation, not just the files that have changed since the previous release.
- The PowerBroker for Unix & Linux Solaris update packages are compatible with JumpStart.
- If you have more than one PowerBroker for Unix & Linux package on a computer, you should update all packages on that computer.
- Removal and reinstallation of the configuration package is not required when updating an existing PowerBroker for Unix & Linux package installation.
- Unlike PowerBroker for Unix & Linux patches that are installed with `pbpatchinstall`, update packages cannot be rolled back to a previous release. However, you can install an older package over a newer one, effectively rolling back to the older release.

Package Update Procedure

Follow this procedure to update your installation of PowerBroker for Unix & Linux using the Solaris package installer:

1. Obtain the tarball file for the Solaris update packages that are appropriate for your hardware. The tarball file name has the format `pb<flavor>-v.v.r-b-pn_pkg.tar.Z`, where:
 - `<flavor>` indicates the operating system and hardware architecture
 - `v.v.r` is the major and minor version number and the release number
 - `b` is the build number
 - `n` is the update number
2. Extract the package tarball files into the `/opt/beyondtrust/` directory of the computer that you are updating by executing the following command:

```
gunzip -c pb<flavor_version>_pkg.tar.Z | tar xvf -
```
3. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/package/
```

4. Optional. To install PowerBroker for Unix & Linux in an alternative base directory, edit the provided BTPBadmin file and change the basedir=default entry as follows: basedir=target_base_directory where target_base_directory is the absolute path of the target base directory.
5. For each required component package, run the Solaris pkgadd utility to update the component package by typing:


```
pkgadd -a BTPBadmin -d pkg-datastream-file pkg-name
```

 where pkg-datastream-file is the name of the component package datastream (.ds) file, and pkg-name is the name of the package. For PowerBroker for Unix & Linux packages, the package name is the same as the datastream file name without the .ds extension.

For example:

```
pkgadd -a BTPBadmin -d BTPBrunh.ds BTPBrunh
```

6. Verify the installation of the packages with the Solaris pkginfo utility by typing:


```
pkginfo -x | grep BTPB
```

Sample Execution for the Solaris Package Installer

The sample execution shows the installation of a PowerBroker for Unix & Linux submit host, run host, and shared libraries using the PowerBroker for Unix & Linux Solaris package installer.

This sample execution is divided into the following parts:

- Generate the PowerBroker for Unix & Linux settings files
- Create the PowerBroker for Unix & Linux configuration package using the pbcreatesolcfgpkg program
- Install the component packages using the pkgadd command
- Install the configuration package using the pkgadd command

Generating the PowerBroker for Unix & Linux Settings Files

This section of the execution shows the generation of the PowerBroker for Unix & Linux settings files (pb.key, pb.cfg, and pb.settings) and also displays the PowerBroker for Unix & Linux installation menu. This output was generated using the pbinstall program with the options: -z, -l, and -r.

```
# ./pbinstall -z -l -r
Starting pbinstall main() from /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/.
solaris9-10.x86
WARNING:
When creating configuration packages to be installed on Solaris Zones, care
must be taken to set log file directories to Zone-writable partitions.
The default Solaris sparse zone has the following read-only and/or shared
partitions, although configuration can vary:
/usr /lib /platform /sbin
The PowerBroker for Unix & Linux log file default directory for Solaris Zones is '/var/adm'.
```

```
PowerBroker for Unix & Linux Settings File Generation
```

Please read the PowerBroker for Unix & Linux Installation Instructions before proceeding.

Checking MANIFEST against release directory

Press return to continue

The Registry Name Service of PowerBroker for Unix & Linux facilitates location of other services within the PBUL enterprise with the aid of a centralized data repository.

IMPORTANT: Client Registration is required if this is not the Primary Server and you intend to use Registry Name Services.

Do you wish to utilize Registry Name Service? [yes]? no

BeyondTrust PowerBroker for Unix & Linux Installation Menu

Opt Description [Value]

- 1 Install Everything Here (Demo Mode)? [no]
- 2 Enter existing 'pb.settings' path [none]
- 3 Enter directory path for settings file ... [/opt/acpkg/powerbroker/v9....]
- 6 Install PowerBroker Policy Server Host? [no]
- 7 Install PowerBroker Run Host? [yes]
- 8 Install PowerBroker Submit Host? [yes]
- 9 Install PowerBroker PBSSH [yes]
- 10 Install PBSUDO Policy Server? [no]
- 11 Install PowerBroker Log Host? [no]
- 15 Install File Integrity Monitoring Polic... [no]
- 17 Path to PowerBroker Password Safe 'pkru... []
- 22 Install PowerBroker GUI Host? [no]
- 23 Install PowerBroker Secure GUI Host? [no]
- 24 Install PowerBroker Utilities: pbvi, pb... [yes]
- 25 Install PowerBroker pbksh? [yes]
- 26 Install PowerBroker pbsh? [yes]
- 27 Install PowerBroker man pages? [yes]
- 28 Will this host use a Log Host? [yes]
- 29 PowerBroker Identity Services Integration? [no]
- 54 PowerBroker synchronization can be init... [no]
- 55 PowerBroker daemon location [/usr/sbin]
- 56 Number of reserved spaces for submit pr... [not-supported]
- 58 User programs location [/usr/local/bin]
- 62 User man page location [/usr/local/man/man1]
- 77 Configure target system's SuperDaemon? [yes]
- 80 Policy Server Delay [500]
- 81 Policy Server Protocol Timeout [-1]
- 88 List of PowerBroker Policy Servers to s... [dev-acayanan-05.unix.symar...]
- 89 pbrun diagnostic log? [none]
- 90 pbssh diagnostic log? [none]
- 91 Allow Local Mode? [yes]
- 92 Additional secured task checks? [no]
- 93 Suppress Policy Server host failover er... [yes]
- 94 List of PowerBroker Policy Servers to a... [dev-acayanan-05.unix.symar...]

```
95 pblockd diagnostic log [/var/adm/pblockd.log]
96 Command line options for pblockd []
97 Syslog pblockd sessions? [no]
98 Record PTY sessions in utmp/utmpx? [yes]
99 Validate Policy Server Host Connections? [no]
100 List of PowerBroker Log Hosts [dev-acayanan-05.unix.symar...]
101 Command line options for pblogd []
102 Log Host Delay [500]
103 Log Host Protocol Timeout [-1]
105 List of PowerBroker log reserved filesy... [none]
113 PowerBroker pbksh diagnostic file [/var/adm/pbksh.log]
114 PowerBroker pbsh diagnostic file [/var/adm/pbsh.log]
115 Stand-alone pblockd command [none]
116 Stand-alone root shell default iolog [/pbshell.iolog]
121 Use syslog? [yes]
122 Syslog facility to use? [LOG_AUTH]
123 Base Daemon port number [49200]
124 pbmasterd port number [49200]
125 pblockd port number [49201]
126 pblogd port number [49202]
131 Add entries to '/etc/inet/services' [yes]
132 Allow non-reserved port connections [yes]
133 Inbound Port range [1025-65535]
134 Outbound Port range [1025-65535]
137 PowerBroker network encryption options [aes-256:keyfile=/etc/pb.key]
142 PowerBroker settings file encryption type [none]
143 PowerBroker REST API encryption options [aes-256:keyfile=/etc/pb.rest.key]
144 Configure with Kerberos v5? [no]
150 Enforce High Security Encryption? [yes]
151 Use SSL? [yes]
152 SSL Configuration? [requiresl]
153 SSL pbrun Certificate Authority Directory? [none]
154 SSL pbrun Certificate Authority File? [none]
155 SSL pbrun Cipher List? [HIGH:!MD5:@STRENGTH]
156 SSL pbrun Certificate Directory? [none]
157 SSL pbrun Certificate File? [none]
158 SSL pbrun Private Key Directory? [none]
159 SSL pbrun Private Key File? [none]
160 SSL pbrun Certificate Subject Checks? [none]
161 SSL Server Certificate Authority Direct... [none]
162 SSL Server Certificate Authority File? [none]
163 SSL Server Cipher List? [HIGH:!MD5:@STRENGTH]
164 SSL Server Certificate Directory? [none]
165 SSL Server Certificate File? [/etc/pbssl.pem]
166 SSL Server Private Key Directory? [none]
167 SSL Server Private Key File? [/etc/pbssl.pem]
168 SSL Server Certificate Subject Checks? [none]
169 PowerBroker SSL Certificate Country Code [US]
```

```

170 PowerBroker SSL Certificate State/Province [AZ]
171 PowerBroker SSL Certificate Location (T... [Phoenix]
172 PowerBroker SSL Certificate Organizatio... [Security]
173 PowerBroker SSL Certificate Organization [BeyondTrust]
174 Configure PowerBroker with LDAP? [no]
175 Install BeyondTrust built-in third-part... [yes]
176 BeyondTrust built-in third-party librar... [/usr/lib/beyondtrust/pb]
188 Use PAM? [no]
194 Enable non-PAM Solaris Projects? [no]
195 Solaris Projects library file name [/usr/lib/libproject.so]
196 Allow Remote Jobs? [yes]
197 PowerBroker UNIX Domain Socket directory [none]
198 Reject Null Passwords? [no]
199 Enable TCP keepalives? [no]
200 Name Resolution Timeout [0]
N for the next menu page, P for the previous menu page, C to continue, X to exit
Please enter a menu option [For technical support call 1-800-234-9072]> c

```

```

Generating key file /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/settings_
files/pb.key...

```

```

Are all the installation settings correct [yes]?

```

```

Generating config file /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/settings_
files/pb.cfg

```

```

Creating the settings file creation script

```

```

Backed up existing settings file creation script to:

```

```

'/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/pbcreatesettingsfile.ctime.May_26_
11:01'

```

```

Running settings file creation script

```

```

Creating settings file /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/settings_
files/pb.settings

```

```

Generated settings files are in directory: /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-
18/install/settings_files

```

```

PowerBroker for Unix & Linux Settings File Generation completed successfully.

```

Creating the PowerBroker for Unix & Linux Configuration Package Using pbcreatesolcfigpkg

This section shows the creation of the PowerBroker for Unix & Linux configuration package using the pbcreatesolcfigpkg program with the -p and -s options.

Note: At the end of its output, the pbcreatesolcfigpkg script shows which PowerBroker for Unix & Linux component packages need to be installed.

```

# cd /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install
# ./pbcreatesolcfigpkg -p CLIENT1 -s /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-
18/install/settings_files
pbcreatesolcfigpkg: starting from /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install

```

```

Warning: Unpatched Solaris 8 has a 9 character package name limitation!

```

The package name created 'BTPBcfCLIENT1' is 13 characters...

```
pbcreatesolcfgpkg: keyfile pb.key will be included in package
Reading /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/settings_files/pb.cfg
## Building pkgmap from package prototype file.
## Processing pkginfo file.
## Attempting to volumize 15 entries in pkgmap.
part 1 -- 637 blocks, 24 entries
## Packaging one part.
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/pkgmap
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/pkginfo
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/root/etc/init.d/sypbcfg_svcsinetdsmf
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/root/etc/pb.cfg
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/root/etc/pb.key
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/root/etc/pb.settings
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/root/etc/rc2.d/S99sypbcfg_pbpatton
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/root/var/adm/pbksh.log
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/root/var/adm/pblocald.log
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/root/var/adm/pbsh.log
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/install/checkinstall
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/install/copyright
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/install/depend
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/install/postinstall
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/install/postremove
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/install/preinstall
/opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1/BTPBcfCLIENT1/install/preremove
## Validating control scripts.
## Packaging complete.
pbcreatesolcfgpkg: created package BTPBcfCLIENT1 in /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1
Checking uninstalled directory format package <BTPBcfCLIENT1> from </opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1>
## Checking control scripts.
## Checking package objects.
## Checking is complete.
pbcreatesolcfgpkg: pkgchk for spooled package BTPBcfCLIENT1 succeeded.
Transferring <BTPBcfCLIENT1> package instance
```

```

pbcreatesolcfigpkg: pkgtrans for package BTPBcfCLIENT1 succeeded.
Checking uninstalled stream format package <BTPBcfCLIENT1> from </opt/acpkg/powerbroker/v9.4/pbul_
solaris9-10.x86_9.4.3-18/install/BTPBcfCLIENT1.ds>
## Checking control scripts.
## Checking package objects.
## Checking is complete.
rm: Cannot remove any directory in the path of the current working directory
/var/tmp/aaaJEaG90/BTPBcfCLIENT1
pbcreatesolcfigpkg: pkgchk for datastream package BTPBcfCLIENT1 succeeded.
pbcreatesolcfigpkg: spooled package BTPBcfCLIENT1 removed.

pbcreatesolcfigpkg: package datastream file is: /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-
18/install/BTPBcfCLIENT1.ds
pbcreatesolcfigpkg: package admin file is: /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-
18/install/BTPBadminCLIENT1

pbcreatesolcfigpkg: the following packages will need to be loaded to the target system:
BTPBrunh BTPBsbmh BTPBlibs

pbcreatesolcfigpkg: completed.

```

Installing Component Packages Using the pkgadd Command

This section shows the execution of the `pkgadd` command to install component packages for the submit host, run host, and shared libraries. The execution text also includes copyright, trademark, trade secrets, and other legal text; however, those notices and text were removed from the following excerpt to save space:

```

# cd /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/package
# ls
BTPBadmin BTPBguih.ds BTPBlibs.ds BTPBlogh.ds BTPBmsth.ds BTPBrest.ds BTPBrnsh.ds BTPBrunh.ds
BTPBsbmh.ds
# pkgadd -a BTPBadmin -d BTPBlibs.ds BTPBlibs
Processing package instance <BTPBlibs> from </opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-
18/package/BTPBlibs.ds>
BeyondTrust PowerBroker Shared Libraries - Root Delegation and Privilege Management
(x86) 9.4.3-18
## Executing checkinstall script.
Using </> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying package dependencies.
## Verifying disk space requirements.
Installing BeyondTrust PowerBroker Shared Libraries - Root Delegation and Privilege Management as
<BTPBlibs>
## Executing preinstall script.
## Installing part 1 of 1.
/usr/lib/beyondtrust/pb/libcom_err.so <symbolic link>
/usr/lib/beyondtrust/pb/libcom_err.so.3 <symbolic link>
/usr/lib/beyondtrust/pb/libcom_err.so.3.0
/usr/lib/beyondtrust/pb/libcrypto.so <symbolic link>

```

```

/usr/lib/beyondtrust/pb/libcrypto.so.1 <symbolic link>
/usr/lib/beyondtrust/pb/libcrypto.so.1.0.0
/usr/lib/beyondtrust/pb/libcurl.so <symbolic link>
/usr/lib/beyondtrust/pb/libcurl.so.4 <symbolic link>
/usr/lib/beyondtrust/pb/libcurl.so.4.3.0
/usr/lib/beyondtrust/pb/libgssapi_krb5.so <symbolic link>
/usr/lib/beyondtrust/pb/libgssapi_krb5.so.2 <symbolic link>
/usr/lib/beyondtrust/pb/libgssapi_krb5.so.2.2
/usr/lib/beyondtrust/pb/libk5crypto.so <symbolic link>
/usr/lib/beyondtrust/pb/libk5crypto.so.3 <symbolic link>
/usr/lib/beyondtrust/pb/libk5crypto.so.3.1
/usr/lib/beyondtrust/pb/libkrb5.so <symbolic link>
/usr/lib/beyondtrust/pb/libkrb5.so.3 <symbolic link>
/usr/lib/beyondtrust/pb/libkrb5.so.3.3
/usr/lib/beyondtrust/pb/libkrb5support.so <symbolic link>
/usr/lib/beyondtrust/pb/libkrb5support.so.0 <symbolic link>
/usr/lib/beyondtrust/pb/libkrb5support.so.0.1
/usr/lib/beyondtrust/pb/liblber-2.4.so <symbolic link>
/usr/lib/beyondtrust/pb/liblber-2.4.so.2 <symbolic link>
/usr/lib/beyondtrust/pb/liblber-2.4.so.2.10.3
/usr/lib/beyondtrust/pb/libldap-2.4.so <symbolic link>
/usr/lib/beyondtrust/pb/libldap-2.4.so.2 <symbolic link>
/usr/lib/beyondtrust/pb/libldap-2.4.so.2.10.3
/usr/lib/beyondtrust/pb/libssl.so <symbolic link>
/usr/lib/beyondtrust/pb/libssl.so.1 <symbolic link>
/usr/lib/beyondtrust/pb/libssl.so.1.0.0
/usr/lib/beyondtrust/pb/pam_radius_auth.so <symbolic link>
/usr/lib/beyondtrust/pb/pam_radius_auth.so.1 <symbolic link>
/usr/lib/beyondtrust/pb/pam_radius_auth.so.1.3.17
[ verifying class <none> ]
## Executing postinstall script.

Checking installation of package: BTPBlibs
Installation of <BTPBlibs> was successful.
# pkgadd -a BTPBadmin -d BTPBsbmh.ds BTPBsbmh
Processing package instance <BTPBsbmh> from </opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/package/BTPBsbmh.ds>
BeyondTrust PowerBroker Submit Host - Root Delegation and Privilege Management
(x86) 9.4.3-18
## Executing checkinstall script.
Using </> as the package base directory.
## Processing package information.
## Processing system information.
1 package pathname is already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
Installing BeyondTrust PowerBroker Submit Host - Root Delegation and Privilege Management as <BTPBsbmh>
## Executing preinstall script.

```



```

## Installing part 1 of 1.
/opt/pbul/scripts/pbrnscfg.sh
/usr/lib/secure/64/libpbul_aca-elf64.so
/usr/lib/secure/libpbul_aca-elf32.so
/usr/local/bin/pbbench
/usr/local/bin/pbcall
/usr/local/bin/pbksh
/usr/local/bin/pbrun
/usr/local/bin/pbrunssh
/usr/local/bin/pbsh
/usr/local/bin/pbssh
/usr/local/man/man1/pbbench.1
/usr/local/man/man1/pbrun.1
/usr/local/man/man1/pbssh.1
/usr/local/man/man8/pbclienthost_uid.8
/usr/local/man/man8/pbcreatesolcftpkg.8
/usr/local/man/man8/pbdbutil.8
/usr/local/man/man8/pbencode.8
/usr/local/man/man8/pbinstall.8
/usr/local/man/man8/pbregister.8
/usr/local/man/man8/pbsum.8
/usr/local/man/man8/pbulpreinstall.sh.8
/usr/local/man/man8/pbversion.8
/usr/sbin/pbclienthost_uid
/usr/sbin/pbdbutil
/usr/sbin/pbencode
/usr/sbin/pbregister
/usr/sbin/pbsnapshot.sh
/usr/sbin/pbsum
/usr/sbin/pbulpreinstall.sh
/usr/sbin/pbversion
[ verifying class <none> ]
## Executing postinstall script.

Checking installation of package: BTPBsbmh
Installation of <BTPBsbmh> was successful.
# pkgadd -a BTPBadmin -d BTPBrunh.ds BTPBrunh
Processing package instance <BTPBrunh> from </opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/package/BTPBrunh.ds>
BeyondTrust PowerBroker Run Host - Root Delegation and Privilege Management
(x86) 9.4.3-18
## Executing checkinstall script.
Using </> as the package base directory.
## Processing package information.
## Processing system information.
25 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.

```

```

Installing BeyondTrust PowerBroker Run Host - Root Delegation and Privilege Management as <BTPBrunh>
## Executing preinstall script.
## Installing part 1 of 1.
/usr/local/bin/pbless
/usr/local/bin/pbmg
/usr/local/bin/pbnvi
/usr/local/bin/pbumacs
/usr/local/bin/pbvi
/usr/local/man/man1/pbless.1
/usr/local/man/man1/pbmg.1
/usr/local/man/man1/pbnvi.1
/usr/local/man/man1/pbumacs.1
/usr/local/man/man1/pbvi.1
/usr/local/man/man8/pblocald.8
/usr/sbin/pblocald
[ verifying class <none> ]
## Executing postinstall script.

Checking installation of package: BTPBrunh
Installation of <BTPBrunh> was successful.

```

Installing the Configuration Package Using the pkgadd Command

This section shows the execution of the Solaris pkgadd command to install the configuration package. Following installation of the configuration package, the installation is verified by submitting the id command to PowerBroker for Unix & Linux, and the Solaris pkginfo utility is used to list the PowerBroker for Unix & Linux packages that are installed.

The execution text also includes copyright, trademark, trade secrets, and other legal text; however, those notices and text were removed from the following excerpt to save space:

```

# cd /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_9.4.3-18/install
# pkgadd -a ./BTPBadminCLIENT1 -d BTPBcfCLIENT1.ds BTPBcfCLIENT1
Processing package instance <BTPBcfCLIENT1> from </opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.x86_
9.4.3-18/install/BTPBcfCLIENT1.ds>
BeyondTrust PowerBroker Unix/Linux Configuration - Root Delegation and Privilege Management
(noarch) 9.4.3-18
BeyondTrust PowerBroker Unix/Linux
## Executing checkinstall script.
Checking installation of dependent component packages...
## Processing package information.
## Processing system information.
6 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
Installing BeyondTrust PowerBroker Unix/Linux Configuration - Root Delegation and Privilege Management
as <BTPBcfCLIENT1>
## Executing preinstall script.
## Installing part 1 of 1.

```

```

/etc/init.d/sypbcfg_svcsetdsmf
/etc/pb.cfg
/etc/pb.key
/etc/pb.settings
/etc/rc2.d/S99sypbcfg_pbpatton
/etc/rc2.d/S99sypbcfg_svcsetdsmf <symbolic link>
/var/adm/pbksh.log
/var/adm/pblocald.log
/var/adm/pbsh.log
[ verifying class <none> ]
## Executing postinstall script.
Checking installation of package: BTPBcfCLIENT1
'pkgchk' of package BTPBcfCLIENT1 succeeded
Reading pb.cfg...
Checking installation of dependent component packages...
'pkgchk' of package BTPBlibs succeeded
'pkgchk' of package BTPBsbmh succeeded
'pkgchk' of package BTPBrunh succeeded
Looking for SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
Removing PowerBroker service definitions (if any) from /etc/inet/services.
Adding PowerBroker service definitions to /etc/inet/services.
Removing any PowerBroker definitions from SuperDaemon inetd file /etc/inet/inetd.conf
Adding PowerBroker definitions to SuperDaemon configurations /etc/inet/inetd.conf .
Reloading SuperDaemon Configurations...
Done Reloading SuperDaemon Configurations...
Updating Settings in database (if any)...

Installation of <BTPBcfCLIENT1> was successful.
# pkginfo | grep BTPB
application BTPBcfCLIENT1 BeyondTrust PowerBroker Unix/Linux Configuration - Root Delegation and
Privilege Management
application BTPBlibs BeyondTrust PowerBroker Shared Libraries - Root Delegation and Privilege Management
application BTPBrunh BeyondTrust PowerBroker Run Host - Root Delegation and Privilege Management
application BTPBsbmh BeyondTrust PowerBroker Submit Host - Root Delegation and Privilege Management

```

Sample of the Uninstall Process from a Package Installation

This section shows the execution of the Solaris `pkgrm` utility to remove the PowerBroker for Unix & Linux packages.

```

# cd /opt/acpkg/powerbroker/v9.4/pbul_solaris9-10.sparc_9.4.3-06/install
# pkgrm -na ./BTPBadminCLIENT1 BTPBcfCLIENT1 BTPBsbmh BTPBrunh BTPBlibs
Reading pb.cfg...
Looking for SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
Removing PowerBroker service definitions (if any) from /etc/inet/services.
Removing any PowerBroker definitions from SuperDaemon inetd file /etc/inet/inetd.conf
Reloading SuperDaemon Configurations...

```

```
Done Reloading SuperDaemon Configurations...
Removal of <BTPBcfCLIENT1> was successful.
Removal of <BTPBsbmh> was successful.
Removal of <BTPBrunh> was successful.
Removal of <BTPBlibs> was successful.
```

Linux Package Installer

This section describes how to install PowerBroker for Unix & Linux using a package installer for Red Hat Enterprise Linux (RHEL) 4 or 5 on an x86, x86_64, ia64, or S/390 computer. Use the Linux package installation if you want to do either of the following:

- Install PowerBroker for Unix & Linux using the Linux RPM package manager.
- Install PowerBroker Express using the Linux RPM package manager.

If you have a previous installation of PowerBroker for Unix & Linux, you must first uninstall it before installing PowerBroker Express. Likewise, if you have an installation of PowerBroker Express, you must uninstall it before installing PowerBroker for Unix & Linux.

The PowerBroker for Unix & Linux Linux package installer that is described here is not compatible with the BeyondTrust PowerBroker V5.x packages. You must remove BeyondTrust PowerBroker Packages V5.x before installing PowerBroker for Unix & Linux Linux packages.

Prerequisites

To use the Linux package installer, you must have the following:

- Package tarball file for the appropriate PowerBroker for Unix & Linux flavor
 - Note:** For the PowerBroker for Unix & Linux Linux package installer, the tarball files are cumulative. That is, an update tarball file contains a complete PowerBroker for Unix & Linux installation. It is not necessary to install a baseline version of PowerBroker for Unix & Linux before installing an upgrade.
- Root access or superuser privileges
- RPM Package Manager (rpm) v4.4 or later

Note: The PowerBroker for Unix & Linux Linux package installer does not support prefix/suffix installations.

Planning Your Installation

When preparing to use the PowerBroker for Unix & Linux package installer, you should be familiar with the following concepts and restrictions:

Component packages - A PowerBroker for Unix & Linux component package is an RPM package manager (.rpm) file that installs a part of the PowerBroker for Unix & Linux application. There are seven PowerBroker for Unix & Linux component packages. In the following file names, the format is:

```
powerbroker-component-v.v.r.bb-pv.arch.rpm
```

where:

`component` = PowerBroker component package name.

`v` = major version `v` = minor version `r` = release

`bb` = build

`pv` = version number of the package

`arch` = architecture (for example, i386)

- `powerbroker-master-v.v.r.bb-pv.arch.rpm` - Contains Policy Server host, pbsync, and pbsyncd.
- `powerbroker-submithost-v.v.r.bb-pv.arch.rpm` - Contains submit host and PowerBroker for Unix & Linux shells.
- `powerbroker-runhost-v.v.r.bb-pv.arch.rpm` - Contains run host and PowerBroker for Unix & Linux utilities.
- `powerbroker-loghost-v.v.r.bb-pv.arch.rpm` - Contains log host, pbsync, and pbsyncd.
- `powerbroker-guihost-v.v.r.bb-pv.arch.rpm` - Contains GUI host and secure GUI host.
- `powerbroker-shlibs-v.v.r.bb-pv.arch.rpm` - Contains shared libraries.
- `powerbroker-express-v.v.r.bb-pv.arch.rpm` - Contains pbssh. This package is only used for PowerBroker Express to install pbssh instead of pbrun. This package is not compatible with PowerBroker Server `powerbroker-submithost-v.v.r.bb-pv.arch.rpm` and `powerbroker-runhost-v.v.r.bb-pv.arch.rpm` packages.
- `powerbroker-pbrest-v.v.r.bb-pv.arch.rpm` - Contains REST API files.
- `powerbroker-rnssvr-v.v.r.bb-pv.arch.rpm` - Contains Registry Name Service files.
- `powerbroker-licsvr-v.v.r.bb-pv.arch.rpm` - Contains License Server files.

Which component packages are required depends on the type of PowerBroker for Unix & Linux host you are creating, such as Policy Server host, submit host, and so on. You can select the types of PowerBroker for Unix & Linux hosts in the `pbinstall` installation menu, as shown in the following table. For readability the ending of each component in the table (`-v.v.r.bb-pv.arch.rpm`) is removed.

Note: For PowerBroker Express, there will not be an option for a run host. For PowerBroker Express, the submit host option installs pbssh. using the `powerbroker-express` component package. For the other PowerBroker for Unix & Linux products, this option installs pbrun.

Table 5. Required Components

Menu Selection	Required Components
Install everything here (demo mode)? = Yes	<code>powerbroker-master</code> <code>powerbroker-runhost</code> <code>powerbroker-submithost</code> <code>powerbroker-loghost</code> <code>powerbroker-guihost</code> <code>powerbroker-shlibs</code>
Install PowerBroker Master Host? = Yes	<code>powerbroker-master</code>
Install PowerBroker Run Host? = Yes	<code>powerbroker-runhost</code> PowerBroker Express does not use the run host package.
Install PowerBroker Submit Host? = Yes	<code>powerbroker-submithost</code> For PowerBroker Express, the <code>powerbroker-express</code> package is installed.
Install PowerBroker Log Host? = Yes	<code>powerbroker-loghost</code>

Install PowerBroker GUI Host? = Yes	powerbroker-guihost
Install PowerBroker Secure GUI Host? = Yes	powerbroker-guihost
Install BeyondTrust built-in third-party libraries? = Yes	powerbroker-shlibs
Install Registry Name Services Server? [yes]	powerbroker-rnssvr
Install License Server? [yes]	powerbroker-licsvr

Configuration package - RPM package that is used to install the following files:

- pb.settings
- pb.cfg
- pb.key (if applicable)
- pb.conf (for Policy Server hosts)
- Man pages for the pbinstall and pbcreatelincfgpkg programs

The PowerBroker for Unix & Linux configuration package is created by the `pbcreatelincfgpkg` program. The component packages must be installed before you install the configuration package.

Package name - Name of the package as stored in the RPM package manager database. For PowerBroker for Unix & Linux package installations, this name is the same as the package file name without the `.arch.rpm` extension.

Relocated base directory - The directory where the PowerBroker for Unix & Linux binary files and log files are installed. You can choose an alternative directory in which to install these files. For more information, see [Relocating the Base Directory](#).

pbinstall program - To create the PowerBroker for Unix & Linux settings files, you use the `pbinstall` program with the `-z` ("settings only") option. `pbinstall -z` only creates the settings files, and is incompatible with the following command line options:

Table 6. Options Incompatible with pbinstall -z

Option	Description
-b	Runs pbinstall in batch mode.
-c	Skip the steps that process or update the PowerBroker for Unix & Linux settings file.
-e	Runs install script automatically by bypassing the menu step of pbinstall.
-i	Ignores previous pb.settings and pb.cfg files.
-p	Sets the pb installation prefix.
-s	Sets the pb installation suffix.
-u	Install the utility programs.
-x	Creates a log synchronization host (that is, installs pbsyncd).

When you execute `pbinstall` with the `-z` option, you can see two menu items that are not otherwise available:

- Enter existing `pb.settings` path

Enables you to specify your own `pb.settings` file. `pbinstall` reads this settings file and populates the remaining menu choices. You can override some menu choices.

If set to `none`, then `pbinstall` does not read a settings file. The remaining menu choices are populated with default values.

- Enter directory path for settings file creation

Enables you to specify an alternative output directory for the settings files. The default directory is:

```
/unzip-dir/powerbroker/v10.0.1/<flavor>install/settings_files
```

where `unzip-dir` is the directory where the package tarball file was unzipped.

The behavior of `pbinstall -z` depends on whether certain additional command line options are specified:

- If no other command line options are specified, `pbinstall` initially presents a short version of the installation menu. Depending on the choices you make in these items, further menu items become available.
- If command line options `-g`, `-l`, `-m`, `-o`, `-r`, or `-w` are specified, `pbinstall` presents an expanded version of the installation menu that reflects the host types that you are configuring.

For complete information about the `pbinstall` command-line options, see [pbinstall](#).

When running `pbinstall` with the `-z` option, the following menu items are pre-programmed and cannot be changed:

- Install man pages?
- PowerBroker daemon location
- Administration programs location
- User programs location
- GUI library location
- Policy include (sub) file directory
- User man page location
- Admin man page location
- Policy filename
- BeyondTrust built-in third-party library directory

In addition, the values of the following menu items determine the values of other menu items:

Table 7. Options Preset When Running `pbinstall -z`

Setting this menu option to Yes	Sets these values to Yes
Install PowerBroker Master Host?	Install PowerBroker Synchronization? PowerBroker synchronization can be initiated from this host?
Install PowerBroker Run Host?	Install PowerBroker Utilities? Note: This is not applicable for PowerBroker Express.

Install PowerBroker Submit Host?	Install PBSSH? Install PowerBroker pbksh? Install PowerBroker pbsh? Will this host use a Log Host? Note: The pbksh and pbsh shell options are not available for PowerBroker Express. For PowerBroker Express, the submit host option installs pbssh. For the other PowerBroker for Unix & Linux products, this option installs pbrun.
Install PowerBroker Log Host?	Install PowerBroker Synchronization? PowerBroker synchronization can be initiated from this host?

Note: If you are using the package installer to install PowerBroker for Unix & Linux on a computer that already has an interactive PowerBroker for Unix & Linux installation on it, see [Interactive and Packaged Installations on the Same Computer](#), for additional considerations.

If you are planning to use Registry Name Service and you are running `pbinstall -z` on a client host (non-primary server), you must perform Client Registration. This is necessary to properly set up the registry name service database. Client Registration will also require that you collect from the PowerBroker for Unix & Linux Primary Server the following information: REST Application ID, REST Application Key, Primary Server network name or IP address, Primary Policy Server REST TCP/IP port, and Registration Client Profile name.

Overview of Steps

Using the Linux package installer involves the following steps (described in more detail in the next section, [Installation Procedure](#)):

1. Unpack the PowerBroker for Unix & Linux package tarball file.
2. Use the `pbinstall` program to create PowerBroker for Unix & Linux settings files.
3. Use the `pbcreatelincfgpkg` program to create the PowerBroker for Unix & Linux configuration package.
4. Perform a package installation using the Linux `rpm` command for any required components.
5. Perform a package installation using the Linux `rpm` command for the PowerBroker for Unix & Linux configuration package.

Installation Procedure

To install PowerBroker for Unix & Linux using the RPM package manager, do the following:

1. Extract the package tarball files into the `/opt/beyondtrust/` directory by executing the following command:


```
tar xvfz pbul<flavor_version>_pkg.tar.Z
```
2. Optional. The PowerBroker for Unix & Linux Linux package files are digitally signed. You can verify that the packages are genuine by doing the following:
 - a. Go to the BeyondTrust web site (www.beyondtrust.com) and click **Support** to display the PowerBroker for Unix & Linux Downloads page.
 - b. In the Customers section, click **Login**. Use your customer user name and password to log in to the PowerBroker for Unix & Linux Downloads page.

- c. Click **Digital Signature file for Linux RPM packages** and download the tar file to the Linux computer.
- d. Extract the key from the tar file.
- e. Import the key to the RPM database with the following command:


```
rpm --import keyfile
```

 where `keyfile` is the file name of the key file.

- f. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/package/
```

- g. Execute the following command:

```
rpm -K *.rpm
```

For each package, you should see output similar to the following:

```
powerbroker-master-6.2.0.11-1.i386.rpm: (sha1) dsa sha1 md5 gpg OK
```

The `OK` at the end of the line indicates that the package is genuine.

3. Navigate to the directory: `/opt/beyondtrust/powerbroker/<version>/<flavor>/install/`

4. Execute the following command:

```
./pbinstall -z
```

You can include other options with the `-z` option (see [Planning Your Installation](#)). Use the `-R` option to specify an alternate base directory for installing the component packages.

`pbinstall` displays the PowerBroker for Unix & Linux installation menu (see [pbinstall Installation Script](#), for details).

You will be asked if you want to use Client Registration. If you are planning to enable Registry Name Service, and are installing on a host that is not designated as a Primary Server, you must run Client Registration.

`pbinstall` will then ask if you want to enable Registry Name Service.

5. Make your menu selections. Note that the `Enter existing pb.settings path menu option` enables you to specify your own `pb.settings` file to use. Also, the `Enter directory path for settings file creation menu option` enables you to specify where to save the generated settings files. These menu options are available only when running `pbinstall` with the `-z` option.

When the menu selection process is complete, `pbinstall` creates the following files in the specified location:

- `pb.settings`
- `pb.cfg`
- `pb.key` (if encryption is enabled)
- `pb.conf` (for Policy Server host)

6. Optional. For a PowerBroker for Unix & Linux client, if client-server communications are to be encrypted, replace the generated `pb.key` file with `pb.key` file from the Policy Server host. Also, copy any other required key files into the same directory.

This step is automatically done if you choose to use Client Registration.

7. Optional. For a Policy Server host, write a policy file (`pb.conf`) and place it in the directory with the other generated files. If you do not provide a `pb.conf` file, a `pb.conf` file with the single command "reject;" will be generated and packaged.

Starting with v8.0, `pbinstall -z` can optionally install the default role-based policies and will ask: Installing default role-based policy `pbul_policy.conf` and `pbul_functions.conf` in `<install_dir>/settings_files`

Would you like to use the default role-based policy in the configuration package?

- Answer Yes for new installs only.
- If you are upgrading an existing configuration package, to avoid overwriting your existing policy, answer No.

Use the default role-based policy [Y]?

- If you answer Yes, the default `pb.conf`, `pbul_policy.conf` and `pbul_functions.conf` will be created and installed on the Policy Server.
- If you are installing over an existing installation, and have an existing policy in place, answer No.

8. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/install/
```

9. Run the `pbcreatelincfgpkg` utility by typing:

```
pbcreatelincfgpkg -p suffix -s directory
```

where:

- `suffix` is appended to the configuration package name; length can be up to 18 characters.
- `directory` contains the PowerBroker for Unix & Linux settings and configuration files to include in the package.

The `pbcreatelincfgpkg` utility creates the PowerBroker for Unix & Linux configuration package file, `powerbroker-config<suffix>-sv-pv.arch.rpm`

For more information about the `pbcreatelincfgpkg` utility, see [pbcreatelincfgpkg](#).

10. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/package/
```

11. For each required component package, run the Linux `rpm` utility to install the component package by typing:

```
rpm -iv package-file
```

where `package-file` is the name of the component package (`.rpm`) file. For example:

```
rpm -iv powerbroker-submithost-9.4.1.03-1.x86_64.rpm
```

Note: To install all component packages, type the following command:

```
rpm -iv *.rpm
```

12. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/install/
```

13. Run the Linux `rpm` utility to install the PowerBroker for Unix & Linux configuration package by typing:

```
rpm -iv package-file
```

where `package-file` is the name of the configuration package (`.rpm`) file that was created in step 9.

14. Verify the installation of the packages by typing:

```
rpm -qa | grep powerbroker
```

Removing PowerBroker for Unix & Linux Packages

Removing the PowerBroker for Unix & Linux packages completely uninstalls PowerBroker for Unix & Linux from a computer.

To remove the PowerBroker for Unix & Linux packages, type the following:

```
rpm -e config-package-name
    component-package-1 ... component-package-n
```

where

- *config-package-name* is the name of the package that you specified when you installed the configuration package. This package name is not required to come first in the list; `rpm` removes it first. (However, if you remove packages with separate `rpm` processes, you must remove the configuration package first.)
- *component-package-1* through *component-package-n* are the names of the packages that you specified when you installed the component packages.

For example:

```
rpm -e powerbroker-configPBUL941-9.4.1.03-1.x86_64 powerbroker-submithost-
    9.4.1.03-1.x86_64
```

Relocating the Base Directory

Using the RPM package management system you can set an alternative base directory for installing packages. With this feature, you can specify a directory to install the PowerBroker for Unix & Linux binary files and log files in. Certain files (such as `pb.settings`, `pb.cfg`, and PowerBroker for Unix & Linux key files) must be located in the `/etc` directory for PowerBroker for Unix & Linux to run. These files are not relocatable.

To relocate the base directory from the default `/` (root) directory, do the following:

1. On the target machine, create the target base directory if it does not already exist.
2. When you run `pbinstall`, use the `-R` option and specify the new base directory.
3. When installing the component packages, execute `rpm` with the `--prefix` option and specify the relocated directory. For example:

```
rpm -ivh --prefix /local/powerbroker powerbroker-runhost-9.4.1.03-1.x86_
    64.rpm
```

See step [11](#) in the “Installation Procedure.”

Note: The files that are installed by the configuration package cannot be relocated. Do not use the `--prefix` option when installing the configuration package.

Upgrading PowerBroker for Unix & Linux with the Linux Package Installer

The PowerBroker for Unix & Linux Linux package installer can be used to upgrade an existing PowerBroker for Unix & Linux installation to a new version. The existing PowerBroker for Unix & Linux version should have been installed with the PowerBroker for Unix & Linux package installer.

If you have a previous installation of PowerBroker for Unix & Linux, you must first uninstall it before installing PowerBroker Express. Likewise, if you have an installation of PowerBroker Express, you must uninstall it before installing PowerBroker for Unix & Linux.

Note: It is possible to use the Linux package installer to install PowerBroker for Unix & Linux over an existing version that was installed with `pbinstall`. However, doing so is not recommended because it can result in unused files from the existing version remaining in the file system.

Package Upgrade Considerations

Installing an upgrade with the Linux package installer is similar to using the Linux package installer to install PowerBroker for Unix & Linux for the first time. Keep these considerations in mind when you prepare to upgrade:

- Technically, the PowerBroker for Unix & Linux Linux packages are upgrade packages, as opposed to update packages. An upgrade package installs the new files before removing the existing files and registering the new version number in the RPM database.
- A PowerBroker for Unix & Linux Linux upgrade package contains a complete PowerBroker for Unix & Linux installation, not just the files that have changed since the previous release.
- If you have more than one PowerBroker for Unix & Linux package on a computer, upgrade all packages on that computer.
- Removal and reinstallation of the configuration package is not required when updating an existing PowerBroker for Unix & Linux package installation.
- Unlike PowerBroker for Unix & Linux patches that are installed with `pbpatchinstall`, upgrade packages cannot be rolled back to a previous release. However, you can install an older package over a newer one, effectively rolling back to the older release. For more information, see [Reverting to a Previous Version](#).

Package Upgrade Procedure

Follow this procedure to upgrade your installation of PowerBroker for Unix & Linux using the Linux package installer:

1. Obtain the tarball file for the Linux upgrade packages that are appropriate for your hardware. The tarball file name has the format `pbul<flavor>-v.v.r-bb-pn_pkg.tar.Z`, where:
 - `<flavor>` indicates the operating system and hardware architecture
 - `v.v.r` is the major and minor version number and the release number
 - `bb` is the build number
 - `n` is the update number
2. Extract the package tarball files into the `/opt/beyondtrust/` directory by executing the following command:


```
tar xvfz pbul<flavor_version>_pkg.tar.Z
```
3. Navigate to the directory:


```
/opt/beyondtrust/powerbroker/v10.0.1/<flavor>/package/
```
4. Use the Linux `rpm` utility to upgrade the component packages by typing:


```
rpm -Uv package-file-1 package-file-2...
```

 where `package-file-n` is the name of a component package (`.rpm`) file.

For example:

```
rpm -Uv powerbroker-submithost-9.4.1.03-1.p2-1.x86_64.rpm powerbroker-runhost-9.4.1.03-1.p2-1.x86_64.rpm
```

Note: If there are multiple component packages on a computer and you want to upgrade each package with a separate rpm command, use the `--replacepkgs` option:

```
rpm -Uv powerbroker-submithost-9.4.1.03-1.p2-1.x86_64.rpm --replacepkgs
```

5. Verify the installation of the packages by typing:

```
rpm -qa | grep powerbroker
```

Reverting to a Previous Version

Unlike PowerBroker for Unix & Linux patches that are installed with `pbpatchinstall`, upgrade packages cannot be rolled back to a previous release. However, you can install an older package over a newer one, effectively rolling back to the older release. To install older packages over newer ones, use the following command:

```
rpm -Uv --oldpackage package-file-1 package file-2...
```

This command restores the previous release. Repeat the command to restore earlier releases. To restore a single package per `rpm` command, add the `--replacepkgs` option.

Sample Execution for the Linux Package Installer

The sample execution shows the installation of a PowerBroker for Unix & Linux submit host, run host, and shared libraries using the PowerBroker for Unix & Linux Linux package installer.

This sample execution is divided into the following parts:

- Generate the PowerBroker for Unix & Linux settings files
- Create the PowerBroker for Unix & Linux configuration package using the `pbcreatelinconfpkg` program
- Install the component packages using the `rpm` command
- Install the configuration package using the `rpm` command

Generating the PowerBroker for Unix & Linux Settings Files

This section of the execution shows the generation of the PowerBroker for Unix & Linux settings files (`pb.key`, `pb.cfg`, and `pb.settings`) and also displays the PowerBroker for Unix & Linux installation menu. This output was generated using the `pbinstall` program with the options: `-z`, `-l`, and `-r`.

```
# ./pbinstall -zlr
Starting pbinstall main() from /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/.
linux.x86-64
```

```
PowerBroker for Unix & Linux Settings File Generation
```

```
Please read the PowerBroker for Unix & Linux Installation Instructions before proceeding.
```

```
Checking MANIFEST against release directory
```

```
Press return to continue
```

```
The Registry Name Service of PowerBroker for Unix & Linux facilitates location
of other services within the PBUL enterprise with the aid of a centralized
data repository.
```

IMPORTANT: Client Registration is required if this is not the Primary Server and you intend to use Registry Name Services.

Do you wish to utilize Registry Name Service? [yes]? no

BeyondTrust PowerBroker for Unix & Linux Installation Menu

Opt Description [Value]

```

1 Install Everything Here (Demo Mode)? [no]
2 Enter existing 'pb.settings' path [none]
3 Enter directory path for settings file ... [/opt/final/powerbroker/v9....]
6 Install PowerBroker Policy Server Host? [no]
7 Install PowerBroker Run Host? [yes]
8 Install PowerBroker Submit Host? [yes]
9 Install PowerBroker PBSSH [yes]
10 Install PBSUDO Policy Server? [no]
11 Install PowerBroker Log Host? [no]
15 Install File Integrity Monitoring Polic... [no]
17 Path to PowerBroker Password Safe 'pkru... []
22 Install PowerBroker GUI Host? [no]
23 Install PowerBroker Secure GUI Host? [no]
24 Install PowerBroker Utilities: pbvi, pb... [yes]
25 Install PowerBroker pbksh? [yes]
26 Install PowerBroker pbsh? [yes]
27 Install PowerBroker man pages? [yes]
28 Will this host use a Log Host? [yes]
29 PowerBroker Identity Services Integration? [no]
54 PowerBroker synchronization can be init... [no]
55 PowerBroker daemon location [/usr/sbin]
56 Number of reserved spaces for submit pr... [80]
58 User programs location [/usr/local/bin]
61 User man page location [/usr/local/man/man1]
76 Configure target system's SuperDaemon? [yes]
79 Policy Server Delay [500]
80 Policy Server Protocol Timeout [-1]
88 List of PowerBroker Policy Servers to s... [mypolicyserver.unix.symar...]
89 pbrun diagnostic log? [none]
90 pbssh diagnostic log? [none]
91 Allow Local Mode? [yes]
92 Additional secured task checks? [no]
93 Suppress Policy Server host failover er... [yes]
94 List of PowerBroker Policy Servers to a... [mypolicyserver.unix.symar...]
95 pblocald diagnostic log [/var/log/pblocald.log]
96 Command line options for pblocald []
97 Syslog pblocald sessions? [no]
98 Record PTY sessions in utmp/utmpx? [yes]
99 Validate Policy Server Host Connections? [no]
100 List of PowerBroker Log Hosts [myloghost.unix.symar...]
101 Command line options for pblogd []
102 Log Host Delay [500]
103 Log Host Protocol Timeout [-1]

```

```
105 List of PowerBroker log reserved filesy... [none]
112 Add installed shells to /etc/shells [no]
113 PowerBroker pbksh diagnostic file [/var/log/pbksh.log]
114 PowerBroker pbsh diagnostic file [/var/log/pbsh.log]
115 Stand-alone pblockd command [none]
116 Stand-alone root shell default iolog [/pbshell.iolog]
121 Use syslog? [yes]
122 Syslog facility to use? [LOG_AUTHPRIV]
123 Base Daemon port number [24345]
124 pbmasterd port number [24345]
125 pblockd port number [24346]
126 pblogd port number [24347]
131 Add entries to '/etc/services' [yes]
132 Allow non-reserved port connections [yes]
133 Inbound Port range [1025-65535]
134 Outbound Port range [1025-65535]
137 PowerBroker network encryption options [aes-256:keyfile=/etc/pb.key]
142 PowerBroker settings file encryption type [none]
143 PowerBroker REST API encryption options [aes-256:keyfile=/etc/pb.key]
144 Configure with Kerberos v5? [no]
150 Enforce High Security Encryption? [yes]
151 Use SSL? [yes]
152 SSL Configuration? [requiresssl]
153 SSL pbrun Certificate Authority Directory? [none]
154 SSL pbrun Certificate Authority File? [none]
155 SSL pbrun Cipher List? [HIGH:!MD5:@STRENGTH]
156 SSL pbrun Certificate Directory? [none]
157 SSL pbrun Certificate File? [none]
158 SSL pbrun Private Key Directory? [none]
159 SSL pbrun Private Key File? [none]
160 SSL pbrun Certificate Subject Checks? [none]
161 SSL Server Certificate Authority Direct... [none]
162 SSL Server Certificate Authority File? [none]
163 SSL Server Cipher List? [HIGH:!MD5:@STRENGTH]
164 SSL Server Certificate Directory? [none]
165 SSL Server Certificate File? [/etc/pbssl.pem]
166 SSL Server Private Key Directory? [none]
167 SSL Server Private Key File? [/etc/pbssl.pem]
168 SSL Server Certificate Subject Checks? [none]
169 PowerBroker SSL Certificate Country Code [US]
170 PowerBroker SSL Certificate State/Province [AZ]
171 PowerBroker SSL Certificate Location (T... [Phoenix]
172 PowerBroker SSL Certificate Organizatio... [Security]
173 PowerBroker SSL Certificate Organization [BeyondTrust]
174 Configure PowerBroker with LDAP? [no]
175 Install BeyondTrust built-in third-part... [yes]
176 BeyondTrust built-in third-party librar... [/usr/lib/beyondtrust/pb]
188 Use PAM? [no]
```



```

196 Allow Remote Jobs? [yes]
197 PowerBroker UNIX Domain Socket directory [none]
198 Reject Null Passwords? [no]
199 Enable TCP keepalives? [no]
200 Name Resolution Timeout [0]
N for the next menu page, P for the previous menu page, C to continue, X to exit
Please enter a menu option [For technical support call 1-800-234-9072]> c

Generating key file /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/settings_
files/pb.key...

Are all the installation settings correct [yes]?
Generating config file /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/settings_
files/pb.cfg
Creating the settings file creation script
Backed up existing settings file creation script to:
'/opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/pbcreatesettingsfile.ctime.Feb_13_16:28'
Running settings file creation script
Creating settings file /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/settings_
files/pb.settings
Generated settings files are in directory: /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-
03/install/settings_files
PowerBroker for Unix & Linux Settings File Generation completed successfully.

```

Creating the PowerBroker for Unix & Linux Configuration Package Using pbcreatelincfgpkg

This section shows the creation of the PowerBroker for Unix & Linux configuration package using the pbcreatelincfgpkg program with the `-p` and `-s` options.

Note: At the end of its output, the pbcreatelincfgpkg script shows which PowerBroker for Unix & Linux component packages need to be installed.

```

# ./pbcreatelincfgpkg -p CLIENTPAKU -s /opt/final/powerbroker/v9.4/CLIENTPAKU_settings_files
pbcreatelincfgpkg: starting from /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install
pbcreatelincfgpkg: keyfile pb.key will be included in package
Reading /opt/final/powerbroker/v9.4/CLIENTPAKU_settings_files/pb.cfg

pbcreatelincfgpkg: making PowerBroker Linux configuration package . . .
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.kq2x6j
+ umask 022
+ cd /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILD
+ LANG=C
+ export LANG
+ unset DISPLAY
+ rm -rf '/opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILD/*'
+ exit 0
Executing(%build): /bin/sh -e /var/tmp/rpm-tmp.Z2J5QI

```

```

+ umask 022
+ cd /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILD
+ LANG=C
+ export LANG
+ unset DISPLAY
+ exit 0
Executing(%install): /bin/sh -e /var/tmp/rpm-tmp.wlumC7
+ umask 022
+ cd /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILD
+ '[' /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64 '!=' / ']'
+ rm -rf /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64
++ dirname /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64
+ mkdir -p /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT
+ mkdir /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64
+ LANG=C
+ export LANG
+ unset DISPLAY
+ mkdir -p /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/etc
+ mkdir -p /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/etc/pb
+ cp /opt/final/powerbroker/v9.4/CLIENTPAKU_settings_files/pb.settings /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/etc/pb.settings
+ cp /opt/final/powerbroker/v9.4/CLIENTPAKU_settings_files/pb.cfg /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/etc/pb.cfg
+ cp /opt/final/powerbroker/v9.4/CLIENTPAKU_settings_files/pb.key /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/etc/pb.key
++ dirname /var/log/pblocald.log
+ logfiledir=/var/log
+ '[' '!' -d /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/var/log ']'
+ mkdir -p /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/var/log
++ dirname /var/log/pbksh.log
+ logfiledir=/var/log
+ '[' '!' -d /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/var/log ']'
++ dirname /var/log/pbsh.log
+ logfiledir=/var/log
+ '[' '!' -d /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/var/log ']'
++ dirname /pbshell.iolog
+ logfiledir=/
+ '[' '!' -d /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/ ']'
+ /usr/lib/rpm/check-buildroot
+ /usr/lib/rpm/redhat/brp-compress

```

```

+ /usr/lib/rpm/redhat/brp-strip /usr/bin/strip
+ /usr/lib/rpm/redhat/brp-strip-static-archive /usr/bin/strip
+ /usr/lib/rpm/redhat/brp-strip-comment-note /usr/bin/strip /usr/bin/objdump
+ /usr/lib/rpm/brp-python-bytecompile /usr/bin/python
+ /usr/lib/rpm/redhat/brp-python-hardlink
+ /usr/lib/rpm/redhat/brp-java-repack-jars
Processing files: powerbroker-configCLIENTPAKU-9.4.1.03-1.noarch
Requires(interp): /bin/sh /bin/sh /bin/sh /bin/sh
Requires(rpmlib): rpmlib(CompressedFileNames) <= 3.0.4-1 rpmlib(FileDigests) <= 4.6.0-1 rpmlib(PayloadFilesHavePrefix) <= 4.0-1
Requires(pre): /bin/sh
Requires(post): /bin/sh
Requires(preun): /bin/sh
Requires(postun): /bin/sh
Checking for unpackaged file(s): /usr/lib/rpm/check-files /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64
Wrote: /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/RPMS/noarch/powerbroker-configCLIENTPAKU-9.4.1.03-1.noarch.rpm
Executing(%clean): /bin/sh -e /var/tmp/rpm-tmp.A8w0eY
+ umask 022
+ cd /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILD
+ rm -rf /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/etc /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/pbshell.iolog /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install/rpmbuild/BUILDROOT/powerbroker-9.4.1.03-1.x86_64/var
+ exit 0
pbcreatelincfgpkg: rpm package built
pbcreatelincfgpkg: rpm package verified
pbcreatelincfgpkg: rpm package 'powerbroker-configCLIENTPAKU-9.4.1.03-1.noarch.rpm' placed in /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install

pbcreatelincfgpkg: the following packages will need to be loaded to the target system:
powerbroker-runhost powerbroker-submitthost powerbroker-shlibs

pbcreatelincfgpkg: completed.

```

Installing Component Packages Using the rpm Command

This section shows the execution of the rpm command to install component packages for the submit host, run host, and shared libraries.

```

# cd /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/package
# rpm -iv powerbroker-shlibs-9.4.1.03-1.x86_64.rpm powerbroker-submitthost-9.4.1.03-1.x86_64.rpm powerbroker-runhost-9.4.1.03-1.x86_64.rpm
warning: powerbroker-shlibs-9.4.1.03-1.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 19227ca5: NOKEY
Preparing packages for installation...
powerbroker-shlibs-9.4.1.03-1
powerbroker-runhost-9.4.1.03-1

```

```
powerbroker-submithost-9.4.1.03-1
```

Installing the Configuration Package Using the rpm Command

This section shows the execution of the Linux rpm command to install the configuration package. Following installation of the configuration package, the installation is verified by submitting the id command to PowerBroker for Unix & Linux, and the Linux rpm -qa utility is used to list the PowerBroker for Unix & Linux packages that are installed.

```
# cd /opt/final/powerbroker/v9.4/pbul_linux.x86-64_9.4.1-03/install
# rpm -iv powerbroker-configCLIENTPAKU-9.4.1.03-1.noarch.rpm
Preparing packages for installation...
powerbroker-configCLIENTPAKU-9.4.1.03-1
Reading pb.cfg...
Updating Settings in database (if any)...
Checking installation of dependent component packages...
'rpm -V' of package powerbroker-shlibs succeeded
'rpm -V' of package powerbroker-submithost succeeded
'rpm -V' of package powerbroker-runhost succeeded
Looking for SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
Removing PowerBroker service definitions (if any) from /etc/services.
Adding PowerBroker service definitions to /etc/services.
Removing any PowerBroker definitions from SuperDaemon xinetd file /etc/xinetd.conf
Adding PowerBroker definitions to SuperDaemon configurations /etc/xinetd.conf.
Reloading SuperDaemon Configurations...
Done Reloading SuperDaemon Configurations...
# rpm -qa | grep powerbroker
powerbroker-runhost-9.4.1.03-1.x86_64
powerbroker-configCLIENTPAKU-9.4.1.03-1.noarch
powerbroker-shlibs-9.4.1.03-1.x86_64
powerbroker-submithost-9.4.1.03-1.x86_64

# pbrun id # test PowerBroker
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),501(amanda)

# rpm -qa | grep powerbroker # list PowerBroker packages
powerbroker-runhost-9.4.1.03-1.x86_64
powerbroker-configCLIENTPAKU-9.4.1.03-1.noarch
powerbroker-shlibs-9.4.1.03-1.x86_64
powerbroker-submithost-9.4.1.03-1.x86_64
```

Sample of the Uninstall Process from a Package Installation

This section shows the execution of the Linux rpm utility to remove the PowerBroker for Unix & Linux packages.

```
# rpm -e powerbroker-configCLIENTPAKU powerbroker-shlibs powerbroker-submithost powerbroker-runhost
Reading pb.cfg...
Looking for SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
```

```
Removing PowerBroker service definitions (if any) from /etc/services. Removing any PowerBroker
definitions from SuperDaemon xinetd file
/etc/xinetd.conf
Reloading SuperDaemon Configurations...
Done Reloading SuperDaemon Configurations...
```

AIX Package Installer

This section describes how to install PowerBroker for Unix & Linux using a package installer for AIX V5.3, 6.1 and 7.0 on a POWER 32-bit or POWER 64-bit computer. AIX package installers are compatible with or without WPARs. Use the AIX package installer if you want to do either of the following:

- Install PowerBroker for Unix & Linux using the AIX `installp` command.
- Install PowerBroker Express using the AIX `installp` command.

If you have a previous installation of PowerBroker for Unix & Linux, you must first uninstall it before installing PowerBroker Express. Likewise, if you have an installation of PowerBroker Express, you must uninstall it before installing PowerBroker for Unix & Linux.

The PowerBroker for Unix & Linux AIX package installer that is described here is not compatible with the BeyondTrust PowerBroker V5.x packages. If the BeyondTrust PowerBroker Packages V5.x packages are installed, you must remove them before installing the PowerBroker for Unix & Linux AIX packages.

WPARs

If you have AIX V6.1 or higher, then you can use Workload Partitions (WPARs). For more information about WPARs and propagating BeyondTrust AIX package installations to them, see [Installing PowerBroker for Unix and Linux onto WPARs](#), [Using syncwpar to Propagate Additional Packages to Shared WPARs](#), [Viewing a List of WPARs](#).

Prerequisites

To use the AIX package installer, you must have the following:

- Package tarball file for the appropriate PowerBroker for Unix & Linux flavor
- Root access or superuser privileges

Note: The PowerBroker for Unix & Linux AIX package installer does not support prefix/suffix installations.

Planning Your Installation

When preparing to use the PowerBroker for Unix & Linux package installer, you should be familiar with the following concepts and restrictions:

Component packages - A PowerBroker for Unix & Linux component package is a AIX backup file format (.bff) file that installs a portion of the PowerBroker for Unix & Linux application. PowerBroker for Unix & Linux component packages use the following format:

```
powerbroker.component-v.v.r.bb.bff
```

where:

v = major version

v = minor version

r = release

bb = build

For example:

```
powerbroker.masterhost-6.2.0.05.bff
```

The component package names and files are described in the following list:

- `powerbroker.masterhost-v.v.r.bb.bff` - Contains the Policy Server `host`, `pbcheck`, `pbkey`, `pbmasterd`, `pbpasswd`, `pbpatton`, `pbprint`, and man pages. `powerbroker.common-v.v.r.bb.bff` is a prerequisite for this package.
- `powerbroker.runhost-v.v.r.bb.bff` - Contains the run host and PowerBroker for Unix & Linux utilities: `pblocald`, `pbless`, `pbmg`, `pbnvi`, `pbumacs`, `pbvi`, and man pages. `powerbroker.common-v.v.r.bb.bff` is a prerequisite for this package.

Note: This component package is not available for PowerBroker Express.
- `powerbroker.submithost-v.v.r.bb.bff` - Contains the submit host and PowerBroker for Unix & Linux shells, `pbksh`, `pbsh`, `pbssh`, `pbrun`, and man pages. `powerbroker.common-v.v.r.bb.bff` is a prerequisite for this package.

Note: This component package is not available for PowerBroker Express.
- `powerbroker.loghost-v.v.r.bb.bff` - Contains the log host, `pblogd`, and man pages. `powerbroker.common-v.v.r.bb.bff` is a prerequisite for this package.
- `powerbroker.guihost-v.v.r.bb.bff` - Contains the GUI host and secure GUI host, `pbguid`, `pbreport`, `pbuvrpg`, man pages, html files, and conf files. `powerbroker.common-v.v.r.bb.bff` is a prerequisite for this package.
- `powerbroker.sharedlibs-v.v.r.bb.bff` - Contains the shared libraries: `libcom_err.a.3.0`, `libcrypto.so.0.9.8`, `libgssapi_krb5.a.2.2`, `libk5crypto.a.3.0`, `libkrb5.a.3.2`, `liblber-2.3.a`, `libldap-2.3.a`, `libssl.so.0.9.8`. `powerbroker.common-v.v.r.bb.bff` is a prerequisite for this package.
- `powerbroker.common-v.v.r.bb.bff` - Contains the shared files and `pbbench`, `pbcall`, `bencode`, `pbsum`, man pages and `pbinstall.8`, and `pbcreateaixcfgpkg.8`. This package is a prerequisite for all the previously listed packages: `powerbroker.masterhost`, `powerbroker.submithost`, `powerbroker.loghost`, `powerbroker.guihost`, `powerbroker.sharedlibs`.
- `powerbroker.mlcommon-v.v.r.bb.bff` - Contains the Policy Server log shared files, `pblog`, `pbmerge`, `pbreplay`, `pbsyncd`, `pbsync`, and man pages. This package is a prerequisite for `powerbroker.masterhost-v.v.r.bb.bff` and `powerbroker.loghost-v.v.r.bb.bff`.
- `powerbroker.express-v.v.r.bb.bff` - Contains `pbssh`. This package is used only for PowerBroker Express to install `pbssh` instead of `pbrun`. This package is not compatible with PowerBroker Server `powerbroker.submithost-v.v.r.bb.bff` and `powerbroker.runhost-v.v.r.bb.bff` packages.
- `powerbroker-pbrest-v.v.r.bb-pv.arch.rpm` - Contains REST API files.
- `powerbroker.rnssvr-v.v.r.bb.bff` - Contains Registry Name Service files.
- `powerbroker.licsvr-v.v.r.bb.bff` - Contains License Server files.

Which component packages are required depends on the type of PowerBroker for Unix & Linux host you are creating, such as Policy Server host, log host, and so on. You can select the types of hosts in the `pbinstall` installation menu, as shown in the following table.

Note: For PowerBroker Express, there will not be an option for a run host. For PowerBroker Express, the submit host option installs `pbssh`. using the `powerbroker.express` component package. For the other PowerBroker for Unix & Linux products, this option installs `pbrun`.

Table 8. Required Components

Menu Selection	Required Components
Install everything here (demo mode)? = Yes	powerbroker.masterhost-v.v.r.bb.bff powerbroker.runhost-v.v.r.bb.bff powerbroker.submithost-v.v.r.bb.bff powerbroker.loghost-v.v.r.bb.bff powerbroker.guihost-v.v.r.bb.bff powerbroker.sharedlibs-v.v.r.bb.bff powerbroker.common-v.v.r.bb.bff powerbroker.mlcommon-v.v.r.bb.bff
Install PowerBroker Policy Server Host? = Yes	powerbroker.masterhost-v.v.r.bb.bff powerbroker.common-v.v.r.bb.bff powerbroker.mlcommon-v.v.r.bb.bff
Install PowerBroker Run Host? = Yes	powerbroker.runhost-v.v.r.bb.bff powerbroker.common-v.v.r.bb.bff PowerBroker Express does not use the run host package.
Install PowerBroker Submit Host? = Yes	powerbroker.submithost-v.v.r.bb.bff powerbroker.common-v.v.r.bb.bff For PowerBroker Express, the powerbroker.express-v.v.r.bb.bff package is installed.
Install PowerBroker Log Host? = Yes	powerbroker.loghost-v.v.r.bb.bff powerbroker.common-v.v.r.bb.bff powerbroker.mlcommon-v.v.r.bb.bff
Install PowerBroker GUI Host? = Yes	powerbroker.guihost-v.v.r.bb.bff powerbroker.common-v.v.r.bb.bff
Install PowerBroker Secure GUI Host? = Yes	powerbroker.guihost-v.v.r.bb.bff powerbroker.common-v.v.r.bb.bff
Install BeyondTrust built-in third-party libraries? = Yes	powerbroker.sharedlibs-v.v.r.bb.bff powerbroker.common-v.v.r.bb.bff
Install Registry Name Services Server? [yes]	powerbroker.rnssvr-v.v.r.bb.bff
Install License Server? [yes]	powerbroker.licsvr-v.v.r.bb.bff

Configuration package - AIX installation package created by the user named `powerbroker.config[suffix]` where suffix is user-defined. It contains the configuration files that are used to install the following files:

- `pb.settings`
- `pb.cfg`
- `pb.key` (if applicable)

- `pb.conf` (for Policy Server hosts)
- Man pages for the `pbinstall` and `pbcreateaixcfgpkg` programs

The PowerBroker for Unix & Linux configuration package is created by the `pbcreateaixcfgpkg` program. The component packages must be installed before you install the configuration package.

Package name - Name of the installation package stored in the AIX database. For PowerBroker for Unix & Linux package installations, this name is the same as the package file name without the `.bff` extension.

pbinstall program - To create the PowerBroker for Unix & Linux settings files, you use the `pbinstall` program with the `-z` ("settings only") option. `pbinstall -z` only creates the settings files and is incompatible with the following command line options:

Table 9. Options Incompatible with `pbinstall -z`

Option	Description
<code>-b</code>	Runs <code>pbinstall</code> in batch mode.
<code>-c</code>	Skip the steps that process or update the PowerBroker for Unix & Linux settings file.
<code>-e</code>	Runs install script automatically by bypassing the menu step of <code>pbinstall</code> .
<code>-i</code>	Ignores previous <code>pb.settings</code> and <code>pb.cfg</code> files.
<code>-p</code>	Sets the <code>pb</code> installation prefix.
<code>-s</code>	Sets the <code>pb</code> installation suffix.
<code>-u</code>	Install the utility programs.
<code>-x</code>	Creates a log synchronization host (that is, installs <code>pbsyncd</code>).

When you execute `pbinstall` with the `-z` option, you can see two menu items that are not otherwise available:

- Enter existing `pb.settings` path
 Enables you to specify your own `pb.settings` file. `pbinstall` reads this settings file and populates the remaining menu choices. You can override some menu choices.
 If set to none, then `pbinstall` does not read a settings file. The remaining menu choices are populated with default values.
- Enter directory path for settings file creation
 Enables you to specify an alternative output directory for the settings files. The default directory is:
`/unzip-dir/powerbroker/<version>/<flavor>/install/settings_files`
 where `unzip-dir` is the directory where the package tarball file was unzipped.

The behavior of `pbinstall -z` depends on whether certain additional command line options are specified:

- If no other command line options are specified, `pbinstall` initially presents a short version of the installation menu (items 1–8 only). Depending on the choices you make in these items, further menu items become available.

- If command line options -g, -l, -m, or -r are specified, `pbinstall` presents an expanded version of the installation menu that reflects the host types that you are configuring.

For complete information about the `pbinstall` command-line options, see [pbinstall](#).

When running `pbinstall` with the `-z` option, the following menu items are pre-programmed and cannot be changed:

```
Install man pages? PowerBroker daemon location
Administration programs location User programs location
GUI library location
Policy include (sub) file directory User man page location
Admin man page location Policy filename
BeyondTrust built-in third-party library directory
```

In addition, the values of the following menu items determine the values of other menu items:

Table 10. Options Preset When Running `pbinstall -z`

Setting this menu option to Yes	Sets these values to Yes
Install PowerBroker Policy Server Host?	Install PowerBroker Synchronization? PowerBroker synchronization can be initiated from this host?
Install PowerBroker Run Host?	Install PowerBroker Utilities? Note: This is not applicable for PowerBroker Express.
Install PowerBroker Submit Host?	Install PBSSH? Install PowerBroker pbksh? Install PowerBroker pbsh? Will this host use a Log Host? Note: The pbksh and pbsh shell options are not available for PowerBroker Express. For PowerBroker Express, the submit host option installs <code>pbssh</code> . For the other PowerBroker for Unix & Linux products, this option installs <code>pbrun</code> .
Install PowerBroker Log Host?	Install PowerBroker Synchronization? PowerBroker synchronization can be initiated from this host?

Note: If you are using the package installer to install PowerBroker for Unix & Linux on a computer that already has an interactive PowerBroker for Unix & Linux installation on it, see [Interactive and Packaged Installations on the Same Computer](#), for additional considerations.

Using PowerBroker for Unix & Linux Packages on AIX WPARs

The PowerBroker for Unix & Linux AIX package installer supports AIX WPARs in AIX V6.1 and higher. The primary operating system instance is referred to as the global WPARs. All WPARs that are not the global are referred to as non-global WPARs.

Note: AIX release V6.1 or higher is required. The use of WPARs is not supported on earlier releases. There are two types of WPARs:

- Shared WPARs share some of the global environment's file systems and are administered by the global environment.
- Non-shared WPARs share none of the global environment's file systems and are treated as stand-alone systems.

Installing PowerBroker for Unix & Linux AIX packages on WPARs is very similar to installing these packages on AIX systems without WPARs. For instructions, see [Installing PowerBroker for Unix & Linux onto WPARs](#).

Overview of Steps

Using the PowerBroker for Unix & Linux AIX package installer involves the following steps (described in more detail in the next section [Installation Procedure](#)):

1. Unpack the PowerBroker for Unix & Linux package tarball file.
2. Use the `pbinstall` program to create PowerBroker for Unix & Linux settings files.

Note: If the `powerbroker.express` package is present, then the `powerbroker.express` package is installed. Otherwise, the PowerBroker for Unix & Linux packages are installed.
3. Use the `pbcreateaixcfgpkg` program to create the PowerBroker for Unix & Linux configuration package.
4. Perform a package installation using the AIX `installp` command for any required components.
5. Perform a package installation using the AIX `installp` command for the PowerBroker for Unix & Linux configuration package.

Installation Procedure

To install PowerBroker for Unix & Linux in the AIX global environment, do the following:

1. Extract the package tarball files into the `/opt/beyondtrust/` directory by executing the following command:


```
gunzip -c pb<flavor_version>_pkg.tar.Z | tar xvf -
```
2. Navigate to the directory: `/opt/beyondtrust/powerbroker/<version>/<flavor>/install/`
3. Execute the following command:


```
./pbinstall -z
```

You can include other options with the `-z` option (see [Planning Your Installation](#)).

You will be asked if you want to use Client Registration. If you are planning to enable Registry Name Service, and are installing on a host that is not designated as a Primary Server, you must run Client Registration.

`pbinstall` will then ask if you want to enable Registry Name Service.

`pbinstall` displays the PowerBroker for Unix & Linux installation menu (see [pbinstall Installation Script](#), for details).

4. Make your menu selections. Note that the Enter existing `pb.settings` path menu option enables you to specify your own `pb.settings` file to use. Also, the Enter directory path for settings file creation menu option enables you to specify where to save the generated settings files. These menu options are available only when running `pbinstall` with the `-z` option.

When the menu selection process is complete, `pbinstall` creates the following files in the specified location:

- `pb.settings`
- `pb.cfg`

- `pb.key` (if encryption is enabled)
 - `pb.conf` (for Policy Server host)
5. Optional. For a PowerBroker for Unix & Linux client, if client-server communications are to be encrypted, replace the generated `pb.key` file with `pb.key` file from the Policy Server host. Also, copy any other required key files into the same directory.
 6. Optional. For a Policy Server host, write a policy file (`pb.conf`) and place it in the directory with the other generated files. If you do not provide a `pb.conf` file, a `pb.conf` file with the single command `reject;` will be generated and packaged.

Starting with v8.0, `pbinstall -z` can optionally install the default role-based policies and will ask: Installing default role-based policy `pbul_policy.conf` and `pbul_functions.conf` in `<install_dir>/settings_files`

Would you like to use the default role-based policy in the configuration package?

- Answer Yes for new installs only.
- If you are upgrading an existing configuration package, to avoid overwriting your existing policy, answer No.

Use the default role-based policy [Y]?

- If you answer Yes, the default `pb.conf`, `pbul_policy.conf` and `pbul_functions.conf` will be created and installed on the Policy Server.
- If you are installing over an existing installing, and have an existing policy in place, answer No.

7. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/install/
```

8. Run the `pbcreateaixcfgpkg` utility by typing:

```
pbcreateaixcfgpkg -p suffix -s directory
```

where:

- `suffix` is appended to the filenames of the configuration package backup file format file and the package administration file; length can be up to 26 characters.
- `directory` contains the PowerBroker for Unix & Linux settings and configuration files to include in the package.

The `pbcreateaixcfgpkg` utility creates the configuration package file `powerbroker.config<suffix>-v.v.r.b.bff`.

For more information about the `pbcreateaixcfgpkg` utility, see [pbcreateaixcfgpkg](#).

9. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/<version>/<flavor>/package/
```

10. For each required component package, run the AIX `installp` command to install one component package by typing:

```
installp -agd ./ powerbroker.pkg-name
```

where `pkg-name` is the name of the component package file.

For example:

```
installp -agd ./ powerbroker.submithost
```

where using the `-g` option installs all the prerequisite packages along with the `powerbroker.submithost` package. In this case, `powerbroker.common` is a prerequisite package for the `powerbroker.submit` package.

Alternately you can install all the component packages by typing:

```
installp -agd ./ powerbroker
```

11. Run the AIX `installp` command to install the PowerBroker for Unix & Linux configuration package by typing:

```
installp -ad ./ powerbroker.config<suffix>
```

where `<suffix>` is the suffix that you set when you created the PowerBroker for Unix & Linux configuration package in step 8.

12. Verify the installation of the packages with the AIX `lspp` command by typing:

```
lspp -l | grep powerbroker
```

Installing PowerBroker for Unix & Linux onto WPARs

The process for installing PowerBroker AIX packages onto non-shared WPARs is similar to the process for installing in the global AIX environment because the installed software is private to the non-shared WPAR. Therefore, there is no need for synchronization. For instructions, see [Installation Procedure](#).

To install PowerBroker for Unix & Linux packages onto shared WPARs, do the following:

1. Follow the procedures in [Installation Procedure](#) to create the AIX packages.
2. Install PowerBroker component (`usr`) packages in the global AIX environment. The `usr` packages are visible to the WPARs.
3. Install PowerBroker configuration (`root`) package in the global AIX environment. The `root` packages are not visible to the WPARs until propagated.
4. To make the PowerBroker configuration (`root`) package visible to the WPARs, use the `syncwpar` command and follow the instructions on propagating packages to WPARs in [Using syncwpar to Propagate Additional Packages to Shared WPARs](#).
5. Optional. List the WPARs. See [Viewing a List of WPARs](#).

Removing PowerBroker for Unix & Linux Packages

Removing the PowerBroker for Unix & Linux packages completely uninstalls PowerBroker for Unix & Linux from a computer. To remove the packages, do the following:

1. Navigate to the directory:


```
/opt/beyondtrust/powerbroker/<version>/aix/install/
```
2. Remove multiple PowerBroker for Unix & Linux packages by typing:


```
installp -u powerbroker.configClient component-package-1 ... component-package-n
```

where:

- `configClient` is the name of the package that you specified when you installed the configuration package. Because of the dependency relationship between the configuration package and the component packages, this package name must come first in the list.
- `component-package-1` through `component-package-n` are the names of the packages that you specified when you installed the component packages, such as `powerbroker.submithost`.

For example:

```
installp -u powerbroker.configClient powerbroker.submithost
powerbroker.loghost
```

or

Remove a package and its prerequisites by using the `installp -gu` command. For example the following command removes the `powerbroker.runhost` package and its prerequisite package `powerbroker.common`:

```
installp -gu powerbroker.runhost
```

Removing PowerBroker AIX Package from Shared WPARs

To remove PowerBroker for Unix & Linux packages from shared WPARs, do the following:

1. Remove the PowerBroker for Unix & Linux packages from the global AIX environment using the following command:

```
installp -u powerbroker
```

All PowerBroker for Unix & Linux `usr` packages and the global `root` package are removed.

2. Remove the PowerBroker for Unix & Linux `root` packages from WPARs by doing either of the following:
 - To remove the PowerBroker for Unix & Linux root package from one or more specified WPAR, type the following command from the global AIX environment:

```
syncwpar [nodeA] [nodeB] ... [nodeX]
```

where `nodeA`, `nodeB`, ... `nodeX` are the names of the WPARs.

- To remove the PowerBroker for Unix & Linux root package from all WPARs, type the following command from the global AIX environment:

```
syncwpar -A
```

When you use the `-A` option, all PowerBroker root packages are removed from WPAR.

Note: The `syncwpar` command synchronizes all packages between the AIX global environment and shared WPARs.

For an example of using the `syncwpar` command, see [Example of Using syncwpar to Propagate Package Removal From Shared WPARs](#).

3. Optional. Verify that the packages are removed from the WPARs by following the instructions in [Verifying Removal of PowerBroker for Unix & Linux Packages](#).

Updating PowerBroker for Unix & Linux with Update Packages

The PowerBroker for Unix & Linux AIX package installer can be used to update an existing PowerBroker for Unix & Linux installation to a new version. The existing PowerBroker for Unix & Linux version should have been installed using the PowerBroker for Unix & Linux package installer.

If you have a previous installation of PowerBroker for Unix & Linux, you must first uninstall it before installing PowerBroker Express. Likewise, if you have an installation of PowerBroker Express, you must uninstall it before installing PowerBroker for Unix & Linux.

Update Package Considerations

Installing a PowerBroker for Unix & Linux update package is similar to using the AIX package installer to install PowerBroker for Unix & Linux for the first time. Keep these considerations in mind when you prepare to upgrade PowerBroker for Unix & Linux:

- Each release of PowerBroker for Unix & Linux AIX update packages is “sparse,” that is, it contains only the updated files. Therefore, a full PowerBroker for Unix & Linux package installation (of the same major and minor version) must be performed before you can install an upgrade package. For example, before you can install update package version 9.2.1, you must have the full PowerBroker for Unix & Linux package version 9.2.0 installed.
- Each successive PowerBroker AIX update package is cumulative; for example, update package version 9.4.1 contains all of the updates in update package version 9.4.0.
- Removal and reinstallation of the configuration fileset is not required when updating an existing PowerBroker for Unix & Linux installation.
- Update packages that have not been committed can be rejected. You cannot reject update packages that have been committed.
- Committing a given update package requires prior or concurrent commit of earlier update packages.
- The PowerBroker for Unix & Linux configuration package does not contain any executable files and therefore does not need to be upgraded. However, if you are creating a new configuration package, you should create it with the same version of PowerBroker for Unix & Linux as the component packages you are installing.

Update Package Procedure

Follow this procedure to update your installation of PowerBroker for Unix & Linux using the update packages:

1. Obtain the tarball file for the AIX update packages that are appropriate for your hardware. The tarball file name has the format `pb<flavor>-v.v.r-bb-update_pkg.tar.Z`, where:
 - `<flavor>` indicates the operating system and hardware architecture
 - `v.v.r` is the major and minor version number and the release number
 - `bb` is the build number
2. Extract the package files into the `/opt/beyondtrust/` directory by executing the following command:


```
gunzip -c pb<flavor_version>-update_pkg.tar.Z | tar xvf -
```
3. Navigate to the directory:


```
/opt/beyondtrust/powerbroker/version/flavor/package/
```
4. Run the AIX `installp` utility to install the PowerBroker for Unix & Linux component package or packages by typing:


```
installp -ad ./ powerbroker.package_name [v.v.r.bb] [powerbroker.package_name [v.v.r.bb] ... ]
```

 where:
 - `package_name` is the name of the PowerBroker for Unix & Linux package to be installed
 - `v.v.r.bb` (optional) is the version, release, and build number, for example, 9.4.1.03
5. Commit the update package by typing:


```
installp -c powerbroker [v.v.r.bb]
```

where *v.v.r.bb* (optional) is the version, release, and build number, for example, 9.4.1.03.

- Verify the installation of the filesets with the AIX `lspp` utility by typing:

```
lspp -al powerbroker.package_name
```

where *package_name* is the name of the PowerBroker for Unix & Linux package that you installed.

Rejecting an Update Package

You can reject an update package that has been applied but not committed by typing

```
installp -r powerbroker.package_name [v.v.r.bb]
```

where

- *package_name* is the name of the PowerBroker for Unix & Linux package that you want to reject
- *v.v.r.bb* (optional) is the version, release, and build number, for example, 6.2.1.11 After an update package has been committed, you can not reject it.

Update Packages and WPARs

Installing update packages on WPARs involves the same considerations as installing a baseline PowerBroker for Unix & Linux package on WPARs. For more information, see [Installing PowerBroker for Unix & Linux onto WPARs](#).

Sample Execution for the AIX Package Installer

The sample execution shows the installation of a PowerBroker for Unix & Linux submit host, run host, and shared libraries using the PowerBroker for Unix & Linux AIX package installer.

This sample execution is divided into the following parts:

- Generate the PowerBroker for Unix & Linux settings files.
- Create the PowerBroker for Unix & Linux configuration package using the `pbcreateaixcfgpkg` program.
- Install the component packages using the `installp -ad` command.
- Install the configuration package using the `installp -ad` command.
- Use `syncwpar` to propagate additional AIX global environment packages to shared WPARs. WPARs are available with AIX V6.1 and higher.

Generating the PowerBroker for Unix & Linux Settings Files

This section of the execution shows the generation of the PowerBroker for Unix & Linux settings files (`pb.key`, `pb.cfg`, and `pb.settings`) and also displays the PowerBroker for Unix & Linux installation menu. This output was generated using the `pbinstall` program with the `-z` option.

```
# ./pbinstall -zlr
Starting pbinstall main() from /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/.
aix52+
WARNING:
When creating configuration packages to be installed on AIX WPARs, care
must be taken to set log file directories to WPAR-writable partitions.
The default AIX shared WPAR has the following read-only and/or shared
partitions, although configuration can vary:
```



```
/usr /opt /proc
```

The PowerBroker for Unix & Linux log file default directory for AIX WPARs is '/var/adm'.

PowerBroker for Unix & Linux Settings File Generation

Please read the PowerBroker for Unix & Linux Installation Instructions before proceeding.

Checking MANIFEST against release directory

Press return to continue

The Registry Name Service of PowerBroker for Unix & Linux facilitates location of other services within the PBUL enterprise with the aid of a centralized data repository.

IMPORTANT: Client Registration is required if this is not the Primary Server and you intend to use Registry Name Services.

Do you wish to utilize Registry Name Service? [yes]? no

BeyondTrust PowerBroker for Unix & Linux Installation Menu

Opt Description [Value]

- 1 Install Everything Here (Demo Mode)? [no]
- 2 Enter existing 'pb.settings' path [none]
- 3 Enter directory path for settings file ... [/opt/bt_pkg/powerbroker/v9...]
- 6 Install PowerBroker Policy Server Host? [no]
- 7 Install PowerBroker Run Host? [yes]
- 8 Install PowerBroker Submit Host? [yes]
- 9 Install PowerBroker PBSSH [yes]
- 10 Install PBSUDO Policy Server? [no]
- 11 Install PowerBroker Log Host? [no]
- 15 Install File Integrity Monitoring Polic... [no]
- 17 Path to PowerBroker Password Safe 'pkru... []
- 22 Install PowerBroker GUI Host? [no]
- 23 Install PowerBroker Secure GUI Host? [no]
- 24 Install PowerBroker Utilities: pbvi, pb... [yes]
- 25 Install PowerBroker pbksh? [yes]
- 26 Install PowerBroker pbsh? [yes]
- 27 Install PowerBroker man pages? [yes]
- 28 Will this host use a Log Host? [yes]
- 29 PowerBroker Identity Services Integration? [no]
- 54 PowerBroker synchronization can be init... [no]
- 55 PowerBroker daemon location [/usr/sbin]
- 56 Number of reserved spaces for submit pr... [80]
- 58 User programs location [/usr/local/bin]
- 62 User man page location [/usr/share/man/man1]
- 77 Configure target system's SuperDaemon? [yes]
- 80 Policy Server Delay [500]
- 81 Policy Server Protocol Timeout [-1]
- 88 List of PowerBroker Policy Servers to s... []
- 89 pbrun diagnostic log? [none]
- 90 pbssh diagnostic log? [none]

```
91 Allow Local Mode? [yes]
92 Additional secured task checks? [no]
93 Suppress Policy Server host failover er... [yes]
94 List of PowerBroker Policy Servers to a... []
95 pblockd diagnostic log [/var/adm/pblockd.log]
96 Command line options for pblockd []
97 Syslog pblockd sessions? [no]
98 Record PTY sessions in utmp/utmpx? [yes]
99 Validate Policy Server Host Connections? [no]
100 List of PowerBroker Log Hosts []
101 Command line options for pblogd []
102 Log Host Delay [500]
103 Log Host Protocol Timeout [-1]
105 List of PowerBroker log reserved filesy... [none]
112 Add installed shells to /etc/shells [no]
113 PowerBroker pbksh diagnostic file [/var/adm/pbksh.log]
114 PowerBroker pbsh diagnostic file [/var/adm/pbsh.log]
115 Stand-alone pblockd command [none]
116 Stand-alone root shell default iolog [/pbshell.iolog]
121 Use syslog? [yes]
122 Syslog facility to use? [LOG_AUTH]
123 Base Daemon port number [24345]
124 pbmasterd port number [24345]
125 pblockd port number [24346]
126 pblogd port number [24347]
131 Add entries to '/etc/services' [yes]
132 Allow non-reserved port connections [yes]
133 Inbound Port range [1025-65535]
134 Outbound Port range [1025-65535]
137 PowerBroker network encryption options [aes-256:keyfile=/etc/pb.key]
142 PowerBroker settings file encryption type [none]
143 PowerBroker REST API encryption options [aes-256:keyfile=/etc/pb.key]
144 Configure with Kerberos v5? [no]
150 Enforce High Security Encryption? [yes]
151 Use SSL? [yes]
152 SSL Configuration? [requiressl]
153 SSL pbrun Certificate Authority Directory? [none]
154 SSL pbrun Certificate Authority File? [none]
155 SSL pbrun Cipher List? [HIGH:!MD5:@STRENGTH]
156 SSL pbrun Certificate Directory? [none]
157 SSL pbrun Certificate File? [none]
158 SSL pbrun Private Key Directory? [none]
159 SSL pbrun Private Key File? [none]
160 SSL pbrun Certificate Subject Checks? [none]
161 SSL Server Certificate Authority Direct... [none]
162 SSL Server Certificate Authority File? [none]
163 SSL Server Cipher List? [HIGH:!MD5:@STRENGTH]
164 SSL Server Certificate Directory? [none]
```

```

165 SSL Server Certificate File? [/etc/pbssl.pem]
166 SSL Server Private Key Directory? [none]
167 SSL Server Private Key File? [/etc/pbssl.pem]
168 SSL Server Certificate Subject Checks? [none]
169 PowerBroker SSL Certificate Country Code [US]
170 PowerBroker SSL Certificate State/Province [AZ]
171 PowerBroker SSL Certificate Location (T... [Phoenix]
172 PowerBroker SSL Certificate Organizatio... [Security]
173 PowerBroker SSL Certificate Organization [BeyondTrust]
174 Configure PowerBroker with LDAP? [no]
175 Install BeyondTrust built-in third-part... [yes]
176 BeyondTrust built-in third-party librar... [/usr/lib/beyondtrust/pb]
188 Use PAM? [no]
196 Allow Remote Jobs? [yes]
197 PowerBroker UNIX Domain Socket directory [none]
198 Reject Null Passwords? [no]
199 Enable TCP keepalives? [no]
200 Name Resolution Timeout [0]
N for the next menu page, P for the previous menu page, C to continue, X to exit
Please enter a menu option [For technical support call 1-800-234-9072]> c

```

```

no such map in server's domain
No submitmasters was specified and no NIS netgroup called pbsubmitmasters found
PowerBroker for Unix & Linux needs to know the submitmasters(s) to work.
The PowerBroker for Unix & Linux programs need to know which Policy Server Host(s) you have
decided to allow to act as submitmaster(s) for this machine.
Submitmasters take requests for secured tasks from Submit Hosts,
accept or reject them, and pass the accepted requests to a Run Host.
To locate submitmasters, programs look for a setting in the settings file
containing the names of the submitmaster machines or a netgroup
called pbsubmitmasters.

```

```

Enter Policy Server list (submitmasters): aix52-ca012-05.unix.symark.com
no such map in server's domain
No acceptmasters was specified and no NIS netgroup called pbacceptmasters found
PowerBroker for Unix & Linux needs to know the acceptmasters(s) to work.

```

```

The PowerBroker for Unix & Linux programs need to know which Policy Server Host(s) you have
decided to allow to request execution of secured tasks to this machine.
Hosts on the acceptmasters list are the Policy Server Hosts which are allowed
to make secured task requests to this machine.

```

```

To do this, programs look for a setting in the settings file containing the
names of the acceptmasters machines or a netgroup called pbacceptmasters.

```

```

Enter Incoming Policy Server list (acceptmasters): aix52-ca012-05.unix.symark.com
no such map in server's domain
No log hosts was specified and no NIS netgroup called pblogservers found

```

PowerBroker for Unix & Linux needs to know the log hosts(s) to work.

The PowerBroker for Unix & Linux programs need to know which machine(s) you have selected as Log Host(s). Log Hosts are hosts which Policy Servers select for Run Hosts to do event and io logging.

To do this, pbmasterd looks for the setting logservers in the settings file. This setting contains the names of the Log Host machines or a netgroup.

Current installation settings for Log Server(s):

Enter Log Server list (logservers): aix52-ca012-05.unix.symark.com

Generating key file /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/settings_files/pb.key...

Are all the installation settings correct [yes]?

Generating config file /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/settings_files/pb.cfg

Creating the settings file creation script

Running settings file creation script

Creating settings file /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/settings_files/pb.settings

Generated settings files are in directory: /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/settings_files

PowerBroker for Unix & Linux Settings File Generation completed successfully.

Creating the PowerBroker for Unix & Linux Configuration Package Using pbcreateaixcfgpkg

This section shows the creation of the PowerBroker for Unix & Linux configuration package using the pbcreateaixcfgpkg program with the -p and -s options. This is done after copying the PowerBroker for Unix & Linux Policy Server keyfile to the client's settings_files directory by doing the following:

```
cp /tmp/pb.key ./settings_files/pb.key
```

Note: At the end of its output, the pbcreateaixcfgpkg script shows which PowerBroker for Unix & Linux component packages need to be installed.

```
# ./pbcreateaixcfgpkg -p CLIENT1 -s /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/settings_files
pbcreateaixcfgpkg: starting from /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install
pbcreateaixcfgpkg: keyfile pb.key will be included in package
pbcreateaixcfgpkg: Reading /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/settings_files/pb.cfg
pbcreateaixcfgpkg: processing, please wait . . .
pbcreateaixcfgpkg: archiving root package control library...
ar: Creating an archive file /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/lppbuild/usr/lpp/powerbroker.configCLIENT1/inst_root/liblpp.a.
q - powerbroker.configCLIENT1.al
q - powerbroker.configCLIENT1.config
```

```

q - powerbroker.configCLIENT1.inventory
q - powerbroker.configCLIENT1.post_i
q - powerbroker.configCLIENT1.pre_i
q - powerbroker.configCLIENT1.size
q - powerbroker.configCLIENT1.unpost_i
q - powerbroker.configCLIENT1.unpre_i
ar: Sequentially ordering and compressing /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-
18/install/lppbuild/usr/lpp/powerbroker.configCLIENT1/inst_root/liblpp.a.
g - powerbroker.configCLIENT1.al
g - powerbroker.configCLIENT1.config
g - powerbroker.configCLIENT1.inventory
g - powerbroker.configCLIENT1.post_i
g - powerbroker.configCLIENT1.pre_i
g - powerbroker.configCLIENT1.size
g - powerbroker.configCLIENT1.unpost_i
g - powerbroker.configCLIENT1.unpre_i
pbcreateaixcftpkg: archiving usr package control library...
ar: Creating an archive file /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-
18/install/lppbuild/usr/lpp/powerbroker.configCLIENT1/liblpp.a.
q - powerbroker.configCLIENT1.al
q - powerbroker.configCLIENT1.copyright
q - powerbroker.configCLIENT1.size
ar: Sequentially ordering and compressing /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-
18/install/lppbuild/usr/lpp/powerbroker.configCLIENT1/liblpp.a.
g - powerbroker.configCLIENT1.al
g - powerbroker.configCLIENT1.copyright
g - powerbroker.configCLIENT1.size

pbcreateaixcftpkg: making PowerBroker Unix/Linux AIX configuration package . . .
Backing up to /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-
18/install/lppbuild/powerbroker.configCLIENT1-9.4.3.18.bff.
Cluster 51200 bytes (100 blocks).
Volume 1 on /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-
18/install/lppbuild/powerbroker.configCLIENT1-9.4.3.18.bff
a 404 ./lpp_name
a 0 ./usr/lpp
a 0 ./usr/lpp/powerbroker.configCLIENT1
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/etc
a 6140 ./usr/lpp/powerbroker.configCLIENT1/inst_root/etc/pb.settings
a 6215 ./usr/lpp/powerbroker.configCLIENT1/inst_root/etc/pb.cfg
a 1045 ./usr/lpp/powerbroker.configCLIENT1/inst_root/etc/pb.key
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/opt
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/opt/pbul
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/opt/pbul/policies
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/var
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/var/adm
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/var/adm/pblocald.log
a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/var/adm/pbksh.log

```

```

a 0 ./usr/lpp/powerbroker.configCLIENT1/inst_root/var/adm/pbsh.log
a 178650 ./usr/lpp/powerbroker.configCLIENT1/inst_root/liblpp.a
a 7554 ./usr/lpp/powerbroker.configCLIENT1/liblpp.a
The total size is 200008 bytes.
Backup finished on Sat May 27 14:01:07 PST 2017; there are 400 blocks on 1 volumes.
pbcreateaixcftpkg: AIX lpp package powerbroker.configCLIENT1-9.4.3.18.bff created in /opt/bt_
pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install/lppbuild
pbcreateaixcftpkg: lpp package 'powerbroker.configCLIENT1-9.4.3.18.bff' placed in
/opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install
pbcreateaixcftpkg: build directory for package powerbroker.configCLIENT1 removed.

pbcreateaixcftpkg: the following packages will need to be loaded to the target system:
powerbroker.common powerbroker.runhost powerbroker.submithost powerbroker.sharedlibs

pbcreateaixcftpkg: completed.

```

Installing Component Packages Using the installp Command

This section shows the execution of the `installp` command to install component packages for the submit host, run host, and shared libraries.

Note: Run host is not available for PowerBroker Express.

The execution text also includes copyright, trademark, trade secrets, and other legal text; however, those notices and text were removed from the following excerpt to save space:

```

# cd /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/package
# installp -ad ./ powerbroker.sharedlibs powerbroker.common powerbroker.runhost powerbroker.submithost
+-----+
Pre-installation Verification...
+-----+
Verifying selections...done
Verifying requisites...done
Results...
SUCCESES
-----
Filesets listed in this section passed pre-installation verification
and will be installed.
Selected Filesets
-----
powerbroker.common 9.4.3.18 # BeyondTrust PowerBroker Comm...
powerbroker.runhost 9.4.3.18 # BeyondTrust PowerBroker Run ...
powerbroker.sharedlibs 9.4.3.18 # BeyondTrust PowerBroker Shar...
powerbroker.submithost 9.4.3.18 # BeyondTrust PowerBroker Subm...
<< End of Success Section >>
+-----+
BUILDDATE Verification ...
+-----+
Verifying build dates...done
FILESET STATISTICS
-----

```

```

4 Selected to be installed, of which:
4 Passed pre-installation verification
----
4 Total to be installed
+-----+
Installing Software...
+-----+
installp: APPLYING software for:
powerbroker.common 9.4.3.18
Filesets processed: 1 of 4 (Total time: 1 secs).
installp: APPLYING software for:
powerbroker.runhost 9.4.3.18
Filesets processed: 2 of 4 (Total time: 3 secs).
installp: APPLYING software for:
powerbroker.submithost 9.4.3.18
sysck: 3001-036 WARNING: File
    /usr/lib//libpbul_aca-xcoeff64.so
    is also owned by fileset powerbroker.runhost.
sysck: 3001-036 WARNING: File
    /usr/share/man/man8/pbclienthost_uuid.8
    is also owned by fileset powerbroker.runhost.
sysck: 3001-036 WARNING: File
    /usr/lib//libpbul_aca-xcoeff32.so
    is also owned by fileset powerbroker.runhost.
sysck: 3001-036 WARNING: File
    /usr/sbin/pbclienthost_uuid
    is also owned by fileset powerbroker.runhost.
Filesets processed: 3 of 4 (Total time: 4 secs).
installp: APPLYING software for:
powerbroker.sharedlibs 9.4.3.18
Finished processing all filesets. (Total time: 5 secs).
+-----+
Summaries:
+-----+
Installation Summary
-----
Name Level Part Event Result
-----
powerbroker.common 9.4.3.18 USR APPLY SUCCESS
powerbroker.runhost 9.4.3.18 USR APPLY SUCCESS
powerbroker.submithost 9.4.3.18 USR APPLY SUCCESS
powerbroker.sharedlibs 9.4.3.18 USR APPLY SUCCESS

```

Installing the Configuration Package Using the installp Command

This section shows the execution of the AIX `installp -ad` command to install the configuration package. Following installation of the configuration package, the installation is verified by submitting the `pbrun id` command to PowerBroker for Unix & Linux, and the AIX `lsllpp -l |grep powerbroker` command is used to list the PowerBroker for Unix & Linux packages that are installed.

Note: PowerBroker Express uses `pbssh` instead of `pbrun`. For more information, see the “`pbssh`” program section in the *PowerBroker for Unix & Linux System Administration Guide*.

The execution text also includes copyright, trademark, trade secrets, and other legal text; however, those notices and text were removed from the following excerpt to save space:

```
# cd /opt/bt_pkg/powerbroker/v9.4/pbul_aix52+_9.4.3-18/install
# installp -ad ./ powerbroker.configCLIENT1-9.4.3.18.bff
+-----+
Pre-installation Verification...
+-----+
Verifying selections...done
Verifying requisites...done
Results...
SUCSESSES
-----
Filesets listed in this section passed pre-installation verification
and will be installed.
Selected Filesets
-----
powerbroker.configCLIENT1 9.4.3.18 # BeyondTrust PowerBroker Unix...
<< End of Success Section >>
+-----+
BUILDDATE Verification ...
+-----+
Verifying build dates...done
FILESET STATISTICS
-----
1 Selected to be installed, of which:
1 Passed pre-installation verification
----
1 Total to be installed
+-----+
Installing Software...
+-----+
installp: APPLYING software for:
powerbroker.configCLIENT1 9.4.3.18
Reading pb.cfg...
Checking installation of dependent component packages...
'lpchk -f/-c' of package powerbroker.common succeeded
'lpchk -f/-c' of package powerbroker.runhost succeeded
'lpchk -f/-c' of package powerbroker.submithost succeeded
'lpchk -f/-c' of package powerbroker.sharedlibs succeeded
```



```
Looking for SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
Removing PowerBroker service definitions (if any) from /etc/services.
Adding PowerBroker service definitions to /etc/services.
Removing any PowerBroker definitions from SuperDaemon inetd file /etc/inetd.conf
Adding PowerBroker definitions to SuperDaemon configurations /etc/inetd.conf .
Reloading SuperDaemon Configurations...
0513-095 The request for subsystem refresh was completed successfully.
Done Reloading SuperDaemon Configurations...
Updating Settings in database (if any)...
```

```
Checking installation of package: powerbroker.configCLIENT1
'lpchk -f/-c' of package powerbroker.configCLIENT1 succeeded
Finished processing all filesets. (Total time: 5 secs).
+-----+
Summaries:
+-----+
Installation Summary
-----
Name Level Part Event Result
-----
powerbroker.configCLIENT1 9.4.3.18 USR APPLY SUCCESS
powerbroker.configCLIENT1 9.4.3.18 ROOT APPLY SUCCESS
```

Viewing a List of Installed PowerBroker for Unix & Linux Packages

To view a list of the installed PowerBroker for Unix & Linux packages, do the following. The PowerBroker for Unix & Linux configuration package appears twice because there are usr and root package portions.

```
# lslpp -l | grep powerbroker

powerbroker.common 9.4.3.18 COMMITTED BeyondTrust PowerBroker Common
powerbroker.configCLIENT1
powerbroker.runhost 9.4.3.18 COMMITTED BeyondTrust PowerBroker Run
powerbroker.sharedlibs 9.4.3.18 COMMITTED BeyondTrust PowerBroker Shared
powerbroker.submithost 9.4.3.18 COMMITTED BeyondTrust PowerBroker Submit
powerbroker.configCLIENT1
```

Performing a Cursory Test of PowerBroker for Unix & Linux on the AIX Global Environment

To perform a cursory test of PowerBroker for Unix & Linux on the AIX global environment, type the following:

```
# pbrun id
```

and see results such as those shown in the following example:

```
uid=0 (root) gid=0 (system) groups=2 (bin) , 3 (sys) , 7 (security) , 8 (cron) , 10
(audit) , 11 (lp) , 4 (adm) , 1 (staff) , 6 (mail) , 501 (amanda)
```

Note: For PowerBroker Express, use the following format:

```
#pbssh -h <hostname> -C id
```

Viewing a List of WPARs

WPARs are a new feature of AIX and exist only in AIX V6.1 and higher. To view a list of WPARs, type the following:

```
# lswpar
```

and view a list as shown in the following example:

```
Name State Type Hostname Directory
-----
wpar01 A S wpar01 /wpars/wpar01
```

Using syncwpar to Propagate Additional Packages to Shared WPARs

The `syncwpar` command synchronizes all packages between the AIX global environment and shared WPARs. This section shows how to use `syncwpar` to propagate additional AIX global environment packages to shared WPARs. WPARs are a feature that exists only in AIX V6.1 and higher.

```
# syncwpar wpar01
*****
**
Synchronizing workload partition wpar01 (1 of 1).
*****
**
Executing /usr/sbin/syncroot in workload partition wpar01. syncroot: Processing root part installation
status. syncroot: Synchronizing installp software.
+-----+
+
Pre-installation Verification...
+-----+
+
Verifying selections...done Verifying requisites...done Results...

SUCSESSES
-----
Filesets listed in this section passed pre-installation verification and will be installed.

Selected Filesets
-----
powerbroker.configClient 6.2.0.1 # BeyondTrust PowerBroker Conf...

<< End of Success Section >>

+-----+
+
BUILDDATE Verification ...
+-----+
+
Verifying build dates...done FILESET STATISTICS
-----
1 Selected to be installed, of which:
1 Passed pre-installation verification
```

```

-----
1 Total to be installed

+-----+
+
Installing Software...
+-----+
+

installp: APPLYING software for: powerbroker.configClient 6.2.0.1

Reading pb.cfg...
Checking installation of dependent component packages... 'lppchk -f/-c' of package powerbroker.common
succeeded 'lppchk -f/-c' of package powerbroker.runhost succeeded 'lppchk -f/-c' of package
powerbroker.submithost succeeded 'lppchk -f/-c' of package powerbroker.sharedlibs succeeded Looking for
SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
Removing PowerBroker service definitions (if any) from /etc/services. Adding PowerBroker service
definitions to /etc/services.
Removing any PowerBroker definitions from SuperDaemon inetd file
/etc/inetd.conf
Adding PowerBroker definitions to SuperDaemon configurations /etc/inetd.conf . Reloading SuperDaemon
Configurations...
0513-095 The request for subsystem refresh was completed successfully. Done Reloading SuperDaemon
Configurations...
Checking installation of package: powerbroker.configClient 'lppchk -f/-c' of package
powerbroker.configClient succeeded Finished processing all filesets. (Total time: 2 secs).

+-----+
+
Summaries:
+-----+
+

Installation Summary
-----
Name Level Part Event Result
-----
-
powerbroker.configClient 6.2.0.1 ROOT APPLY SUCCESS syncroot: Processing root part installation status.
syncroot: Installp root packages are currently synchronized. syncroot: RPM root packages are currently
synchronized. syncroot: Root part is currently synchronized.
syncroot: Returns Status = SUCCESS
Workload partition wpar01 synchronized successfully. Return Status = SUCCESS.

```

Logging into Shared WPARs

WPARs are a feature that exists only in AIX V6.1 and higher.

To login to shared WPARs, type the following:

```
# clogin wpar01
```

and see a welcome message such as the following:

```
* *
* Welcome to AIX Version 6.1! *
* *
```

Running a Cursory Test of PowerBroker on a Shared WPAR System

WPARs are a feature that exists only in AIX V6.1 and higher.

To run a cursory test of PowerBroker for Unix & Linux on a shared WPAR system, type the following:

```
# pbrun id
```

and see results such as those shown in the following example:

```
uid=0(root) gid=0(system) groups=2(bin),3(sys),7(security),8(cron),10
(audit),11(lp)
```

Note: For PowerBroker Express, use the following format:

```
#pbssh -h <hostname> -C id
```

Sample of Removing an AIX Package Installation

This section shows the execution of the AIX `installp -u` command to remove the PowerBroker for Unix & Linux packages.

```
# installp -u powerbroker
+-----+
Pre-deinstall Verification...
+-----+
Verifying selections...done
Verifying requisites...done
Results...
SUCCESES
-----
Filesets listed in this section passed pre-deinstall verification
and will be removed.
Selected Filesets
-----
powerbroker.common 9.4.3.18 # BeyondTrust PowerBroker Comm...
powerbroker.configCLIENT1 9.4.3.18 # BeyondTrust PowerBroker Unix...
powerbroker.runhost 9.4.3.18 # BeyondTrust PowerBroker Run ...
powerbroker.sharedlibs 9.4.3.18 # BeyondTrust PowerBroker Shar...
powerbroker.submithost 9.4.3.18 # BeyondTrust PowerBroker Subm...
<< End of Success Section >>
FILESET STATISTICS
-----
5 Selected to be deinstalled, of which:
5 Passed pre-deinstall verification
-----
```

```

5 Total to be deinstalled
+-----+
Deinstalling Software...
+-----+
installp: DEINSTALLING software for:
powerbroker.configCLIENT1 9.4.3.18
Reading pb.cfg...
Looking for SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
Removing PowerBroker service definitions (if any) from /etc/services.
Removing any PowerBroker definitions from SuperDaemon inetd file /etc/inetd.conf
Reloading SuperDaemon Configurations...
0513-095 The request for subsystem refresh was completed successfully.
Done Reloading SuperDaemon Configurations...
Filesets processed: 1 of 5 (Total time: 6 secs).
installp: DEINSTALLING software for:
powerbroker.runhost 9.4.3.18
Filesets processed: 2 of 5 (Total time: 6 secs).
installp: DEINSTALLING software for:
powerbroker.sharedlibs 9.4.3.18
Filesets processed: 3 of 5 (Total time: 7 secs).
installp: DEINSTALLING software for:
powerbroker.submithost 9.4.3.18
Filesets processed: 4 of 5 (Total time: 7 secs).
installp: DEINSTALLING software for:
powerbroker.common 9.4.3.18
Removing /opt/pbul
Finished processing all filesets. (Total time: 8 secs).
+-----+
Summaries:
+-----+
Installation Summary
-----
Name Level Part Event Result
-----
powerbroker.configCLIENT1 9.4.3.18 ROOT DEINSTALL SUCCESS
powerbroker.configCLIENT1 9.4.3.18 USR DEINSTALL SUCCESS
powerbroker.runhost 9.4.3.18 USR DEINSTALL SUCCESS
powerbroker.sharedlibs 9.4.3.18 USR DEINSTALL SUCCESS
powerbroker.submithost 9.4.3.18 USR DEINSTALL SUCCESS
powerbroker.common 9.4.3.18 USR DEINSTALL SUCCESS

```

Example of Using syncwpar to Propagate Package Removal From Shared WPARs

The `syncwpar` command synchronizes all packages between the AIX global environment and shared WPARs. This section shows an example of how to use the `syncwpar` command to propagate removal of AIX global environment packages from shared WPARs. WPARs are a feature that exists only in AIX V6.1 and higher.

Note: When syncwpar is run and a PowerBroker configuration package is removed, the following message may display:

```
inulag: The file system has read permission only.
```

This message can be ignored.

```
# syncwpar wpar01
*****
**
Synchronizing workload partition wpar01 (1 of 1).
*****
**
Executing /usr/sbin/syncroot in workload partition wpar01. syncroot: Processing root part installation
status. syncroot: Synchronizing installp software.
+-----+
+
Pre-deinstall Verification...
+-----+
+
Verifying selections...done Verifying requisites...done Results...

SUCSESSES
-----
Filesets listed in this section passed pre-deinstall verification and will be removed.

Selected Filesets
-----
powerbroker.configClient 6.2.0.1 # BeyondTrust PowerBroker Conf...

<< End of Success Section >> FILESET STATISTICS

-----
1 Selected to be deinstalled, of which:
1 Passed pre-deinstall verification
----
1 Total to be deinstalled

+-----+
+
Deinstalling Software...
+-----+
+

installp: DEINSTALLING software for: powerbroker.configClient 6.2.0.1

Reading pb.cfg...
Looking for SuperDaemons to configure...
Finished looking for SuperDaemons to configure...
```

```
Removing PowerBroker service definitions (if any) from /etc/services. Removing any PowerBroker
definitions from SuperDaemon inetd file
/etc/inetd.conf
Reloading SuperDaemon Configurations...
0513-095 The request for subsystem refresh was completed successfully. Done Reloading SuperDaemon
Configurations...
inulag: The file system has read permission only. Finished processing all filesets. (Total time: 1
secs).
```

```
+-----+
+
Summaries:
+-----+
+
```

Installation Summary

```
-----
```

```
Name Level Part Event Result
```

```
-----
-
```

```
powerbroker.configClient 6.2.0.1 ROOT DEINSTALL SUCCESS syncroot: Processing root part installation
status.
```

```
syncroot: Installp root packages are currently synchronized. syncroot: RPM root packages are currently
synchronized. syncroot: Root part is currently synchronized.
```

```
syncroot: Returns Status = SUCCESS
```

```
Workload partition wpar01 synchronized successfully. Return Status = SUCCESS.
```

Verifying Removal of PowerBroker for Unix & Linux Packages

To verify that all PowerBroker for Unix & Linux packages were removed, type the following:

```
# lslpp -l | grep powerbroker
```

and receive the message:

```
# <no output.>
```

HP-UX Package Installer

This section describes how to install PowerBroker for Unix & Linux using a package installer for HP-UX 11i v1, 11i v2, or 11i v3. Use the HP-UX package installation if you want to install PowerBroker for Unix & Linux using the HP-UX Software Distributor (SD) on a local or remote computer.

Note: The PowerBroker for Unix & Linux HP-UX package installer that is described here is not compatible with the PowerBroker version 5 HP-UX depots. If the PowerBroker version 5 HP-UX depots are installed, you must remove them before installing the PowerBroker for Unix & Linux version 6 HP-UX depots.

Prerequisites

To use the PowerBroker for Unix & Linux HP-UX package installer, you must have the following:

- Package tarball file for the appropriate PowerBroker for Unix & Linux flavor

Note: For the PowerBroker for Unix & Linux HP-UX package installer, the tarball files are cumulative. That is, an update tarball file contains a complete PowerBroker for Unix & Linux installation. It is not necessary to install a baseline version of PowerBroker for Unix & Linux before installing an update.

- Root access or superuser privileges

Note: The PowerBroker for Unix & Linux HP-UX package installer does not support prefix/suffix installations.

Planning Your Installation

When preparing to use the PowerBroker for Unix & Linux HP-UX package installer, you should be familiar with the following concepts and restrictions:

Depots and Filesets - HP-UX packaged software is delivered as a single file called a depot (`.depot`) file. A depot can be thought of as a compressed file that contains one or more filesets. A fileset is a component of the software and may contain many files. Installing an HP-UX depot extracts the files from the filesets and writes them to the appropriate directory locations.

Component depot and component filesets - A PowerBroker for Unix & Linux component fileset is a part of the PowerBroker for Unix & Linux component depot that installs a portion of the PowerBroker for Unix & Linux application. There are seven PowerBroker for Unix & Linux component filesets. In the following list, arch is the architecture of the target platform; for example, ia64A.

- `PowerBroker-arch.MASTERHOST` - Contains Policy Server host, pbsync, and pbsyncd
- `PowerBroker-arch.SUBMITHOST` - Contains submit host and PowerBroker for Unix & Linux shells
- `PowerBrokerX-arch.EXPRESS` - Contains files that are specific to the PowerBroker Express product; cannot be installed with the submit host or run host components
- `PowerBroker-arch.RUNHOST` - Contains run host and PowerBroker for Unix & Linux utilities
- `PowerBroker-arch.LOGHOST` - Contains log host, pbsync, and pbsyncd
- `PowerBroker-arch.GUIHOST` - Contains GUI host and secure GUI host
- `PowerBroker-arch.SHAREDLIBS` - Contains shared libraries
- `PowerBroker-arch.RESTHOST` - Contains REST API files.
- `PowerBroker-arch.RNSSVR` - Contains Registry Name Service files.

- `PowerBroker-arch.LICSVR` - Contains License Server files.

Which component filesets are required depends on the type of PowerBroker for Unix & Linux host you are creating, such as Policy Server host, submit host, and so on. PowerBroker Express uses the MASTER, EXPRESS, LOG, GUI, and SHAREDLIBS component filesets. Other PowerBroker for Unix & Linux products use the MASTER, SUBMIT, RUN, LOG, GUI, and SHAREDLIBS filesets. You can select the types of PowerBroker for Unix & Linux hosts in the `pbinstall` installation menu, as shown in the following table:

Table 11. Required Components

Menu Selection	Required Components
Install everything here (demo mode)? = Yes	MASTERHOST RUNHOST SUBMITHOST LOGHOST GUIHOST SHAREDLIBS
Install PowerBroker Policy Server Host? = Yes	MASTERHOST
Install PowerBroker Run Host? = Yes	RUNHOST
Install PowerBroker Submit Host? = Yes	SUBMITHOST
Install PowerBroker Log Host? = Yes	LOGHOST
Install PowerBroker GUI Host? = Yes	GUIHOST
Install PowerBroker Secure GUI Host? = Yes	GUIHOST
Install BeyondTrust built-in third-party libraries? = Yes	SHAREDLIBS
Install Registry Name Services Server? [yes]	RNSSVR
Install License Server? [yes]	LICSVR

Configuration depot - HP-UX depot (separate from the component depot) that is used to install the following files:

- `pb.settings`
- `pb.cfg`
- `pb.key` (if applicable)
- `pb.conf` (for Policy Server hosts)
- Diagnostic logs files

The PowerBroker for Unix & Linux configuration depot is created by the `pbcreatehpuxcfgpkg` program. The component filesets must be copied to the SD depot (using the `swcopy` command) before you copy the configuration fileset to the distribution depot.

SD Depot - The SD depot is the software distribution depot, to which software depots are copied (by using the HP-UX `swcopy` command) prior to the installation of their filesets. By default, `/var/spool/sw` is the location of the SD depot.

pbinstall program - To create the PowerBroker for Unix & Linux settings files, you use the pbinstall program with the -z (“settings only”) option. pbinstall -z only creates the settings files and is incompatible with the following command line options:

Table 12. Options Incompatible with pbinstall -z

Option	Description
-b	Runs pbinstall in batch mode.
-c	Skip the steps that process or update the PowerBroker for Unix & Linux settings file.
-e	Runs install script automatically by bypassing the menu step of pbinstall.
-i	Ignores previous pb.settings and pb.cfg files.
-p	Sets the pb installation prefix.
-s	Sets the pb installation suffix.
-u	Install the utility programs.
-x	Creates a log synchronization host (that is, installs pbsyncd).

When you execute `pbinstall` with the -z option, you can see two menu items that are not otherwise available:

Enter existing pb.settings path

Enables you to specify your own pb.settings file. pbinstall reads this settings file and populates the remaining menu choices. You can override some menu choices.

If set to none, then pbinstall does not read a settings file. The remaining menu choices are populated with default values.

Enter directory path for settings file creation

Enables you to specify an alternative output directory for the settings files. The default directory is:

```
/unzip-dir/powerbroker/version/<flavor>/install/settings_files
```

where unzip-dir is the directory where the package tarball file was unzipped and version is the PowerBroker for Unix & Linux version number.

The behavior of `pbinstall -z` depends on whether certain additional command line options are specified:

- If no other command line options are specified, pbinstall initially presents a short version of the installation menu (items 1–8 only). Depending on the choices you make in these items, further menu items become available.
- If command line options -g, -l, -m, -o, -r, or -w are specified, pbinstall presents an expanded version of the installation menu that reflects the host types that you are configuring.

For complete information about the pbinstall command-line options, see [pbinstall](#).

When running pbinstall with the -z option, the following menu items are pre-programmed and cannot be changed:

Install man pages? PowerBroker daemon location

Administration programs location User programs location

GUI library location

Policy include (sub) file directory User man page location
 Admin man page location Policy filename
 BeyondTrust built-in third-party library directory

In addition, the values of the following menu items determine the values of other menu items:

Table 13. Options Preset When Running `pbinstall -z`

Setting this menu option to Yes	Sets these values to Yes
Install PowerBroker Policy Server Host?	Install PowerBroker Synchronization? PowerBroker synchronization can be initiated from this host?
Install PowerBroker Run Host?	Install PowerBroker Utilities? Note: This item is not applicable for PowerBroker Express.
Install PowerBroker Submit Host?	Install PBSSH? Install PowerBroker pbksh? Install PowerBroker pbsh? Will this host use a Log Host? Note: The pbksh and pbsh shell options are not available for PowerBroker Express. For PowerBroker Express, the submit host option installs <code>pbssh</code> . For the other PowerBroker for Unix & Linux products, this option installs <code>pbrun</code> .
Install PowerBroker Log Host?	Install PowerBroker Synchronization? PowerBroker synchronization can be initiated from this host?

Note: If you are using the package installer to install PowerBroker for Unix & Linux on a computer that already has an interactive PowerBroker for Unix & Linux installation on it, see [Interactive and Packaged Installations on the Same Computer](#), for additional considerations.

Overview of Steps

Using the PowerBroker for Unix & Linux HP-UX package installer involves the following steps (described in more detail in the next section [Installation Procedure](#)):

1. Unpack the PowerBroker for Unix & Linux or PowerBroker Express HP-UX package tarball file.
2. Use the `pbinstall` program to create PowerBroker for Unix & Linux settings files.
3. Use the `pbcreatehpuxcfgpkg` program to create the PowerBroker for Unix & Linux configuration depot.
4. Use the HP-UX `swcopy` command to copy the PowerBroker for Unix & Linux component depot to the desired SD depot.
5. Use the HP-UX `swcopy` command to copy the PowerBroker for Unix & Linux configuration depot to the desired SD depot.
6. Use the HP-UX `swinstall` command to install the PowerBroker for Unix & Linux configuration depot. The dependencies that are identified in the configuration files set will cause the appropriate component files sets to be installed as well.

Installation Procedure

To install PowerBroker for Unix & Linux using the HP-UX SD feature, do the following:

1. Extract the package tarball files into the `/opt/beyondtrust/` directory by executing the following command:


```
gunzip -c pb<flavor_version>_pkg.tar.Z | tar xvf -
```
2. Navigate to the directory: `/opt/beyondtrust/powerbroker/version/flavor/install/`
3. Execute the following command:


```
./pbinstall -z
```

You can include other options with the `-z` option (see [Planning Your Installation](#)).

You will be asked if you want to use Client Registration. If you are planning to enable Registry Name Service, and are installing on a host that is not designated as a Primary Server, you must run Client Registration.

`pbinstall` will then ask if you want to enable Registry Name Service.

`pbinstall` displays the PowerBroker for Unix & Linux installation menu (see [pbinstall Installation Script](#) for details).

4. Make your menu selections. Note that the `Enter existing pb.settings` path menu option enables you to specify your own `pb.settings` file to use. Also, the `Enter directory path for settings file creation` menu option enables you to specify where to save the generated settings files. These menu options are available only when running `pbinstall` with the `-z` option.

When the menu selection process is complete, `pbinstall` creates the following files in the specified location:

 - `pb.settings`
 - `pb.cfg`
 - `pb.key` (if encryption is enabled)
 - `pb.conf` (for Policy Server host)
5. Optional. For a PowerBroker for Unix & Linux client, if client-server communications are to be encrypted, replace the generated `pb.key` file with `pb.key` file from the Policy Server host. Also, copy any other required key files into the same directory.
6. Optional. For a Policy Server host, write a policy file (`pb.conf`) and place it in the directory with the other generated files. If you do not provide a `pb.conf` file, a `pb.conf` file with the single command “reject;” will be generated and packaged.

Starting with v8.0, `pbinstall -z` can optionally install the default role-based policies and will ask: Installing default role-based policy `pbul_policy.conf` and `pbul_functions.conf` in `<install_dir>/settings_files`

Would you like to use the default role-based policy in the configuration package?

- Answer Yes for new installs only.
- If you are upgrading an existing configuration package, to avoid overwriting your existing policy, answer No.

Use the default role-based policy [Y]?

- If you answer Yes, the default `pb.conf`, `pbul_policy.conf` and `pbul_functions.conf` will be created and installed on the Policy Server.
- If you are installing over an existing installing, and have an existing policy in place, answer No.

7. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/version/flavor/install/
```

8. Run the `pbcreatehpuxcfgpkg` utility by typing:

```
pbcreatehpuxcfgpkg [-d] -p depot-fileset-name -s directory
```

where:

- `-d` is an option that sets the component fileset dependency to `hppaD` rather than the default `hppaB`.
- `depot-fileset-name` is a user-specified name for the configuration fileset. The resulting fileset will be `PowerBroker-Cfg.depot-fileset-name` (`PowerBrokerX-Cfg.depot-fileset-name` for `PowerBroker Express`).
- `directory` is the directory that contains the `PowerBroker` for Unix & Linux settings and configuration files to include in the configuration fileset.

The `pbcreatehpuxcfgpkg` utility creates the configuration depot with the file name `PowerBroker-Cfg-version.depot-fileset-name.depot` (`PowerBrokerX-Cfg-version.depot-fileset-name.depot` for `PowerBroker Express`).

For more information about the `pbcreatehpuxcfgpkg` utility, see [pbcreatehpuxcfgpkg](#).

9. Navigate to the directory:

```
/opt/beyondtrust/powerbroker/version/flavor/package/
```

10. Run the HP-UX `swcopy` utility to copy the `PowerBroker` for Unix & Linux component depot to the desired SD depot by typing:

```
swcopy -s /path/PowerBroker-arch.depot PowerBroker-arch.FILESET [@ sd-  
directory]
```

where

- `path` is the absolute path to the directory that contains the `PowerBroker` for Unix & Linux component depot
- `arch` is the target platform architecture
- `FILESET` is the specific fileset to be copied; alternatively, use `*` instead of `PowerBroker-arch.FILESET` to copy all filesets
- `sd-directory` is the desired SD directory; if you omit `@ sd-directory`, the default `/var/spool/sw` is used.

For example, to copy only the log host component fileset:

```
# swcopy -s /opt/beyondtrust/powerbroker/v9.4/pbul_hpux.hppa64_  
9.4.3/package/PowerBroker-hppa64-9.4.3.06.depot PowerBroker-  
hppa64.LOGHOST @ /var/spool/sw
```

To copy the log host and Policy Server host component filesets to the default SD depot:

```
# swcopy -s /opt/beyondtrust/powerbroker/v9.4/pbul_hpux.hppa64_9.4.3-  
06/package/PowerBroker-hppa64-9.4.3.06.depot PowerBroker-hppa64.LOGHOST  
PowerBroker-hppa64.MASTERHOST
```

To copy all component filesets to the default SD depot:

```
swcopy -s /opt/beyondtrust/powerbroker/v9.4/pbul_hpux.hppa64_9.4.3-  
06/package/PowerBroker-hppa64-9.4.3.06.depot \*
```

11. Run the HP-UX `swcopy` utility to copy the PowerBroker for Unix & Linux configuration fileset to the desired SD depot. For example:

```
# swcopy -s /opt/beyondtrust/powerbroker/v9.4/pbul_hpux.hppa64_9.4.3-
  06/install/PowerBroker-Cfg-9.4.3.06.CLIENT.depot PowerBroker-Cfg.CLIENT
  @ /var/spool/sw
```

12. Run the HP-UX `swinstall` utility to install the PowerBroker for Unix & Linux configuration fileset by typing:

```
swinstall PowerBroker-Cfg.depot-fileset-name
```

where `depot-fileset-name` is the configuration fileset name that you specified when you created the PowerBroker for Unix & Linux configuration package in step 8. Any component dependencies that are identified by the configuration fileset are automatically installed as well.

Note: If you attempt to install filesets from more than one flavor onto a single system, the installation will fail with an error message.

13. Verify the installation of the filesets with the HP-UX `swverify` utility by typing:

```
swverify PowerBroker-arch
```

or

```
swverify PowerBroker-Cfg
```

Note: Many of the HP-UX depot management commands display a message regarding where to find a log file that contains additional information. It is recommended that you look at these log files, because some important diagnostic information appears in the log file but not in the utility's standard output.

Removing PowerBroker for Unix & Linux Filesets

Removing the PowerBroker for Unix & Linux depots completely uninstalls PowerBroker for Unix & Linux from a computer. Because the component filesets are dependencies of the configuration fileset, the configuration fileset must be removed first. To remove the PowerBroker for Unix & Linux filesets, do the following:

1. Remove the PowerBroker for Unix & Linux configuration fileset by typing:

```
swremove PowerBroker-Cfg.depot-fileset-name
```

where `depot-fileset-name` is the name of the fileset that you specified when you created the configuration depot.

Note: For PowerBroker Express, use the depot name `PowerBrokerX-Cfg`

2. Remove the PowerBroker for Unix & Linux component filesets by typing:

```
swremove PowerBroker-arch
```

Note: You can remove the configuration and component filesets in the same command, for example:

```
swremove PowerBroker-Cfg.FILESET PowerBroker-arch
```

Remote Installation

Because the HP-UX SD system uses a daemon for software administration, you can install from a local depot to a remote machine, or install from a remote depot to a local machine. Additionally, you can install a depot to an “alternate root” and then remount the alternate root as an actual root on another node.

To install a depot on a remote system, you must have ACL access to that remote system; you can use the `swacl` command to manage these access controls. Use the `@` argument with the `swinstall` command, for example:

```
swinstall PowerBroker-hppaB @ remotehost:/
```

To install a depot on an alternate root, you also use the `@` argument, for example:

```
swinstall PowerBroker-hppaB @ /export/shared_root/node1
```

Note: For alternate root installation, you must run the swconfig utility on the actual node, after the alternate root is remounted as the node's actual root.

For more information, see the man pages for the HP-UX SD commands.

Updating PowerBroker for Unix & Linux with Update Depots

The PowerBroker for Unix & Linux HP-UX package installer can be used to update an existing PowerBroker for Unix & Linux installation to a new version. The existing PowerBroker for Unix & Linux version should have been installed using the PowerBroker for Unix & Linux package installer.

If you have a previous installation of PowerBroker for Unix & Linux, you must first uninstall it before installing PowerBroker Express. Likewise, if you have an installation of PowerBroker Express, you must uninstall it before installing PowerBroker for Unix & Linux.

Update Depot Considerations

Installing a PowerBroker for Unix & Linux update depot is similar to using the HP-UX package installer to install PowerBroker for Unix & Linux for the first time. Keep these considerations in mind when you prepare to upgrade PowerBroker for Unix & Linux:

- A PowerBroker for Unix & Linux HP-UX update depot contains a complete PowerBroker for Unix & Linux installation, not just the files that have changed since the previous release.
- Each PowerBroker for Unix & Linux update depot is cumulative; that is, it includes all previous update filesets that BeyondTrust released since the baseline version. Therefore, there is no need to install the previous update depots.
- Removal and reinstallation of the configuration fileset is not required when updating an existing PowerBroker for Unix & Linux installation.

Unlike PowerBroker for Unix & Linux patches that are installed with `pbpatchinstall`, update filesets cannot be rolled back to a previous release. However, you can install an older fileset over a newer one, effectively rolling back to the older release. For more information, see [Reverting to a Previous Version](#).

Update Depot Procedure

Follow this procedure to update your installation of PowerBroker for Unix & Linux using the update depots:

1. Obtain the tarball file for the HP-UX update depots that are appropriate for your hardware. The tarball file name has the format `pb<flavor>-v.v.r-bb-update_pkg.tar.Z`, where:
 - `<flavor>` indicates the operating system and hardware architecture
 - `v.v.r` is the major and minor version number and the release number
 - `bb` is the build number
2. Extract the depot files into the `/opt/beyondtrust/` directory by executing the following command:


```
tar xvfz pb<flavor_version>-update_pkg.tar.Z
```
3. Navigate to the directory:


```
/opt/beyondtrust/powerbroker/version/flavor/package/
```
4. Run the HP-UX `swcopy` utility to copy the PowerBroker for Unix & Linux component depot to the desired SD depot by typing:

```
swcopy -s /path/PowerBroker-arch.depot PowerBroker-arch.FILESET [@ sd-
  directory]
```

where

- path is the absolute path to the directory that contains the PowerBroker for Unix & Linux component depot
 - arch is the target platform architecture
 - FILESET is the specific fileset to be copied; alternatively, use `*` instead of `PowerBroker-arch.FILESET` to copy all filesets
 - sd-directory is the desired SD directory; if you omit `@ sd-directory`, the default `/var/spool/sw` is used.
5. Run the HP-UX `swinstall` utility to install the PowerBroker for Unix & Linux component filesets by typing:


```
swinstall PowerBroker-arch
```
 6. Verify the installation of the filesets with the HP-UX `swverify` utility by typing:


```
swverify PowerBroker-arch
```

Reverting to a Previous Version

Unlike PowerBroker for Unix & Linux patches that are installed with `pbpatchinstall`, update depots cannot be rolled back to a previous release. However, you can install an older fileset over a newer one, effectively rolling back to the older release. To install older filesets over newer ones, use the following command:

```
swinstall -x allow_downdate=true PowerBroker-arch
```

This command restores the previous release. Repeat the command to restore earlier releases.

Sample Execution for the HP-UX Package Installer

The sample execution shows the installation of a PowerBroker for Unix & Linux submit host, run host, and shared libraries using the PowerBroker for Unix & Linux HP-UX package installer.

This sample execution is divided into the following parts:

- Generate the PowerBroker for Unix & Linux settings files
- Create the PowerBroker for Unix & Linux configuration depot using the `pbcreatehpuxcfgpkg` program
- Copy the component and configuration depots to the SD depot using the `swcopy` command
- Install the configuration and component filesets using the `swinstall` command

Generating the PowerBroker for Unix & Linux Settings Files

This section of the execution shows the generation of the PowerBroker for Unix & Linux settings files (`pb.key`, `pb.cfg`, and `pb.settings`) and also displays the PowerBroker for Unix & Linux installation menu. This output was generated using the `pbinstall` program with the `-z` option and selecting menu options to install a run host and a submit host.

```
# ./pbinstall -z
Starting pbinstall main() from /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/.
hpux.ia64
PowerBroker for Unix & Linux Settings File Generation
```


Please read the PowerBroker for Unix & Linux Installation Instructions before proceeding.

Checking MANIFEST against release directory

Press return to continue

The Registry Name Service of PowerBroker for Unix & Linux facilitates location of other services within the PBUL enterprise with the aid of a centralized data repository.

IMPORTANT: Client Registration is required if this is not the Primary Server and you intend to use Registry Name Services.

Do you wish to utilize Registry Name Service? [yes]? no

BeyondTrust PowerBroker for Unix & Linux Installation Menu

Opt Description [Value]

- 1 Install Everything Here (Demo Mode)? [no]
- 2 Enter existing 'pb.settings' path [none]
- 3 Enter directory path for settings file ... [/opt/pbpkg/powerbroker/v9....]
- 6 Install PowerBroker Policy Server Host? [no]
- 7 Install PowerBroker Run Host? [no]
- 8 Install PowerBroker Submit Host? [no]
- 10 Install PBSUDO Policy Server? [no]
- 11 Install PowerBroker Log Host? [no]
- 15 Install File Integrity Monitoring Polic... [no]

N for the next menu page, C to continue, X to exit

Please enter a menu option [For technical support call 1-800-234-9072]> 7

PowerBroker for Unix & Linux executes secured tasks on hosts which are designated as Run Hosts. These hosts execute the commands using the plocald daemon.

To allow PowerBroker for Unix & Linux to execute a command, a host must be configured as a Run Host.

Do you want this host to be a Run Host [no]? yes

BeyondTrust PowerBroker for Unix & Linux Installation Menu

Opt Description [Value]

- 1 Install Everything Here (Demo Mode)? [no]
- 2 Enter existing 'pb.settings' path [none]
- 3 Enter directory path for settings file ... [/opt/pbpkg/powerbroker/v9....]
- 6 Install PowerBroker Policy Server Host? [no]
- 7 Install PowerBroker Run Host? [yes]
- 8 Install PowerBroker Submit Host? [no]
- 10 Install PBSUDO Policy Server? [no]
- 11 Install PowerBroker Log Host? [no]
- 15 Install File Integrity Monitoring Polic... [no]
- 22 Install PowerBroker GUI Host? [no]
- 23 Install PowerBroker Secure GUI Host? [no]
- 24 Install PowerBroker Utilities: pbvi, pb... [yes]
- 27 Install PowerBroker man pages? [yes]

```

28 Will this host use a Log Host? [no]
29 PowerBroker Identity Services Integration? [no]
54 PowerBroker synchronization can be init... [no]
55 PowerBroker daemon location [/usr/sbin]
58 User programs location [/usr/local/bin]
N for the next menu page, C to continue, X to exit
Please enter a menu option [For technical support call 1-800-234-9072]> 8

```

PowerBroker for Unix & Linux allows requests for secured tasks to be made on hosts configured as Submit Hosts.

To have pbrun initiate requests for secured tasks, this host must be a Submit Host.

```

Do you want this host to be a Submit Host [no]? yes
BeyondTrust PowerBroker for Unix & Linux Installation Menu
Opt Description [Value]
1 Install Everything Here (Demo Mode)? [no]
2 Enter existing 'pb.settings' path [none]
3 Enter directory path for settings file ... [/opt/pbpkg/powerbroker/v9....]
6 Install PowerBroker Policy Server Host? [no]
7 Install PowerBroker Run Host? [yes]
8 Install PowerBroker Submit Host? [yes]
9 Install PowerBroker PBSSH [yes]
10 Install PBSUDO Policy Server? [no]
11 Install PowerBroker Log Host? [no]
15 Install File Integrity Monitoring Polic... [no]
17 Path to PowerBroker Password Safe 'pkru... []
22 Install PowerBroker GUI Host? [no]
23 Install PowerBroker Secure GUI Host? [no]
24 Install PowerBroker Utilities: pbvi, pb... [yes]
25 Install PowerBroker pbksh? [yes]
26 Install PowerBroker pbsh? [yes]
27 Install PowerBroker man pages? [yes]
28 Will this host use a Log Host? [yes]
29 PowerBroker Identity Services Integration? [no]
54 PowerBroker synchronization can be init... [no]
55 PowerBroker daemon location [/usr/sbin]
56 Number of reserved spaces for submit pr... [not-supported]
58 User programs location [/usr/local/bin]
62 User man page location [/usr/local/man/man1]
77 Configure target system's SuperDaemon? [yes]
80 Policy Server Delay [500]
81 Policy Server Protocol Timeout [-1]
88 List of PowerBroker Policy Servers to s... []
89 pbrun diagnostic log? [none]
90 pbssh diagnostic log? [none]
91 Allow Local Mode? [yes]

```

```
92 Additional secured task checks? [no]
93 Suppress Policy Server host failover er... [yes]
94 List of PowerBroker Policy Servers to a... []
95 pblocald diagnostic log [/var/adm/pblocald.log]
96 Command line options for pblocald []
97 Syslog pblocald sessions? [no]
98 Record PTY sessions in utmp/utmpx? [yes]
99 Validate Policy Server Host Connections? [no]
100 List of PowerBroker Log Hosts []
101 Command line options for pblogd []
102 Log Host Delay [500]
103 Log Host Protocol Timeout [-1]
105 List of PowerBroker log reserved filesy... [none]
112 Add installed shells to /etc/shells [no]
113 PowerBroker pbksh diagnostic file [/var/adm/pbksh.log]
114 PowerBroker pbsh diagnostic file [/var/adm/pbsh.log]
115 Stand-alone pblocald command [none]
116 Stand-alone root shell default iolog [/pbshell.iolog]
121 Use syslog? [yes]
122 Syslog facility to use? [LOG_AUTH]
123 Base Daemon port number [49200]
124 pbmasterd port number [49200]
125 pblocald port number [49201]
126 pblogd port number [49202]
131 Add entries to '/etc/services' [yes]
132 Allow non-reserved port connections [yes]
133 Inbound Port range [1025-65535]
134 Outbound Port range [1025-65535]
137 PowerBroker network encryption options [aes-256:keyfile=/etc/pb.key]
142 PowerBroker settings file encryption type [none]
143 PowerBroker REST API encryption options [aes-256:keyfile=/etc/pb.key]
144 Configure with Kerberos v5? [no]
150 Enforce High Security Encryption? [yes]
151 Use SSL? [yes]
152 SSL Configuration? [requiresl]
153 SSL pbrun Certificate Authority Directory? [none]
154 SSL pbrun Certificate Authority File? [none]
155 SSL pbrun Cipher List? [HIGH:!MD5:@STRENGTH]
156 SSL pbrun Certificate Directory? [none]
157 SSL pbrun Certificate File? [none]
158 SSL pbrun Private Key Directory? [none]
159 SSL pbrun Private Key File? [none]
160 SSL pbrun Certificate Subject Checks? [none]
161 SSL Server Certificate Authority Direct... [none]
162 SSL Server Certificate Authority File? [none]
163 SSL Server Cipher List? [HIGH:!MD5:@STRENGTH]
164 SSL Server Certificate Directory? [none]
165 SSL Server Certificate File? [/etc/pbssl.pem]
```

```

166 SSL Server Private Key Directory? [none]
167 SSL Server Private Key File? [/etc/pbssl.pem]
168 SSL Server Certificate Subject Checks? [none]
169 PowerBroker SSL Certificate Country Code [US]
170 PowerBroker SSL Certificate State/Province [AZ]
171 PowerBroker SSL Certificate Location (T... [Phoenix]
172 PowerBroker SSL Certificate Organizatio... [Security]
173 PowerBroker SSL Certificate Organization [BeyondTrust]
174 Configure PowerBroker with LDAP? [no]
175 Install BeyondTrust built-in third-part... [yes]
176 BeyondTrust built-in third-party librar... [/usr/lib/beyondtrust/pb]
188 Use PAM? [no]
196 Allow Remote Jobs? [yes]
197 PowerBroker UNIX Domain Socket directory [none]
198 Reject Null Passwords? [no]
199 Enable TCP keepalives? [no]
200 Name Resolution Timeout [0]
N for the next menu page, P for the previous menu page, C to continue, X to exit
Please enter a menu option [For technical support call 1-800-234-9072]> c

```

```

ypcat: no such map in server's NIS domain
No submitmasters was specified and no NIS netgroup called pbsubmitmasters found
PowerBroker for Unix & Linux needs to know the submitmasters(s) to work.
The PowerBroker for Unix & Linux programs need to know which Policy Server Host(s) you have
decided to allow to act as submitmaster(s) for this machine.
Submitmasters take requests for secured tasks from Submit Hosts,
accept or reject them, and pass the accepted requests to a Run Host.
To locate submitmasters, programs look for a setting in the settings file
containing the names of the submitmaster machines or a netgroup
called pbsubmitmasters.

```

```

Enter Policy Server list (submitmasters): hp113-ca025-012.unix.symark.com
ypcat: no such map in server's NIS domain
No acceptmasters was specified and no NIS netgroup called pbacceptmasters found
PowerBroker for Unix & Linux needs to know the acceptmasters(s) to work.

```

```

The PowerBroker for Unix & Linux programs need to know which Policy Server Host(s) you have
decided to allow to request execution of secured tasks to this machine.
Hosts on the acceptmasters list are the Policy Server Hosts which are allowed
to make secured task requests to this machine.

```

```

To do this, programs look for a setting in the settings file containing the
names of the acceptmasters machines or a netgroup called pbacceptmasters.

```

```

Enter Incoming Policy Server list (acceptmasters): hp113-ca025-012.unix.symark.com
ypcat: no such map in server's NIS domain
No log hosts was specified and no NIS netgroup called pblogservers found
PowerBroker for Unix & Linux needs to know the log hosts(s) to work.

```

The PowerBroker for Unix & Linux programs need to know which machine(s) you have selected as Log Host(s). Log Hosts are hosts which Policy Servers select for Run Hosts to do event and io logging.

To do this, pbmasterd looks for the setting logservers in the settings file. This setting contains the names of the Log Host machines or a netgroup.

Current installation settings for Log Server(s):

Enter Log Server list (logservers): hp113-ca025-012.unix.symark.com

Generating key file /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/settings_files/pb.key...

Are all the installation settings correct [yes]?

Generating config file /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/settings_files/pb.cfg

Creating the settings file creation script

Backed up existing settings file creation script to:

'/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/pbcreatesettingsfile.ctime.May_26_15:05'

Running settings file creation script

Creating settings file /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/settings_files/pb.settings

Generated settings files are in directory: /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/settings_files

PowerBroker for Unix & Linux Settings File Generation completed successfully.

Creating the PowerBroker for Unix & Linux Configuration Package Using pbcreatehpuxcfgpkg

This section shows the creation of the PowerBroker for Unix & Linux configuration depot using the pbcreatehpuxcfgpkg program with the -p and -s options.

Note: At the end of its output, the pbcreatehpuxcfgpkg script shows which PowerBroker for Unix & Linux component filesets need to be copied to the SD depot.

```
# cd /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install
# ./pbcreatehpuxcfgpkg -p CLIENT1 -s /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/settings_files
pbcreatehpuxcfgpkg: starting from /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install
pbcreatehpuxcfgpkg: keyfile pb.key will be included in package
pbcreatehpuxcfgpkg: reading /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/settings_files/pb.cfg
pbcreatehpuxcfgpkg: processing, please wait . . .

pbcreatehpuxcfgpkg: packaging PowerBroker Unix/Linux Configuration HP-UX Depot . . .
===== 05/26/17 15:19:42 PDT BEGIN swpackage SESSION
* Session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com".

* Source:
```

```
pbul-qa-hpux11v3-01.unix.symark.com:/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg/psf/PowerBroker-Cfg.psf
```

* Target:

```
pbul-qa-hpux11v3-01.unix.symark.com:/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg/depot/PowerBroker-Cfg-9.4.3.18.CLIENT1.depot
```

* Software selections:

*

* Beginning Selection Phase.

* Reading the Product Specification File (PSF)

```
"/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg/psf/PowerBroker-Cfg.psf".
```

* Reading the product "PowerBroker-Cfg" at line 11.

* Reading the fileset "CLIENT1" at line 48.

NOTE: The temporary target depot "/var/tmp/pkgAAA005165" has been created.

* Selection Phase succeeded.

* Beginning Analysis Phase.

NOTE: The fileset "PowerBroker-Cfg.CLIENT1" has a prerequisite dependency on a software object which exists in another product, "PowerBroker-hppa64.RUNHOST", which was not selected for packaging and does not exist in the target depot.

NOTE: The fileset "PowerBroker-Cfg.CLIENT1" has a prerequisite dependency on a software object which exists in another product, "PowerBroker-hpia64.RUNHOST", which was not selected for packaging and does not exist in the target depot.

NOTE: The fileset "PowerBroker-Cfg.CLIENT1" has a prerequisite dependency on a software object which exists in another product, "PowerBroker-hppa64.SUBMITHOST", which was not selected for packaging and does not exist in the target depot.

NOTE: The fileset "PowerBroker-Cfg.CLIENT1" has a prerequisite dependency on a software object which exists in another product, "PowerBroker-hpia64.SUBMITHOST", which was not selected for packaging and does not exist in the target depot.

NOTE: The fileset "PowerBroker-Cfg.CLIENT1" has a prerequisite dependency on a software object which exists in another product, "PowerBroker-hppa64.SHAREDLIBS", which was not selected for packaging and does not exist in the target depot.

NOTE: The fileset "PowerBroker-Cfg.CLIENT1" has a prerequisite dependency on a software object which exists in another product, "PowerBroker-hpia64.SHAREDLIBS", which was not selected for packaging and does not exist in the target depot.

NOTE: One or more of the filesets you selected specify a dependency on software which exists in another product. (See above).

The other software was not selected for packaging and does not exist in the target depot. (An unresolved dependency on another product may prevent the dependent product from being

```

installed.)
* Analysis Phase succeeded.
* Beginning Package Phase.
* Packaging the product "PowerBroker-Cfg".
* Packaging the fileset "PowerBroker-Cfg.CLIENT1".
* Package Phase succeeded.
* Beginning Tapemaker Phase.
* Copying the temporary depot to the tape
"/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg/depot/PowerBroker-Cfg-
9.4.3.18.CLIENT1.depot".

* Calculating the tape blocks required to copy the temporary
depot to the tape
"/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg/depot/PowerBroker-Cfg-
9.4.3.18.CLIENT1.depot".

NOTE: The temporary depot requires 220 Kbytes on the tape
"/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg/depot/PowerBroker-Cfg-
9.4.3.18.CLIENT1.depot".

* Writing the tape
"/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg/depot/PowerBroker-Cfg-
9.4.3.18.CLIENT1.depot"
(tape 1 of 1).
* Writing the fileset "PowerBroker-Cfg.CLIENT1" (1 of 1)
* Tape #1: CRC-32 checksum & size: 2376197741 225280
* Removing the temporary depot.
* Tapemaker Phase succeeded.
===== 05/26/17 15:19:42 PDT END swpackage SESSION
pbcreatehpuxcfpgk: depot 'PowerBroker-Cfg-9.4.3.18.CLIENT1.depot' placed in
/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install

pbcreatehpuxcfpgk: the following depot filesets will need to be loaded to the target system:
PowerBroker-{arch}.RUNHOST PowerBroker-{arch}.SUBMITHOST PowerBroker-{arch}.SHAREDLIBS
where {arch} is the appropriate architecture for the target system, 'hppa64' or 'ia64'.

pbcreatehpuxcfpgk: completed.

```

Copying the PowerBroker for Unix & Linux Depots Using the swcopy Command

This section shows the execution of the `swcopy` command to copy the PowerBroker component and configuration depots to the default SD depot. This section also includes execution of the `swjob` and `swlist` commands to verify that the depots have been copied.

```

# swcopy -s /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/package/PowerBroker-hpia64-
9.4.3.18.depot PowerBroker-hpia64.SHAREDLIBS PowerBroker-hpia64.SUBMITHOST PowerBroker-hpia64.RUNHOST
===== 05/26/17 16:47:14 PDT BEGIN swcopy SESSION (non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0263)

```

```

* Session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com".

* Beginning Selection
* "pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw": This
target does not exist and will be created.
* Source:
/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/package/PowerBroker-hpia64-9.4.3.18.depot

* Targets:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw
* Software selections:
PowerBroker-hpia64.RUNHOST,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
PowerBroker-hpia64.SHAREDLIBS,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
PowerBroker-hpia64.SUBMITHOST,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
* Selection succeeded.

* Beginning Analysis and Execution
* Session selections have been saved in the file
"/.sw/sessions/swcopy.last".
* The analysis phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
* The execution phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
* Analysis and Execution succeeded.

NOTE: More information may be found in the agent logfile using the
command "swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0263
@ pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
===== 05/26/17 16:47:21 PDT END swcopy SESSION (non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0263)
# swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0263 @ pbul-qa-hpux11v3-
01.unix.symark.com:/var/spool/sw
===== 05/26/17 16:47:15 PDT BEGIN copy AGENT SESSION (pid=7319)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0263)
* Agent session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com". (pid=7319)
* Beginning Analysis Phase.
* Source:
pbul-qa-hpux11v3-01.unix.symark.com:/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-
18/package/PowerBroker-hpia64-9.4.3.18.depot

* Target:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw
* Target logfile:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw/swagent.log

```



```

* Reading source for product information.
* Reading source for file information.
NOTE: The used disk space on filesystem "/var" is estimated to
increase by 91664 Kbytes.
This will leave 5407144 Kbytes of available user disk space
after the installation.
* Summary of Analysis Phase:
* 3 of 3 filesets had no Errors or Warnings.
* The Analysis Phase succeeded.
* Beginning the Copy Execution Phase.
* Filesets: 3
* Files: 105
* Kbytes: 90877
* Copying fileset "PowerBroker-hpia64.RUNHOST,r=9.4.3.18" (1 of
3).
* Copying fileset "PowerBroker-hpia64.SHAREDLIBS,r=9.4.3.18" (2
of 3).
* Copying fileset "PowerBroker-hpia64.SUBMITHOST,r=9.4.3.18" (3
of 3).
* Summary of Execution Phase:
* 3 of 3 filesets had no Errors or Warnings.
* The Execution Phase succeeded.
===== 05/26/17 16:47:21 PDT END copy AGENT SESSION (pid=7319)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0263)
# swcopy -s /opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg-
9.4.3.18.CLIENT1.depot PowerBroker-Cfg.CLIENT1
===== 05/26/17 16:49:48 PDT BEGIN swcopy SESSION (non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0264)
* Session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com".

* Beginning Selection
* Target connection succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
* Source:
/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-18/install/PowerBroker-Cfg-9.4.3.18.CLIENT1.depot

* Targets:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw
* Software selections:
PowerBroker-Cfg.CLIENT1,r=9.4.3.18,a=HP-UX_B.11.11/23/31_32/64_IA/PA,v=BeyondTrust
* Selection succeeded.

* Beginning Analysis and Execution
* Session selections have been saved in the file
"/.sw/sessions/swcopy.last".
* The analysis phase succeeded for

```

```

"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
* The execution phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
* Analysis and Execution succeeded.

NOTE: More information may be found in the agent logfile using the
command "swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0264
@ pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
===== 05/26/17 16:49:48 PDT END swcopy SESSION (non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0264)
# swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0264 @ pbul-qa-hpux11v3-
01.unix.symark.com:/var/spool/sw
===== 05/26/17 16:49:48 PDT BEGIN copy AGENT SESSION (pid=7373)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0264)
* Agent session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com". (pid=7373)
* Beginning Analysis Phase.
* Source:
pbul-qa-hpux11v3-01.unix.symark.com:/opt/pbpkg/powerbroker/v9.4/pbul_hpux.ia64_9.4.3-
18/install/PowerBroker-Cfg-9.4.3.18.CLIENT1.depot

* Target:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw
* Target logfile:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw/swagent.log
* Reading source for product information.
* Reading source for file information.
NOTE: The used disk space on filesystem "/var" is estimated to
increase by 232 Kbytes.
This will leave 5446360 Kbytes of available user disk space
after the installation.
* Summary of Analysis Phase:
* 1 of 1 filesets had no Errors or Warnings.
* The Analysis Phase succeeded.
* Beginning the Copy Execution Phase.
* Filesets: 1
* Files: 6
* Kbytes: 186
* Copying fileset "PowerBroker-Cfg.CLIENT1,r=9.4.3.18" (1 of 1).
* Summary of Execution Phase:
* 1 of 1 filesets had no Errors or Warnings.
* The Execution Phase succeeded.
===== 05/26/17 16:49:48 PDT END copy AGENT SESSION (pid=7373)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0264)

```

Installing the PowerBroker for Unix & Linux Filesets Using the swinstall Command

This section shows the execution of the HP-UX `swinstall` command to install the PowerBroker for Unix & Linux filesets. Because the `swinstall` command automatically installs the dependent filesets, you need only run the `swinstall` command for the configuration fileset. Following installation of the configuration package, the installation is verified by submitting the `swlist`, `swjob`, and `swverify` commands. Finally, the `id` command is submitted to PowerBroker for Unix & Linux to test the installation.

Note: During the PowerBroker for Unix & Linux fileset installation process, you might see a warning message regarding “core transition links.” You can ignore this warning.

```
# swinstall PowerBroker-Cfg.CLIENT1
===== 05/26/17 16:50:39 PDT BEGIN swinstall SESSION
(non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0265)
* Session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com".

* Beginning Selection
* Target connection succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:".
* Source connection succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
* Source: /var/spool/sw
* Targets: pbul-qa-hpux11v3-01.unix.symark.com:/
* Software selections:
PowerBroker-Cfg.CLIENT1, r=9.4.3.18, a=HP-UX_B.11.11/23/31_32/64_IA/PA, v=BeyondTrust
+ PowerBroker-hpia64.RUNHOST, r=9.4.3.18, a=HP-UX_B.11.23/31_64_IA, v=BeyondTrust
+ PowerBroker-hpia64.SHAREDLIBS, r=9.4.3.18, a=HP-UX_B.11.23/31_64_IA, v=BeyondTrust
+ PowerBroker-hpia64.SUBMITHOST, r=9.4.3.18, a=HP-UX_B.11.23/31_64_IA, v=BeyondTrust
* A "+" indicates an automatic selection due to dependency or
the automatic selection of a patch or reference bundle.
* Selection succeeded.

* Beginning Analysis and Execution
* Session selections have been saved in the file
"/.sw/sessions/swinstall.last".
* The analysis phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:".
* The execution phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:".
* Analysis and Execution succeeded.

NOTE: More information may be found in the agent logfile using the
command "swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0265
@ pbul-qa-hpux11v3-01.unix.symark.com:".
===== 05/26/17 16:50:54 PDT END swinstall SESSION (non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0265)
```

```
# swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0265 @ pbul-qa-hpux11v3-01.unix.symark.com:/
===== 05/26/17 16:50:39 PDT BEGIN install AGENT SESSION (pid=7464)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0265)
* Agent session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com". (pid=7464)
* Beginning Analysis Phase.
* Source:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw
* Target: pbul-qa-hpux11v3-01.unix.symark.com:/
* Target logfile:
pbul-qa-hpux11v3-01.unix.symark.com:/var/adm/sw/swagent.log
* Reading source for product information.
* Reading source for file information.
* Executing preDSA command.
NOTE: The used disk space on filesystem "/" is estimated to increase
by 24 Kbytes.
This will leave 205712 Kbytes of available user disk space
after the installation.
NOTE: The used disk space on filesystem "/opt" is estimated to
increase by 32 Kbytes.
This will leave 2466280 Kbytes of available user disk space
after the installation.
NOTE: The used disk space on filesystem "/usr" is estimated to
increase by 91552 Kbytes.
This will leave 3519968 Kbytes of available user disk space
after the installation.
NOTE: The used disk space on filesystem "/var" is estimated to
increase by 288 Kbytes.
This will leave 5410848 Kbytes of available user disk space
after the installation.
* Summary of Analysis Phase:
* 4 of 4 filesets had no Errors or Warnings.
* The Analysis Phase succeeded.
* Beginning the Install Execution Phase.
* Filesets: 4
* Files: 111
* Kbytes: 91063
* Installing fileset "PowerBroker-hpia64.SUBMITHOST,r=9.4.3.18"
because one or more other selected filesets depend on it (1 of
4).
* Installing fileset "PowerBroker-hpia64.SHAREDLIBS,r=9.4.3.18"
because one or more other selected filesets depend on it (2 of
4).
* Installing fileset "PowerBroker-hpia64.RUNHOST,r=9.4.3.18"
because one or more other selected filesets depend on it (3 of
4).
* Installing fileset "PowerBroker-Cfg.CLIENT1,r=9.4.3.18" (4 of
4).
```

```

* Beginning the Configure Execution Phase.
NOTE: Reading pb.cfg...
NOTE: Looking for SuperDaemons to configure...
NOTE: Finished looking for SuperDaemons to configure...
NOTE: Removing PowerBroker service definitions (if any)
from /etc/services.
NOTE: Adding PowerBroker service definitions to /etc/services
NOTE: Removing any PowerBroker definitions from SuperDaemon
inetd file /etc/inetd.conf
NOTE: Adding PowerBroker definitions to SuperDaemon configurations
/etc/inetd.conf
NOTE: Reloading SuperDaemon Configurations...
NOTE: Done Reloading SuperDaemon Configurations...
Updating Settings in database (if any)...
* Summary of Execution Phase:
* 4 of 4 filesets had no Errors or Warnings.
* The Execution Phase succeeded.
===== 05/26/17 16:50:54 PDT END install AGENT SESSION (pid=7464)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0265)

# swlist PowerBroker\*
# Initializing...
# Contacting target "pbul-qa-hpux11v3-01.unix.symark.com"...
#
# Target: pbul-qa-hpux11v3-01.unix.symark.com:/
#
# PowerBroker-Cfg 9.4.3.18 BeyondTrust PowerBroker Unix/Linux - Root Delegation and Privilege Management
PowerBroker-Cfg.CLIENT1 9.4.3.18 BeyondTrust PowerBroker Unix/Linux Configuration - Root Delegation and
Privilege Management
# PowerBroker-hpia64 9.4.3.18 BeyondTrust PowerBroker - Root Delegation and Privilege Management
PowerBroker-hpia64.RUNHOST 9.4.3.18 BeyondTrust PowerBroker Run Host - Root Delegation and Privilege
Management
PowerBroker-hpia64.SHAREDLIBS 9.4.3.18 BeyondTrust PowerBroker Shared Libraries - Root Delegation and
Privilege Management
PowerBroker-hpia64.SUBMITHOST 9.4.3.18 BeyondTrust PowerBroker Submit Host - Root Delegation and
Privilege Management
# swverify PowerBroker-Cfg PowerBroker-hpia64
===== 05/26/17 16:52:13 PDT BEGIN swverify SESSION
(non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0266)
* Session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com".

* Beginning Selection
* Target connection succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:".
* Software selections:
PowerBroker-Cfg.CLIENT1,l=/,r=9.4.3.18,a=HP-UX_B.11.11/23/31_32/64_IA/PA,v=BeyondTrust
PowerBroker-hpia64.RUNHOST,l=/,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust

```

```
PowerBroker-hpia64.SHAREDLIBS,l=/,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
PowerBroker-hpia64.SUBMITHOST,l=/,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
* Selection succeeded.
```

```
* Beginning Analysis
* Session selections have been saved in the file
"/.sw/sessions/swverify.last".
* The analysis phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:".
* Verification succeeded.
```

```
NOTE: More information may be found in the agent logfile using the
command "swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0266
@ pbul-qa-hpux11v3-01.unix.symark.com:".
===== 05/26/17 16:52:17 PDT END swverify SESSION (non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0266)
# swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0266 @ pbul-qa-hpux11v3-01.unix.symark.com:/
===== 05/26/17 16:52:14 PDT BEGIN verify AGENT SESSION (pid=7787)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0266)
* Agent session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com". (pid=7787)
* Beginning Analysis Phase.
* Target: pbul-qa-hpux11v3-01.unix.symark.com:/
* Target logfile:
pbul-qa-hpux11v3-01.unix.symark.com:/var/adm/sw/swagent.log
* Reading source for file information.
* Configured PowerBroker-hpia64.SUBMITHOST,l=/,r=9.4.3.18
* Configured PowerBroker-hpia64.SHAREDLIBS,l=/,r=9.4.3.18
* Configured PowerBroker-hpia64.RUNHOST,l=/,r=9.4.3.18
* Configured PowerBroker-Cfg.CLIENT1,l=/,r=9.4.3.18
* Summary of Analysis Phase:
Verified PowerBroker-hpia64.SUBMITHOST,l=/,r=9.4.3.18
Verified PowerBroker-hpia64.SHAREDLIBS,l=/,r=9.4.3.18
Verified PowerBroker-hpia64.RUNHOST,l=/,r=9.4.3.18
Verified PowerBroker-Cfg.CLIENT1,l=/,r=9.4.3.18
* 4 of 4 filesets had no Errors or Warnings.
* The Analysis Phase succeeded.
===== 05/26/17 16:52:17 PDT END verify AGENT SESSION (pid=7787)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0266)
```

Sample of the Uninstall Process from a Package Installation

This section shows the execution of the HP-UX `swremove` utility to remove the PowerBroker for Unix & Linux depots. First, `swremove` is used to uninstall PowerBroker for Unix & Linux software from the host. Then, `swremove` is used to remove the PowerBroker for Unix & Linux depots from the SD depot.

```
# swremove PowerBroker-Cfg PowerBroker-hpia64
```

```

===== 05/27/17 09:54:20 PDT BEGIN swremove SESSION
(non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0267)
* Session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com".

* Beginning Selection
* Target connection succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/".
* Software selections:
PowerBroker-Cfg.CLIENT1,l=/,r=9.4.3.18,a=HP-UX_B.11.11/23/31_32/64_IA/PA,v=BeyondTrust
PowerBroker-hpia64.RUNHOST,l=/,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
PowerBroker-hpia64.SHAREDLIBS,l=/,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
PowerBroker-hpia64.SUBMITHOST,l=/,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
* Selection succeeded.

* Beginning Analysis
* Session selections have been saved in the file
"/.sw/sessions/swremove.last".
* The analysis phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/".
* Analysis succeeded.

* Beginning Execution
* The execution phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/".
* Execution succeeded.

NOTE: More information may be found in the agent logfile using the
command "swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0267
@ pbul-qa-hpux11v3-01.unix.symark.com:/".
===== 05/27/17 09:54:26 PDT END swremove SESSION (non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0267)
# swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0267 @ pbul-qa-hpux11v3-01.unix.symark.com:/
===== 05/27/17 09:54:20 PDT BEGIN remove AGENT SESSION (pid=16901)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0267)
* Agent session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com". (pid=16901)
* Beginning Analysis Phase.
* Target: pbul-qa-hpux11v3-01.unix.symark.com:/
* Target logfile:
pbul-qa-hpux11v3-01.unix.symark.com:/var/adm/sw/swagent.log
* Reading source for file information.
* Summary of Analysis Phase:
* 4 of 4 filesets had no Errors or Warnings.
* The Analysis Phase succeeded.
* Beginning the Unconfigure Execution Phase.
* Filesets: 4

```

```

* Files: 111
* Kbytes: 91063
NOTE: Reading pb.cfg...
NOTE: Looking for SuperDaemons to configure...
NOTE: Finished looking for SuperDaemons to configure...
NOTE: Removing PowerBroker service definitions (if any)
from /etc/services.
NOTE: Removing any PowerBroker definitions from SuperDaemon
inetd file /etc/inetd.conf
NOTE: Reloading SuperDaemon Configurations...
NOTE: Done Reloading SuperDaemon Configurations...
* Beginning the Remove Execution Phase.
* Removing fileset "PowerBroker-Cfg.CLIENT1,l=/,r=9.4.3.18" (1
of 4).
* Removing fileset "PowerBroker-hpia64.RUNHOST,l=/,r=9.4.3.18"
(2 of 4).
Removing /opt/pbul/scripts
* Removing fileset
"PowerBroker-hpia64.SHAREDLIBS,l=/,r=9.4.3.18" (3 of 4).
* Removing fileset
"PowerBroker-hpia64.SUBMITHOST,l=/,r=9.4.3.18" (4 of 4).
* Summary of Execution Phase:
* 4 of 4 filesets had no Errors or Warnings.
* The Execution Phase succeeded.
===== 05/27/17 09:54:26 PDT END remove AGENT SESSION (pid=16901)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0267)
# swremove -d PowerBroker-Cfg PowerBroker-hpia64
===== 05/27/17 09:56:54 PDT BEGIN swremove SESSION
(non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0268)
* Session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com".

* Beginning Selection
* Target connection succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".
* Software selections:
PowerBroker-Cfg.CLIENT1,r=9.4.3.18,a=HP-UX_B.11.11/23/31_32/64_IA/PA,v=BeyondTrust
PowerBroker-hpia64.RUNHOST,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
PowerBroker-hpia64.SHAREDLIBS,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
PowerBroker-hpia64.SUBMITHOST,r=9.4.3.18,a=HP-UX_B.11.23/31_64_IA,v=BeyondTrust
* Selection succeeded.

* Beginning Analysis
* Session selections have been saved in the file
"/.sw/sessions/swremove.last".
* The analysis phase succeeded for
"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".

```


* Analysis succeeded.

* Beginning Execution

* The execution phase succeeded for

"pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".

* Execution succeeded.

NOTE: More information may be found in the agent logfile using the command "swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0268 @ pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw".

```

===== 05/27/17 09:56:54 PDT END swremove SESSION (non-interactive)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0268)
# swjob -a log pbul-qa-hpux11v3-01.unix.symark.com-0268 @ pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw
===== 05/27/17 09:56:54 PDT BEGIN remove AGENT SESSION (pid=17066)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0268)
* Agent session started for user
"root@pbul-qa-hpux11v3-01.unix.symark.com". (pid=17066)
* Beginning Analysis Phase.
* Target:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw
* Target logfile:
pbul-qa-hpux11v3-01.unix.symark.com:/var/spool/sw/swagent.log
* Reading source for file information.
* Summary of Analysis Phase:
* 4 of 4 filesets had no Errors or Warnings.
* The Analysis Phase succeeded.
* Beginning the Remove Execution Phase.
* Filesets: 4
* Files: 111
* Kbytes: 91063
* Removing fileset "PowerBroker-Cfg.CLIENT1,r=9.4.3.18" (1 of 4).
* Removing fileset "PowerBroker-hpia64.RUNHOST,r=9.4.3.18" (2 of 4).
* Removing fileset "PowerBroker-hpia64.SHAREDLIBS,r=9.4.3.18" (3 of 4).
* Removing fileset "PowerBroker-hpia64.SUBMITHOST,r=9.4.3.18" (4 of 4).
* Summary of Execution Phase:
* 4 of 4 filesets had no Errors or Warnings.
* The Execution Phase succeeded.
===== 05/27/17 09:56:54 PDT END remove AGENT SESSION (pid=17066)
(jobid=pbul-qa-hpux11v3-01.unix.symark.com-0268)

```

Installing Multiple Copies

Beginning with the release of PowerBroker for Unix & Linux V3.0, it is possible to install multiple concurrent PowerBroker for Unix & Linux copies on the same machine. To install multiple copies, each copy must be a logically distinct installation. This type of installation is performed by using an installation prefix and/or suffix. Installing multiple, concurrent copies of PowerBroker for Unix & Linux affects the following:

- `pbinstall` and `pbuninstall`
- Remote installation using `pbmakeremotetar`
- Program names and execution
- Service names and port numbers
- NIS(+) netgroups
- PowerBroker for Unix & Linux settings file
- `root` policy file name
- Policy file contents
- Key file name
- Log file names

For information about prefixed and suffixed installations, see [Prefix and Suffix Installations](#).

Remote Installation Using `pbmakeremotetar` with Prefixes and Suffixes

To make a remote tar archive using `pbmakeremotetar` for a prefixed installation, specify the prefix and/or suffix on the `pbmakeremotetar` command line with the `-p` and `-s` switches (as appropriate). The tar file name that is specified on the command line should be unique to avoid overwriting an existing tar archive.

Program Names and Execution

All program names are prefixed in a prefixed installation. `pbmasterd` is `{prefix}pbmasterd`, `pbrun` is `{prefix}pbrun`, and so forth. For example, if the prefix is `test`, `pbrun` is executed as follows:

```
$ testpbrun date
```

Suffixes are implemented in the same way.

Service Names and Port Numbers

All PowerBroker for Unix & Linux service names are prefixed and/or suffixed. For example, using a prefix of `test`, the service name for `pbmasterd` is `testpbmasterd`. The entries are added to `/etc/services` by `pbinstall`.

PowerBroker for Unix & Linux service names and port numbers (whether prefixed and/or suffixed) must be added manually to the NIS database on the NIS Policy Server.

When installing prefixed (and/or suffixed) installations of PowerBroker for Unix & Linux on a host with other PowerBroker for Unix & Linux installations, unique port numbers must be assigned for each installation. The installers do not check for unique port numbers and specifying overlapping ports may cause PowerBroker for Unix & Linux to function incorrectly.

NIS(+) Netgroup Names

All PowerBroker for Unix & Linux netgroup names (for example, `pbacceptmasters`) are prefixed (for example, `{prefix}pbacceptmasters`). Suffixes are added to the end of PowerBroker for Unix & Linux netgroup names.

Settings File

The `pb.settings` file name is prefixed with the prefix (for example, `/etc/{prefix}pb.settings`). Suffixes are added to the end of the filename. The installer work file name, `pb.cfg`, is also prefixed or suffixed.

root Policy Filename

The default root policy file name's basename is prefixed like any other PowerBroker for Unix & Linux component: `{prefix}pb.conf`. This enables the prefixed installation to have a policy file set that is separate from any other PowerBroker for Unix & Linux installation on the system. Suffixes are appended to the policy file name.

Policy File Contents

Client names (`pbrun`, `pbguid`, and `pbsguid`) are prefixed and/or suffixed like any other PowerBroker for Unix & Linux program. This means that any policy that checks for any of these clients must also take prefixes and/or suffixes into account.

If any PowerBroker for Unix & Linux programs are requested from the policy (that is, `pbrun` or `pbcall`), then the references to these programs must also be prefixed and/or suffixed. If the prefix/suffix is not specified, the default (unprefixed) installation of PowerBroker for Unix & Linux is used for the called `pbrun`—most likely with unintended results.

Policy subfiles may or may not be prefixed, depending on the needs of the installation.

Key File Name

The default key file name's basename is prefixed or suffixed like any other PowerBroker for Unix & Linux component: `{prefix}pb.key{suffix}`. This enables the prefixed/suffixed installation to have its own encryption key and be logically separate from any other PowerBroker for Unix & Linux installation on the system. If a different key file is specified in the `{prefix}pb.settings{suffix}` file and the `{prefix}pb.settings{suffix}` file is encrypted, then the default named `{prefix}pb.key{suffix}` must exist and is used to decrypt the `{prefix}pb.settings{suffix}` file.

Log File Names

For event logs, the default event log file name for a prefixed installation is `{prefix}pb.eventlog`. Event log files are prefixed and suffixed by default in the same way that the executable files are, unless the file names are overridden in the policy or the `pb.settings` file.

For error logs, the default error log for the PowerBroker for Unix & Linux daemons is `{prefix}{daemonname}.log`. Suffixes are placed before the `.log` part of the file name for daemon error log files.

I/O logs are not prefixed or suffixed unless specified in the policy. I/O logs have no default name. The name of these files must be explicitly set in the policy.

Man Pages

If man pages are installed in a prefixed and/or suffixed installation, then the man page file names have the prefix or suffix added to the file name, using the format: `{prefix}pbrun{suffix}.1` where 1 is the section number of the man page. The text in the man page is not changed to reflect the prefix and/or suffix. In this example format, the displayed man page shows the command as `pbrun` regardless of the prefix or suffix in use.

Sample Policy Files

The sample policy files are not renamed with a prefix or suffix, but the directory that they are stored in is changed to reflect the prefix or suffix. For instance, with a prefix of `test`, the default location for the sample policy files on Linux is `/usr/local/lib/testpbbuilder`.

Installation Verification

After you install PowerBroker for Unix & Linux, you should use the `pbbench` utility to identify any PowerBroker for Unix & Linux configuration, file permission, and network problems. `pbbench` reads and verifies the settings in the PowerBroker for Unix & Linux configuration file on the machine on which it is run. The `pbbench` utility uses system information, such as that found in `/etc/services` and `/etc/hosts` and/or NIS, to verify the information in the settings file.

The `pbbench` output can consist of informational, warning, and error messages. By default, this output appears on the monitor. It can be redirected to a file other than standard error using the command:

```
pbbench > filename
```

Example:

```
pbbench > pbbench.output
```

Only root users can run `pbbench` because it is treated as an administration program. By default, `pbbench` is installed in the `/usr/local/bin` directory.

To verify an installation using `pbbench`, do the following:

1. Start `pbbench` by executing the following command:

```
pbbench -V
```
2. If the utility does not report any warnings or errors, then the installation is complete. If `pbbench` returns warnings or errors, then identify and correct the problems, and rerun `pbbench`. Repeat this process until there are no problems.

Note: An error will inhibit PowerBroker for Unix & Linux functionality, but a warning may or may not. All errors must be corrected, but it is only necessary to correct those warnings that affect PowerBroker for Unix & Linux operation.

3. If you are unable to correct the problems, contact BeyondTrust Technical Support.

For more detailed information regarding `pbbench`, see “PowerBroker for Unix & Linux Administration Programs” in the *PowerBroker for Unix & Linux System Administration Guide*.

Installation Environment Variables

PowerBroker for Unix & Linux uses several environment variables to direct and modify the execution of programs and scripts. The following table describes these variables.

Table 14. Installation Environment Variables

Variable	Description
COLUMNS	Specifies the width, in characters, of the current screen. This is used internally by the installation suite to request ps(1) (on most systems) to give more output on a line for the ps -ef command when determining current system state.
EDITOR	If the VISUAL environment variable is not set and this environment variable is set, then the specified editor becomes the default for editing files during the installation.
LINES	Specifies the number of lines on a page (screen). This variable is used on some systems by some programs to determine how many lines can be output.
PAGER	Specifies the page-viewing program. Use more, less, pg, or administrator-specified programs.
PATH	Specifies the locations of utilities such as awk, fgrep, grep, gunzip, sed, tar, uncompress, wc, and other Unix/Linux commands that are used by the installer.
ROWS	Used on some systems in place of the LINES environment variable. This usage is system dependant.
SHELL	Specifies the shell that is used by sub-shells of the installer. If specified, this should be /bin/sh or /usr/bin/sh as appropriate for your system.
SY_InstallBackupLimit	This variable limits the number of backup copies (the *.sybak.#### files) of a given file that are allowed on a system for a given original file. The value of this environment variable must be an integer greater than or equal to 4. The default value (if this variable is not defined) is 10. The minimum value of 4 is necessary for internal installation suite processing. Some files (notably /etc/services) undergo two phases of processing. The first phase ensures that no entries that are relevant to the new installation exist in the file. The second adds any entries that are required by the new installation.
SY_InstallPageSize	Specifies the page size for the pbinstall menu page.
SY_InstallPageWidth	Specifies the page width for the pbinstall menu page.
TMPDIR	In addition to its traditional Unix/Linux usage, this variable specifies the directory of the .cfg* files that are produced (and read) by the installation suite. Files that are saved by pbuninstall can be saved in this directory. Temporary files that are created by the installation suite can also be created in this directory. The default value is /tmp. For some systems, and some sites which periodically clean out /tmp, this is an undesirable location for the installer files if an uninstallation or re-installation is being performed after these files have been removed. Debian appears to, by design, clean /tmp when it boots. It is a good idea to point TMPDIR somewhere else (such as

	/opt/beyondtrust/pb/TMPDIR, after it is created) for these systems during the installation, pbmakeremotetar, and uninstallation processes.
VISUAL	Specifies the visual editor to use when editing parameter files during the installation process.

Installation Programs

This section describes the PowerBroker for Unix & Linux installation programs and their options. For detailed information on PowerBroker for Unix & Linux administration-related programs, see the *PowerBroker for Unix & Linux System Administration Guide*.

pbinstall

pbinstall installs, updates, and configures all PowerBroker for Unix & Linux products. pbinstall is a menu-driven, interactive installation script. It enables the superuser installer to install, update, or reconfigure

PowerBroker for Unix & Linux as required by configuration changes or updates. pbinstall properly configures (as appropriate) /etc/services, the superdaemon configuration files (/etc/inetd.conf and/or /etc/xinetd.conf), and PowerBroker for Unix & Linux for most execution environments.

An initial screen of legal information and credits is displayed, followed by a check to determine if the VISUAL or EDITOR environment variables select the editor to use during the installation. If you have not set either of these environment variables, then you are prompted to supply the path to an editor, with vi as the default.

PowerBroker for Unix & Linux is configured by a menu system with a menu of numbered selections and lettered options. For a complete listing of the options, see [PowerBroker for Unix & Linux pbinstall Installation Menu](#).

- To select an item to configure, type the number of that item and press ENTER to display the configuration prompts.
- To navigate the menu pages, use the following commands:
 - C Continue installation
 - N Next menu page
 - P Previous menu page
 - R Redraw menu (not shown due to space limitations)
 - X Exit script without performing any configuration
- After C is selected, the you are asked if the settings are acceptable. If you indicate that they are not, then pbinstall returns to the configuration menu.
- If the settings are acceptable, then pbinstall asks if you want to view the generated installation script.

Caution! The generated installation script contains thousands of lines of code; therefore, viewing this script is recommended for advanced users only. To view the script, type y.

- Then you are asked if the generated installation script is to be executed. If it is not to be executed, then the name of that script is displayed and pbinstall exits. Otherwise, the script is immediately executed.

Multiple command line options can be used together. During an update installation, the -m, -l, -r, -g, and -i arguments have no effect and must be explicitly changed using the PowerBroker for Unix & Linux installation menu for pbinstall.

An update installation is an installation in which the previous PowerBroker for Unix & Linux version has not been uninstalled. It uses the same installation directories as the previous installation (including the untar and unpack occurring in the same directories as the previous installation if the distribution was using FTP), and uses the existing pb.settings, pb.key, and pb.conf files. If done properly, all (or almost all) of the previous installation parameters carry forward to the new installation.

Syntax

```
pbinstall [options]
```

```
-a architecture
```

```
-b
```

```
-B baseport
```

```
-c
```

```
-d
```

```
-e
```

```
-g
```

```
-h
```

```
-i
```

```
-I
```

```
-l
```

```
-L hostname
```

```
-m
```

```
-M hostname
```

```
-o
```

```
-p prefix
```

```
-r
```

```
-s suffix
```

```
-u
```

```
-w
```

```
-x
```

```
-y hostname
```

```
-Y
```

```
-z
```

```
-u
```

```
-v
```

```
-w
```

```
-x
```

```
-z
```

```
-A
```

```
-K
```

```
-D
```

```
-P
```

```
-N
```

```
-S
```

```
-R
```

-Q
-T
-W
-X
-Z
-x

pbinstall -h
pbinstall -v

Arguments

-a architecture	<p>This option and its required argument explicitly specify which Unix/Linux architecture file to install.</p> <p>If the <code>-a</code> option is used, then the installer compares the expected flavor and the flavor that is specified with the <code>-a</code> option and displays a warning if they do not match.</p> <p>In PowerBroker for Unix & Linux V3.2 and earlier, the installation does not cross-check flavors. Beginning with PowerBroker for Unix & Linux V3.5, the installation script cross-checks flavors.</p>
-A	Set the Application Id for Client Registration
-b	Runs pbinstall in batch mode. In batch mode, the specified existing and then default settings are automatically used. User intervention is not allowed and “hit enter” prompts are suppressed. This option also invokes <code>-e</code> .
-B	Specify base daemon port number.
-c	<p>Causes pbinstall to skip the steps that process or update the PowerBroker for Unix & Linux settings file (<code>/etc/pb.settings</code>). This option is often used during the upgrade of an existing PowerBroker for Unix & Linux installation.</p> <p>The <code>/etc/pb.settings</code> file is not changed. It is backed up (to <code>/etc/pb.settings.sybak####</code>) and replaced. Therefore, the creation and/or modification dates on the file may be changed.</p>
-d	Installs the static <code>pbdemo.key</code> for a fresh install. This keyfile is static and shipped as part of the tar file. Therefore it should only be used for demo purposes and should not be used in production environment.
-D	Set the address for the Primary Policy Server for Client Registration.
-e	Runs pbinstall automatically by bypassing the menu step of pbinstall. Bypassing the pbinstall menu step makes it impossible to change installation options or configurations.
-g	Creates a log host (that is, installs <code>pblogd</code>).
-h	Prints the usage information for pbinstall and causes it to exit.
-i	Ignores previous <code>pb.settings</code> files.
-l	Install Primary License Server (infers <code>-X</code> and <code>-Y</code>)
-K	Set the Application Key for Client Registration.

-l	Creates a run host (that is, installs pblockd). Not applicable to installation of PowerBroker Express.
-L host	This option with a following word argument specifies the hostname to be used in the "logservers" in pb.settings. A list of hosts can be specified by repeating the -L argument followed by the host: -L host1 -L host2
-m	Creates a Policy Server host (that is, installs pbmasterd).
-M host	This option with a following word argument specifies the hostame to be used in the "acceptmasters" and "submitmasters" in pb.settings. A list of hosts can be specified by repeating the -M argument followed by the host: -M host1 -M host2
-N	Set the Registration Profile name for Client Registration.
-o	Creates a GUI host (that is, installs pbguid).
-p prefix	This option with a following word argument specifies an installation prefix for this installation.
-P	Set the port for the Primary Policy Server for Client Registration.
-Q	Install Primary Registry Name Server (infers -S, -W and -X)
-r	Creates a submit host; installs client software (pbrun, pbsh, pbksh). For installation of PowerBroker Express, installs, pbssh.
-R directory	Specifies a base directory for applicable settings in the generated pb.settings file. Used with -z option only.
-s suffix	This option with a following word argument specifies an installation suffix for this installation.
-S	Specify "y" or "n" to enable or disable Registry Name Service
-T	Install PBSUDO Policy Server
-u	Installs the PowerBroker for Unix & Linux utility programs (pbvi, pbless, and so forth).
-v	Prints pbinstall version information and exits.
-w	Creates a secure GUI host (that is, installs pbsguid).
-W	Install Registry Name Server
-y <hostname>	Specify License Server(s) with one or more -y <hostname> arguments. Note: The first host specified must be the Primary License Server
-Y	Install License Server
-x	Creates a log synchronization host (that is, installs pbsyncd).
-X	Install Client Registration Services
-z	Creates pb.settings, pb.conf, and (if applicable) pb.key files only. For use when installing PowerBroker for Unix & Linux with package installers.

	Cannot be combined with the -b, -c, -e, -i, -o, -p, -s, -u, -w, or -x options.
-Z	Install File Integrity Policy Services

Files

Not applicable

Example

For a pbinstall execution example, see “Example of a pbinstall Execution,” page 68.

See Also

pbmakeremotetar(8), pbuninstall(8), pbversion (8)

run_pbinstall

run_pbinstall is a wrapper script for pbinstall that simplifies installation of PowerBroker for Unix & Linux components, providing a smaller set of options. It is meant to be used for fresh installation where it is acceptable to use default settings.

Syntax

```
run_pbinstall [options]
-a|b|c [--L host [-L host]...] [-M host [[-M host]...] [-p prefix] [-s
suffix]
-a
-b
-c
-p prefix
-s suffix
-L hostname
-M hostname
run_pbinstall -h
```

-a	Install all components of PowerBroker for Unix & Linux. Equivalent to running “pbinstall -i -e -mgrlowux”
-b	Install server (back-end) components of PowerBroker for Unix & Linux. It creates a Policy Server host (installs pbmasterd, Log host (pblogd), GUI host (pbguid), secure GUI host (pbsguid), and log synchronization host (pbsyncd). Equivalent to running “pbinstall -i -e -mgowx”
-c	Install client components of PowerBroker for Unix & Linux. It creates a submit host (installs pbrun, pbsh, pbksh), run host (pblocald), and PowerBroker Servers utility programs (pbvi, pbless, etc). Equivalent to running “pbinstall -i -e -rul”
-p prefix	Specify PowerBroker installation prefix.
-s suffix	Specify PowerBroker installation suffix.
-L <i>hostname</i>	Specify logservers with one or more “-L <hostname>” arguments. The hostname will be used for "logservers" in pb.settings.
-M <i>hostname</i>	Specify Policy Servers with one or more “-M <hostname>” arguments. The hostname

	will be used for "acceptmasters" and "submitmasters" in pb.settings.
-h	Prints the usage information for run_pbinstall and exits.

Example

```
run_pbinstall -a
run_pbinstall -b
run_pbinstall -c
run_pbinstall -a -p adm1 -L lhost1 -L lhost2 -M mhost1
```

pbmakeremotetar

pbmakeremotetar makes a clone of a configuration for a binary and configuration-compatible target environment for PowerBroker for Unix & Linux.

pbmakeremotetar is a menu-driven, interactive installation script. It enables the superuser installer to install, update, or reconfigure PowerBroker for Unix & Linux as required by configuration changes or updates.

pbmakeremotetar properly configures (as appropriate) /etc/services, the superdaemon configuration files (/etc/inetd.conf and/or /etc/xinetd.conf), and PowerBroker for Unix & Linux for most execution environments.

pbmakeremotetar must be executed where the default directory is the directory in which

pbmakeremotetar resides or the parent directory to the directory containing pbmakeremotetar.

An initial screen appears, reminding the user about the function of pbmakeremotetar. A prompt also appears, allowing a SIGINT (CTRL+C) to abort the script.

When the script continues, it determines the switches that are necessary for tar to function as desired. A list of files to transfer to the target system is generated and presented to the user for approval and/or editing.

When the file list is accepted, a tarball file that contains the selected files is created, with the specified tarfilename and with the additional file type of.tar appended. The remote_unpack script is generated. Finally, a tarball file that contains both the first tarball file and the remote_unpack script is generated at the location that is specified by tarfilename.

After the final tarball file is created, it must be made available to the target systems. This can be done in any manner that preserves the security and binary integrity of the tarball file.

An installation work directory should be selected other than /tmp (for the same reasons as with pbinstall(8)). The tarball file should be unpacked with the following commands:

```
$ cd {installation_directory}
$ tar -xvf {tarfilename_on_local_system}
$ ./remote_unpack
```

The remote_unpack script unpacks the encapsulated tarball file into the proper locations. The script then prompts you to allow the configuration of the system (/etc/services, superdaemon configuration files). If you allow this configuration, then these configuration files are automatically modified with the appropriate superdaemons instructed to reload their databases. If you decide not to do the configuration at this time, then the name of the script to continue with the configuration is displayed and the script exits.

For Policy Server target installations, an initial installation (using pbinstall) must be done before a target remote install. Doing so ensures the proper handling of all licensing issues.

Different target system installation (working) directories should be used for different prefix and/or suffix versions of cloned installations.

Encrypted policy files are not scanned for included policy files. You must process the encrypted policy files by restoring the unencrypted ones before running `pbmakeremotetar`, or by manually moving the encrypted files.

Note: If the settings file is encrypted, then `pbmakeremotetar` will not work. An unencrypted version of the settings file must be restored before `pbmakeremotetar` can work. An encrypted policy file will not be handled properly. For details about including encrypted policy files or policy subfiles, see “`pbmakeremotetar` Installation Information,” page 74.

Syntax

```
pbmakeremotetar [options] tarfilename
```

```
-a
-b
-c
-g
-l
-m
-p prefix
-r
-s suffix
-t
-w dirspec
```

```
pbmakeremotetar -h
```

```
pbmakeremotetar -v
```

Arguments

-a	Includes all PowerBroker for Unix & Linux installation types.
-b	Runs in batch mode (no confirmation prompts).
-c	Includes submit host software for target system.
-g	Includes GUI host software for target system.
-h	Displays this usage text and exits.
-l	Includes log host software for target system.
-m	Includes Policy Server software for target system.
-p prefix	Sets the PowerBroker for Unix & Linux installation prefix.
-r	Includes run host software for target system.
-s suffix	Sets the PowerBroker for Unix & Linux installation suffix.
-t	Rebuilds off of a previously generated file name list.

-v	Displays the script version and exits.
-w dirspeg	Specifies the work directory to use when the directory containing pbmakeremotetar is read-only (for example, on a CD).
	Specifies the name of the tarball file to create (may include the full path).

Note: Both -c and -m may be specified if the current installation has both.

Files

Not applicable

Example

For a pbmakeremotetar execution example, see [Example of a pbmakeremotetar Execution](#).

See Also

pbinstall(8), pbuninstall(8), pbversion(8)

pbpatchinstall

[ver 5.1.2 and earlier] pbpatchinstall not available

[ver 5.2 and later] pbpatchinstall available

pbpatchinstall enables you to install and uninstall patches for installations that are running PowerBroker for Unix & Linux V4 and later.

Note: All PowerBroker for Unix & Linux daemons running a process during the patch installation should be stopped before using pbpatchinstall and restarted after using pbpatchinstall.

Only root can run pbpatchinstall. It must be run from the install directory where the PowerBroker for Unix & Linux patch was untarred. For example, if you untarred the PowerBroker for Unix & Linux patch from the directory:

```
/opt/beyondtrust
```

The patch install directory is then:

```
/opt/beyondtrust/powerbroker/v6.0/ pbx86_linuxA-6.0.0-16-sp1/install
```

pbpatchinstall should not be moved from this install directory because it is dependent on the included PowerBroker for Unix & Linux installer scripts (sy_install_support and pb_install_support) that are located there.

pbpatchinstall allows a PowerBroker for Unix & Linux patch to load if the patch release number differs from the PowerBroker for Unix & Linux installation release number. However, it does not allow a patch to load if the patch version does not match the PowerBroker for Unix & Linux installation major and minor version numbers. For more information on PowerBroker for Unix & Linux version numbering, see [Release Numbers](#).

pbpatchinstall does not run on PowerBroker for Unix & Linux versions earlier than V4.0 due to binary -- version argument limitations. Also, pbpatchinstall does not report the binary version for executable files pbnvi or pbuvqrg.

To uninstall a patch, go to the install directory where the patch was originally installed and execute pbpatchinstall -u. pbpatchinstall then attempts to uninstall the patch version that is defined by the install directory where pbpatchinstall resides.

For example, if you are running pbpatchinstall from the directory:

```
/opt/beyondtrust/powerbroker/v5.1/ pbx86_linuxA-5.1.2-03-sp1/install
```

pbpatchinstall attempts to uninstall the PowerBroker for Unix & Linux patch: pbx86_linuxA-5.1.2- 03-sp1 from that install directory.

If multiple patches are installed and you need to remove one or more of them, they must be removed in the reverse order from the order in which they were added.

Syntax

```
pbpatchinstall [options]
```

```
-p prefix
```

```
-s suffix
```

```
-a
```

```
-f
```

```
-u
```

```
pbpatchinstall -h
```

```
pbpatchinstall -v
```

Arguments

-a	This option and its required argument explicitly specify which Unix/Linux architecture file to install. If the <code>-a</code> option is used, then the installer compares the expected flavor and the flavor that is specified with the <code>-a</code> option and displays a warning if they do not match. In PowerBroker for Unix & Linux V3.2 and earlier, the installation does not cross-check flavors. Beginning with PowerBroker for Unix & Linux V3.5, the installation script cross-checks flavors.
-f	Forces the installation of the patch without a prompt, regardless of the release number.
-h	Displays the usage message and exits.
-p prefix	Sets the PowerBroker for Unix & Linux installation prefix.
-s suffix	Sets the PowerBroker for Unix & Linux installation suffix.
-u	Uninstalls the PowerBroker for Unix & Linux patch installation.
-v	Displays the version of pbpatchinstall and exits.

Example

```
pbpatchinstall -p test
```

creates a PowerBroker for Unix & Linux installation using the prefix test.

See Also

pbinstall(8), pbuninstall(8), pbversion(8)

pbcreateaixcfgpkg

[ver 6.1 and earlier] pbcreateaixcfgpkg not available

[ver 6.2 and later] pbcreateaixcfgpkg available

`pbcreateaixcfgpkg` creates an AIX lpp configuration package for BeyondTrust PowerBroker. `pbcreateaixcfgpkg` is a script that can be run interactively or non-interactively. The script enables a user to build a BeyondTrust PowerBroker AIX lpp configuration package, which is loaded along with one or more BeyondTrust PowerBroker AIX lpp component packages.

Unlike the PowerBroker AIX lpp component packages, which are created and distributed by BeyondTrust, AIX lpp configuration packages are created by the user. First, settings files must be created. This is accomplished by running `pbinstall` with the `-z` argument. Settings files are created by default in directory `install/settings_files`, although the user can specify the directory. The user may optionally put a policy file `pb.conf` in the `settings_files` directory to be included in the configuration package. After the settings files have been created, user runs `pbcreateaixcfgpkg` from the PowerBroker install directory. `pbcreateaixcfgpkg` accepts the following arguments:

- `-h` Help (this message) and exit.
- `-l` Save (do not delete) package build directory.
- `-p` User-specified lpp package name to be appended to `powerbroker.config`.
- `-s` Settings files directory location.
- `-v` Print version of `pbcreateaixcfgpkg` and exit.

If the `-p` and/or `-s` arguments are not supplied on the command line, the `pbcreateaixcfgpkg` script becomes interactive and prompts the user for input. The `-p` argument, user-specified package suffix, allows the user to suffix the package name with any name they wish, up to a total of 24 ASCII characters a-z, A-Z, 0-9 (including package base name `config`). For example, if the user enters `Client_Asia`, the configuration package is named `powerbroker.configClient_Asia`. If the length of the package name exceeds 24 characters, an error message is displayed, and the user is again prompted for the configuration package suffix.

The `-s` argument, settings files directory location, allows the user to specify the directory where the settings files to be included in the configuration package reside. The default value is:

```
{pbinstall_directory}/settings_files
```

If the user wishes to include other PowerBroker installations keyfiles in the configuration package, the user needs to copy the keyfiles to the settings files directory prior to building the configuration package.

If a PowerBroker Policy Server configuration package is to be built, the user can include an existing policy file `pb.conf` in the settings files directory prior to building the config- the configuration package. If a PowerBroker Policy Server configuration package is to be built, the user can include an existing policy file `pb.conf` in the settings files directory prior to building the configuration package. If `pb.conf` is not included, a new `pb.conf` is created and packaged containing the entry:

```
reject;
```

The optional `-l` argument, save (do not delete) package build directory, allows the user to build the configuration package and not remove the package build directory, which is normally done after the package is built. The created package can be found in the current (`install`) directory, and will be the package name. For example:

```
powerbroker.configClient_Asia
```

where the `-p` argument had been set to `Client_Asia`

Upon running `pbcreateaixcfgpkg`, note that the script informs the user as to which PowerBroker component packages need to be loaded on the target system. The PowerBroker configuration package will not load until the required component packages are loaded on the target system. AIX lpp packages are loaded using the `installp` command.

Syntax

```
pbcreateaixcfgpkg [options]
  -l
  -p package_suffix
  -s settings_files_directory_location
pbcreateaixcfgpkg -h
pbcreateaixcfgpkg -v
```

Arguments

-h	Print usage message and exit.
-l	Save (do not delete) package build directory.
-p suffix	User-specified lpp package name to be appended to powerbroker.config.
-s directory	Settings files directory location.
-v	Print version of pbcreateaixcfgpkg and exit.

See Also

pbinstall(8)

pbcreatehpuxcfgpkg

[ver 6.2 and earlier] pbcreatehpuxcfgpkg not available

[ver 6.2.1 and later] pbcreatehpuxcfgpkg available

pbcreatehpuxcfgpkg creates an HP-UX configuration depot for BeyondTrust PowerBroker. pbcreatehpuxcfgpkg is a script that can be run interactively or non-interactively. The script enables a user to build a BeyondTrust PowerBroker HP-UX configuration depot, which is loaded along with one or more BeyondTrust PowerBroker HP-UX component filesets.

Unlike the PowerBroker HP-UX component depot, which is created and distributed by BeyondTrust, HP-UX configuration depots are created by the user. First, settings files must be created by running pbinstall with the -z argument. Settings files are created by default in directory install/settings_files, although the user can specify the directory. The user may optionally put a policy file pb.conf in the settings_files directory to be included in the configuration package. After the settings files have been created, user runs pbcreatehpuxcfgpkg from the PowerBroker for Unix & Linux install directory. pbcreatehpuxcfgpkg accepts the following arguments:

- dSet the component fileset dependency to hppaD rather than hppaB (default)
- hHelp (this message) and exit.
- lSave (do not delete) depot build directory.
- pUser-specified name for the configuration fileset.
- sSettings files directory location.
- vPrint version of pbcreatehpuxcfgpkg and exit.

If the `-p` and/or `-s` arguments are not supplied on the command line, the `pbcreatehpuxcfgpkg` script becomes interactive and prompts you for input. The `-p` argument, user-specified fileset name, enables you to specify the configuration fileset name. The name can be between 4 and 15 ASCII characters (inclusive), and can be A-Z, 0-9, and the hyphen (-). The first character cannot be a hyphen. For example, if you specify `CLIENT-ASIA`, the configuration fileset is named `PowerBroker-Cfg[X].CLIENT-ASIA`. If the length of the fileset name is more than 15 or less than 4 characters, or if a hyphen is the first character, then an error message is displayed, and you are again prompted for the fileset name.

The `-s` argument, settings files directory location, enables you to specify the directory that contains the settings files to be included in the configuration package. The default value is:

```
<pbinstall_directory>/settings_files
```

If you want to include other PowerBroker for Unix & Linux installations keyfiles in the configuration depot, you must copy the keyfiles to the settings files directory prior to building the configuration depot.

If a PowerBroker for Unix & Linux Policy Server configuration depot is to be built, you can include an existing policy file `pb.conf` in the settings files directory prior to building the configuration depot. If `pb.conf` is not included, a new `pb.conf` is created and packaged containing the entry:

```
reject;
```

The optional `-d` argument, set component fileset dependency to 'hppaD' rather than 'hppaB' (default), enables you to generate a PowerBroker for Unix & Linux configuration depot that can be used for either `hppaD` or `ia64A` systems. If you do not use this option, then `pbcreatehpuxcfgpkg` creates a configuration depot that can be used for either `hppaB` or `ia64A` systems.

Note: If you are creating configuration depots for different flavors, use the `-p` argument to specify different fileset names for each flavor.

The optional `-l` argument, save (do not delete) depot build directory, enables you to build the configuration depot and not remove the depot build directory, which is normally removed after the depot is built. The created depot can be found in the current (install) directory, and will be the depot name. For example:

```
PowerBroker-Cfg[X]-version.CLIENT-ASIA.depot
```

where the `-p` argument had been set to `CLIENT-ASIA`.

Upon running `pbcreatehpuxcfgpkg`, note that the script informs you as to which PowerBroker for Unix & Linux component filesets need to be installed on the target system. The PowerBroker for Unix & Linux configuration package will install the required component filesets if they are not already installed, provided they have been copied into the appropriate SD depot. HP-UX depots are copied into the desired SD depot using the `swcopy` command and are installed using the `swinstall` command.

Syntax

```
pbcreatehpuxcfgpkg [options]
  -l
  -d
  -p depot_filesset_name
  -s settings_files_directory_location
pbcreatehpuxcfgpkg -h
pbcreatehpuxcfgpkg -v
```

Arguments

-d	Generate a configuration depot that has, as its dependencies, component filesets for hppaD (these component filesets can also be used on ia64A systems). Without this argument, pbcreatehpuxcfgpkg generates a configuration depot that has, as its dependencies, component filesets for hppaB (which also can be used on ia64A systems).
-h	Print usage message and exit.
-l	Save (do not delete) package build directory.
-p depot _fileset_name	User-specified name for the configuration fileset. The resulting fileset will be PowerBroker-Cfg[X].depot-fileset-name. The value of depot-fileset-name can be between 4 and 15 characters (inclusive), and allowed characters are A-Z, 0-9, and the hyphen (-); the first character cannot be a hyphen.
-s settings_files_directory _location	Settings files directory location.
-v	Print version of pbcreatehpuxcfgpkg and exit.

See Also

pbinstall(8)

pbcreatelincfgpkg

[ver 5.2 and earlier] pbcreatelincfgpkg not available

[ver 6.0 and later] pbcreatelincfgpkg available

pbcreatelincfgpkg creates a Linux RPM installation package for PowerBroker for Unix & Linux configuration and settings files. Installing this package after the required PowerBroker for Unix & Linux component packages completes the PowerBroker for Unix & Linux package installation.

If the -p option or -s option is not specified, then you are prompted to supply these values.

The output from pbcreatelincfgpkg indicates which PowerBroker for Unix & Linux component packages must be installed before the PowerBroker for Unix & Linux configuration package.

After you create the configuration package with pbcreatelincfgpkg, you install the required component packages, then install the configuration package.

Syntax

```
pbcreatelincfgpkg [options]
```

```
    -p package_suffix
```

```
    -s directory
```

```
pbcreatelincfgpkg -h
```

```
pbcreatelincfgpkg -v
```

Arguments

-h	Displays the usage message and exits.
----	---------------------------------------

-p package_suffix	Specifies a suffix of up to 18 characters to append to the configuration package name.
-s directory	Specifies the directory that contains the PowerBroker for Unix & Linux settings and configuration files to include in the package. The default value is ./settings_files.
-v	Displays the version of pbcreatelincfgpkg and exits.

Example

```
pbcreatelincfgpkg -p SBM -s /opt/beyondtrust/powerbroker/v6.0/ pbx86_linuxB-6.0.0-09/install/settings_files
```

Uses the PowerBroker for Unix & Linux settings and configuration files that are located in /opt/beyondtrust/powerbroker/v6.0/pbx86_linuxB-6.0.0-09/ install/settings_files and creates an RPM file (powerbroker-configSBM-6.0.0-09-1- noarch.rpm) in the current directory.

pbcreatesolcfgpkg

[ver 5.2 and earlier] pbcreatesolcfgpkg not available

[ver 6.0 and later] pbcreatesolcfgpkg available

pbcreatesolcfgpkg creates a Solaris installation package and corresponding package administration file for PowerBroker for Unix & Linux configuration and settings files. Installing this package after the required PowerBroker for Unix & Linux component packages completes the PowerBroker for Unix & Linux package installation.

If the -p option or -s option is not specified, then you are prompted to supply these values.

The output from pbcreatesolcfgpkg indicates which PowerBroker for Unix & Linux component packages must be installed before the PowerBroker for Unix & Linux configuration package.

After you create the configuration package with pbcreatesolcfgpkg, you install the required component packages, then install the configuration package.

Syntax

```
pbcreatesolcfgpkg [options]
```

```
-l
```

```
-p package_suffix
```

```
-s directory
```

```
pbcreatesolcfgpkg -h
```

```
pbcreatesolcfgpkg -v
```

Arguments

-h	Displays the usage message and exits.
-l	Saves (does not delete) the spooled package directory, from which the package datastream (.ds) file is created. The spooled package directory is normally deleted after the datastream file is created. Saving the spooled package directory can help BeyondTrust Technical Support to diagnose installation problems.

-p package_suffix	Specifies a suffix to append to the file names of the PowerBroker for Unix & Linux configuration package file and package admin file. This suffix can be up to 26 characters in length (3 characters for unpatched Solaris 8).
-s directory	Specifies the directory that contains the PowerBroker for Unix & Linux settings and configuration files to include in the package. The default value is ./settings_files.
-v	Displays the version of pbcreatesolcfgpkg and exits.

Example

```
pbcreatesolcfgpkg -p SBM -s /opt/beyondtrust/powerbroker/v6.0/ pbsparc_
solarisC-6.0.0-09/install/settings_files
```

Uses the PowerBroker for Unix & Linux settings and configuration files that are located in /opt/beyondtrust/powerbroker/v6.0/pbsparc_solarisC-6.0.0-09/install/settings_files and creates a datastream file (SYPBcfSBM.ds) and package admin file (SYPBcfSBM) in the current directory.

pbuninstall

pbuninstall is a menu-driven, interactive script that is used to uninstall PowerBroker for Unix & Linux. pbuninstall properly configures (as appropriate) /etc/services and the superdaemon configuration files (/etc/inetd.conf and/or /etc/xinetd.conf) for the removal of PowerBroker for Unix & Linux from most execution environments.

pbuninstall must be executed where the default directory is the directory in which pbuninstall resides, or the parent directory to the directory containing pbuninstall.

When pbuninstall is executed, you are presented with a reminder of the script's function and prompted: Hit return to continue. Using CTRL+C at this time stops the execution of the script.

Note: pbuninstall removes only those installations that are explicitly named on the command line. It must be run separately for each prefixed and suffixed installation.

During execution, the script identifies files to move to \$TMPDIR (log, policy, and configuration files), copies them to \$TMPDIR (typically /tmp) and removes them from their original location. Files to be removed are removed.

/etc/services and the superdaemon configuration files have the appropriate PowerBroker for Unix & Linux configuration lines removed. The appropriate superdaemon processes are requested to reload their configuration files.

Syntax

```
pbuninstall [options]
```

```
-a architecture
```

```
-A appid
```

```
-b
```

```
-K appkey
```

```
-p prefix
```

```
-s suffix
```

```
pbuninstall -h
```

Arguments

-a	Explicitly sets the computer architecture.
----	--

-A appid	Allow the cleanup of RNS and SUDO policy on the policy server.
-b	Runs in batch mode (no confirmation prompts).
-K appkey	
-h	Displays the usage message and exits.
-p prefix	Sets the PowerBroker for Unix & Linux installation prefix.
-s suffix	Sets the PowerBroker for Unix & Linux installation suffix.

Files

Not applicable

Example

For a pbuninstall execution example, see [Example of a pbuninstall Execution](#).

See Also

pbmakeremotetar(8), pbinstall(8), pbversion(8)

Upgrades and Reinstallations

The PowerBroker for Unix & Linux installers are designed to enable easy upgrades of an installed version of to a new version. During an upgrade, the current PowerBroker for Unix & Linux configuration can be retained, or a new PowerBroker for Unix & Linux configuration can be put in place.

PowerBroker for Unix & Linux installation scripts `pbininstall` and `pbmakeremotetar` can also be used to perform upgrades and reinstallations.

If you want to return to an older version of PowerBroker for Unix & Linux or reinstall the current version with a different configuration, PowerBroker for Unix & Linux can be reinstalled to the current or older version without uninstalling, as long as the older version is V2.8.1 or later.

Caution! If you have PowerBroker Express installed and you want to install PowerBroker for Unix & Linux UNIX, PowerBroker Linux, or PowerBroker Virtualization, you must first uninstall PowerBroker Express and then perform a full installation of PowerBroker for Unix & Linux as described in [pbininstall Installation Script](#).

Pre-Upgrade Instructions

Before performing an upgrade or reinstallation, do the following:

1. Obtain the new release, either on a CD or using FTP.
2. Read the release notes and installation instructions.
3. Determine the order for updating the Policy Server host machines. Note that `pbrun` clients need to be redirected to a new Policy Server Host while their primary Policy Server host is updated. If your current PowerBroker for Unix & Linux installation includes Policy Server host failover machines, you may want to consider upgrading the Policy Server hosts failover machines first, followed by the submit hosts and run hosts, followed by the primary Policy Server hosts.
Remember that the PowerBroker for Unix & Linux settings files on the Policy Server hosts may need to be updated as each Policy Server host is upgraded.
4. If your current PowerBroker for Unix & Linux installation includes one or more Policy Server host failover machines, then ensure that the security policy files on the primary Policy Server host and the Policy Server host failover machines are synchronized.
5. Verify the current location of the PowerBroker for Unix & Linux administration programs, user programs, and log files. This information is in the `pb.cfg` file (`/etc/pb.cfg` or `pb/install/pb.cfg.{flavor}`) and the settings file, `/etc/pb.settings`.
6. If you do not have a recent backup of the host, or if it is imperative that no log entries can be lost, then create a save directory (for example, `/var/tmp/pb.{rev_rel}`) that can be used to restore PowerBroker for Unix & Linux files from in case the upgrade fails. After creating the directory, copy (do not use move) the files that are listed below to the new save directory (a shell script can be created to copy the necessary files).

PowerBroker for Unix & Linux files for all host types:

```

/etc/services
/etc/pb.settings
/etc/pb.cfg (and pb.cfg.* on older installations)
/etc/pb.key (if encryption is in use on the system)
pb* log files (typically in /var/adm, /var/log or /usr/adm)

```

PowerBroker for Unix & Linux Policy Server Host Files:


```
/opt/pbul/policies/pb.conf
```

All included Security Policy Sub Files

```
/etc/inetd.conf
```

Any event log or I/O log files to save

PowerBroker for Unix & Linux Submit Host and Run Host files:

```
/etc/inetd.conf
```

PowerBroker for Unix & Linux Log Host files:

```
/etc/inetd.conf
```

Any event log or I/O log files to save

PowerBroker for Unix & Linux GUI Host files:

```
/etc/inetd.conf
```

7. Determine in which directories to install the new PowerBroker for Unix & Linux log files, administration programs, and user programs. If you chose different directories for the PowerBroker for Unix & Linux programs, you might need to update the path variable for the root user and other users.
8. Be aware that users cannot submit monitored task requests while PowerBroker for Unix & Linux updates are in progress. Consider writing a PowerBroker for Unix & Linux configuration policy file that rejects all users from executing pbrun and echoes a print statement to their screen, informing them that a PowerBroker for Unix & Linux upgrade is in progress.
9. PowerBroker for Unix & Linux releases are always upward-compatible when encryption is not used. It is recommended that you perform an uninstall if a release is replaced by a PowerBroker for Unix & Linux version before V2.8.1.
10. If you are using an encrypted settings file and intend to do an upgrade or reinstall, then the unencrypted version of the settings file needs to be restored before performing an upgrade or reinstall; otherwise, the settings file cannot be read.
11. If you have a previous installation of PowerBroker for Unix & Linux for V5.1 or earlier and your encryption is set to none, then when you install PowerBroker for Unix & Linux V5.2, all the encryption options (options 98 through 103) will be set to none. You can change these options during installation (for instructions, see [Advanced Installation Instructions Using pbininstall](#)).

pbininstall Install Upgrades

Starting with PowerBroker for Unix & Linux V2.8.1, it is not necessary to uninstall PowerBroker for Unix & Linux to upgrade to a new version, reinstall the same version, or reinstall an older version.

To upgrade or reinstall PowerBroker for Unix & Linux with the same configuration as the currently installed version, run pbininstall in batch mode:

```
./pbininstall -b
```

If you are performing a reinstall of an older version, then be aware that the older version may not have the same features as the newer version. In this case, the upgrade process discards the configuration of the features that are not available in the older version of PowerBroker for Unix & Linux. When you upgrade to the newer version, make sure to configure the newer features when running pbininstall.

To change the configuration of PowerBroker for Unix & Linux during the upgrade or reinstall, run pbininstall in interactive mode.

```
./pbininstall
```

The present configuration is read into pbininstall. Make the desired configuration changes and then use the `c` command to continue. pbininstall then installs PowerBroker for Unix & Linux with the new configuration.

For step-by-step instructions for using pbininstall, see [Step-by-Step Instructions for a Basic Installation Using pbininstall](#).

pbininstall Install Upgrades and Reinstallations

Upgrading or reinstalling PowerBroker for Unix & Linux with pbininstall is the same process as installing with pbininstall. There is one difference to be aware of. In pbininstall, the in-place files are backed up as sybak files during the upgrade process; whereas in a pbininstall upgrade or reinstall, the files are overwritten.

Post-Upgrade Instructions

If you want to encrypt your settings file after upgrading PowerBroker for Unix & Linux, then save a copy of the unencrypted file (for future upgrades) and re-encrypt the settings file.

Patch Installations

To perform a patch installation, see [pbpatchinstall](#).

Uninstalling PowerBroker for Unix & Linux

If `pbinstall` (or `pbmakeremotetar`) was used to install PowerBroker for Unix & Linux on the host, then use `pbuninstall` and the supporting files that were saved from the original `pbinstall` (or `pbmakeremotetar`) session to remove PowerBroker for Unix & Linux.

PowerBroker for Unix & Linux can be uninstalled by running the `uninstall` script, `pbuninstall`, located in the `powerbroker/<version>/<flavor>/install` directory. Running `pbuninstall` removes PowerBroker for Unix & Linux from only the machine that `pbuninstall` is run on. Other PowerBroker for Unix & Linux hosts and concurrent (prefixed and/or suffixed) installations are not affected unless the other hosts rely upon the host that is performing the `uninstall` for PowerBroker for Unix & Linux services.

For example, uninstalling PowerBroker for Unix & Linux from the only (or last) Policy Server host will probably severely impact PowerBroker for Unix & Linux functionality on your network.

IMPORTANT: If you are uninstalling the Policy Server v9 or higher, and you have selected "Integrate with Sudo" during installation, please be aware that your sudoers policies of all PowerBroker Sudo hosts are now centrally stored in the database in 'sudoersdb' settings. If you don't have the latest copy of these sudoers policies on the sudo hosts, it is recommended that you check out all the sudoers policies from this database (using `pbdbutil --sudo -e <options>`) prior to uninstalling the Policy Server, otherwise all your modifications will be lost.

To successfully uninstall PowerBroker for Unix & Linux, you need access to the directory from which `pbinstall` was executed, and to the `pb.cfg` file. If the installation was from a CD-ROM, then the CD must be mounted. If the distribution was using FTP, then the original installation tree must exist or be restored if it was removed.

You should back up the installation directory tree and the directory that contains the created `pb.cfg` file before you remove them. If you removed these directories without first performing a backup, then call BeyondTrust Technical Support for help.

If the distribution used FTP and the environment variable `TMPDIR` was not set during the installation, then these two directory trees are the same. If the distribution used a CD-ROM and `TMPDIR` was not set during the installation, then these files were created in `/tmp`.

The `uninstall` succeeds if the original files were not cleaned up before the `uninstall` or if the defaults were accepted during installation. If you intend to reinstall PowerBroker for Unix & Linux after the `uninstall`, then save copies, under different names, of any files you may want to look at later (for example, `/etc/pb.settings`).

The `pbuninstall` script, like the rest of the PowerBroker for Unix & Linux uninstallation suite, does not work with an encrypted settings file.

Example of a `pbuninstall` Execution

The following listing shows the `pbuninstall` and execution:

```
# ./pbuninstall
Starting pbuninstall main() from /opt/beyondtrust/powerbroker/v6.0/pbx86_linuxB-6.0.0-01/install/.
x86_linuxB
```

```
BeyondTrust PowerBroker Installation Removal
```

```
This script will remove PowerBroker programs and files from the system.
```

```
Hit return to continue
```

```
Looking for SuperDaemons to configure...
```

Finished looking for SuperDaemons to configure... Reading /etc/pb.cfg

Trying /etc/pb.settings

De-configuring system /etc/services and superdaemon configurations.

Removing PowerBroker 'prefix ' service definitions (if any) from /etc/services. Removing any PowerBroker definitions from SuperDaemon xinetd file /etc/xinetd.conf

Restarting superdaemons

Reloading SuperDaemon Configurations...

Done Reloading SuperDaemon Configurations... Moving /etc/pb.settings to /tmp

Moving /etc/pb.conf to /tmp Moving /etc/pb.key to /tmp Moving /etc/pb.key to /tmp Removing /usr/sbin/pbmasterd...

Removing /usr/local/man/man8/pbmasterd.8 ... Removing /usr/sbin/pblocald...

Removing /usr/local/man/man8/pblocald.8 ... Removing /usr/sbin/pbguid...

Removing /usr/local/man/man8/pbguid.8 ... Removing /usr/sbin/pblogd...

Removing /usr/local/man/man8/pblogd.8 ... Removing /usr/sbin/pbreport...

Removing /usr/local/man/man8/pbreport.8 ... Removing /usr/sbin/pbuvrpg...

Removing /usr/local/man/man8/pbuvrpg.8 ... Removing /usr/sbin/pbsyncd...

Removing /usr/local/man/man8/pbsyncd.8 ... Removing /usr/sbin/pbhostid...

Removing /usr/local/man/man8/pbhostid.8 ... Removing /usr/sbin/pbkey...

Removing /usr/local/man/man8/pbkey.8 ... Removing /usr/sbin/pbpasswd...

Removing /usr/local/man/man8/pbpasswd.8 ... Removing /usr/sbin/pbsum...

Removing /usr/local/man/man8/pbsum.8 ... Removing /usr/sbin/pblog...

Removing /usr/local/man/man8/pblog.8 ... Removing /usr/sbin/pbencode...

Removing /usr/local/man/man8/pbencode.8 ... Removing /usr/sbin/pblicense...

Removing /usr/local/man/man8/pblicense.8 ... Removing /usr/sbin/pbsync...

Removing /usr/local/man/man8/pbsync.8 ... Removing /usr/local/bin/pbcall...

Removing /usr/sbin/pbcheck...

Removing /usr/local/man/man8/pbcheck.8 ... Removing /usr/sbin/pbprint...

Removing /usr/local/man/man8/pbprint.8 ... Removing /usr/sbin/pbreplay...

Removing /usr/local/man/man8/pbreplay.8 ... Removing /usr/sbin/pbmerge...

Removing /usr/local/man/man8/pbmerge.8 ... Removing /usr/local/bin/pbrun...

Removing /usr/local/man/man1/pbrun.1 ... Removing /usr/local/bin/pbbench...

Removing /usr/local/man/man1/pbbench.1 ... Removing /usr/local/bin/pbksh...

Removing /usr/local/bin/pbsh...

Removing /usr/local/man/man8/pbinstall.8 ... Removing /usr/local/man/man8/pbuninstall.8 ... Removing

/usr/local/man/man8/pbmakeremotetar.8 ... Removing /usr/local/man/man8/pbversion.8 ...

Removing /usr/local/man/man8/pbpatchinstall.8 ... Removing /usr/local/bin/pbless...

Removing /usr/local/man/man1/pbless.1 ... Removing /usr/local/bin/pbmg...

Removing /usr/local/man/man1/pbmg.1 ... Removing /usr/local/bin/pbnvi...

Removing /usr/local/man/man1/pbnvi.1 ... Removing /usr/local/bin/pbumacs...

Removing /usr/local/man/man1/pbumacs.1 ... Removing /usr/local/bin/pbvi...

```
Removing /usr/local/man/man1/pbvi.1 ... Moving /var/log/pbmasterd.log to /tmp Moving
/var/log/pblocald.log to /tmp Moving /var/log/pblogd.log to /tmp Moving /var/log/pb.eventlog to /tmp
Moving /var/log/pbguid.log to /tmp Moving /etc/pb.cfg to /tmp
Moving /var/log/pbksh.log to /tmp Moving /var/log/pbsh.log to /tmp Moving /var/log/pbsync.log to /tmp
Moving /var/log/pbsyncd.log to /tmp
Removing pbguid html help and policy example files from '/usr/local/lib/pbbuilder' Removing
/usr/local/lib/pbbuilder -- empty BeyondTrust Created Directory
```

BeyondTrust PowerBroker Installation Removal was successful

PowerBroker configuration files and logs were moved to /tmp for removal

Solr Installations

Solr can be used to index PowerBroker for Unix & Linux IO logs to provide improved search capability. Indexing can be done on the IO log files on the PowerBroker for Unix & Linux logserver.

Installation Considerations

PowerBroker Solr is installed in a user-defined directory, logs to a second user-defined directory. The defaults are `/opt/pbul-solr` and `/var/log/solr`.

Supported Platforms

Solr is supported on various Linux, AIX, HPUX and Solaris platforms. Refer to PowerBroker for Unix & Linux ReadMe for more details on the specific platforms supported.

PowerBroker for Unix & Linux Solr Java Requirements

- Solr 4.1 (included)
- Java 1.6+ JRE or JDK

System Requirements

- Disk – PBUL Solr 4.1: 18MB
- Disk – Java 1.7: 58MB
- RAM – Solr: 2GB dedicated
- RAM – Java 1.7: 64MB

Unix/Linux Utilities

The PowerBroker for Unix & Linux installer requires the following Unix/Linux utilities and built-in commands:

- awk cut getopt ps sort unset
- basename date grep pwd stty vi
- cat diff id read tar wc
- cd dirname kill rm tee xargs
- chmod df ls rmdir touch
- chown echo mkdir sed tr
- cksum eval more set trap
- clear exec mv shift umask
- cp export od sleep uname

System File Modifications

AIX: /etc/inittab modified, backed up prior as inittab.bak.####

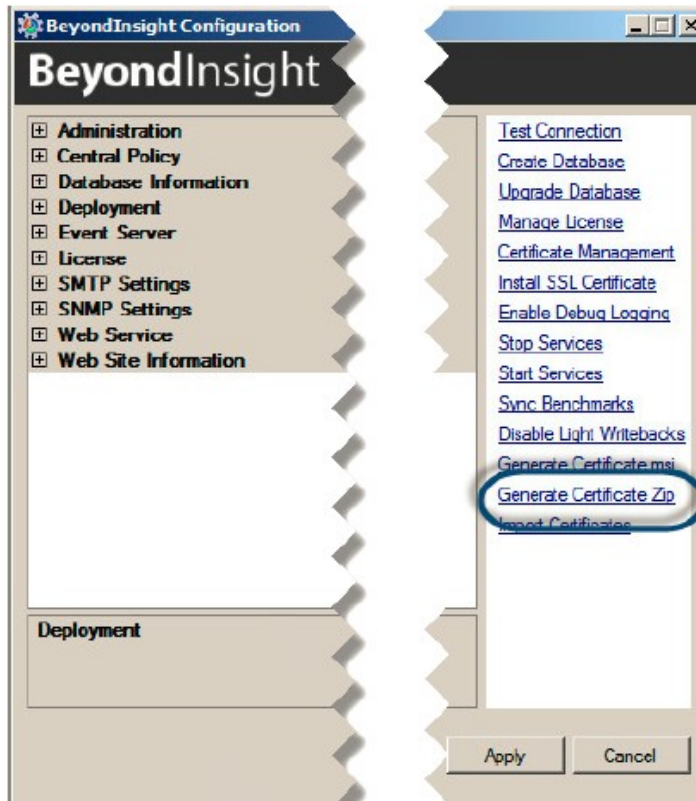
SSL Certificates and Search Interface

PBUL's Solr can be installed with and work with either BeyondInsight, or PBSMC. At this time, Solr cannot work with both, and cannot be changed from working with one to working with the other.

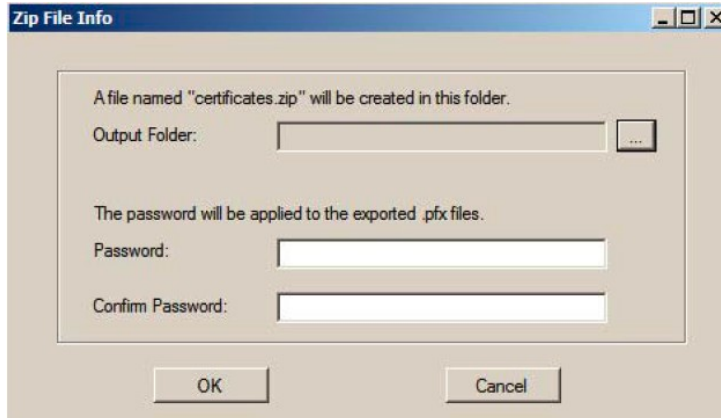
Prerequisites when Installing with BeyondInsight

Obtain the BeyondInsight Cert and CA files by copying the certificates from the BeyondInsight Windows Server machine to the Solr host machine:

1. Start the BeyondInsight Configuration Tool on the BeyondInsight Windows Server machine.
2. Click **Generate Certificate Zip** in the BeyondInsight Configuration Tool.



3. Select the output folder for the zip file and a password to apply to the exported .pfx file. This password is not used during the Solr install.



4. Select a folder where you can securely copy the file, and move it to your Unix/Linux server where you are planning to install Solr.

Command Line Options

When installing with BeyondInsight, an installation menu can be used to specify all options. When installing with PBSMC, or with manually generated certificates, the `-M` option at a minimum must be specified on the command line. Other options are available both on the command line and via menu.

Options for Use with BeyondInsight

<code>-a rcsuser</code>	Specify RCS Admin user
<code>-A file</code>	Specify file containing rcs admin password
<code>-s</code>	Configure local pb.settings
<code>-r</code>	Re-install with BI, without generating new certificates

Options for Use with PBSMC

<code>-M</code>	Install via PBSMC (skip BI registration and certificates)
<code>-K</code>	Filename of SSL Server certificate PEM file containing the private key May also contain the public certificate
<code>-k</code>	Filename of SSL Server certificate PEM file containing public certificate
<code>-C</code>	Filename of any CA certificate PEM file containing the CA public certificate May be used multiple time for multiple CA files
<code>-o</code>	Fully qualified path for openssl

Command Options

<code>-b basedir</code>	Set solr installation base directory
<code>-p port</code>	Set solr/jetty port

-j javahome	Set JAVA_HOME
-u user	Set Solr user
-c	If specified, create Solr user
-I uid	If creating Solr user, specify the UID
-G gid	If creating Solr user, specify the GID
-i	Configure init script/SMF/inittab
-l logdir	Specify solr log directory
-P file	Specify file containing java keystore password
-q	Quiet mode
-h	Display help

Installation

Solr is provided as a tarball named 'pbul-solr_multiarch-{version}.tar.Z' As root:

1. Make sure you have Java 1.6+ installed and know the Home directory of Java.
2. Create directory /opt/beyondtrust and cd to that directory.
3. Extract the Solr installation files:

```
# gunzip -c pbul-solr_multiarch-{version}.tar.Z | tar xvf -
```
4. Navigate to the install directory:

```
# cd powerbroker-solr/v7.5/install
```
5. Copy the file certificate.zip generated by BeyondInsight (as described in Prerequisites section above) in the install directory.
6. Start the solrinstall script with the following command; solrinstall has no command line options:

```
# ./solrinstall
```

The solrinstall menu displays options similar to the following:

BeyondTrust PowerBroker Solr Installation Menu

Opt	Description	[Value]
1	PowerBroker Solr installation directory	[/opt/pbul-solr]
2	Solr SSL port number	[8443]
3	JAVA_HOME environmental variable	[/usr/java/jre1.7.0_40]
4	Solr user	[solr]
5	Create Solr user?	[yes]
6	Solr user UID	[]
7	Solr user GID	[]
8	Configure init?	[yes]
9	Solr log directory	[/var/log/solr]
10	BeyondInsight certificate admin user name	[administrator]*
11	Configure local pb.settings with Solr	[no]

C to continue, X to exit

Please enter a menu option [For technical support call 1-800-234-9072]

7. During the install, you will be prompted for the keystore password:
Enter a keystore password (minimum 6 characters).

Note: This is a new password you will provide. You will enter this password during the Post-Install when you import the Solr certificates using the BeyondInsight Configuration Tool.

Menu Options

1. `PowerBroker Solr installation directory`
This is the directory where the Solr installation files are placed. The default value is `/opt/pbul-solr`.
2. `Solr port number`
The port number to be used for the Solr service. The default is 8983.
3. `JAVA_HOME environmental variable`
The value of `$JAVA_HOME`. This is set if environmental variable `JAVA_HOME` is set. Prior to installation, `$JAVA_HOME/bin/java` is tested for version compatibility.
4. `Solr user`
The non-root user that will run the Solr server. The default is 'solr'. If user 'solr' does not exist, menu displays options 5, 6, and 7 specifying whether to create the solr user, and optionally specifying the the uid/gid. The Solr user requires bash shell in order to run the Solr (jetty) startup script.
8. `Configure init (Linux/HP-UX; AIX uses inittab, Solaris 10+ uses SMF)`
Solr startup and shutdown are accomplished via init. Selecting 'yes' to this menu option configures init to startup and shutdown Solr.
9. `Solr log directory`
This is the directory where the Solr log files are placed. The default value is `/var/log/solr` (Linux). Other operating systems may use `/var/adm` or `/usr/adm` rather than `/var/log`.
10. `BeyondInsight Certificate Administrator user name`
The BeyondInsight Admin user; admin user password will be prompted for.
11. `Configure local pb.settings with Solr`
Answering yes will configure the local `pb.settings` file with the solr related keywords, configured for this solr installation. The keywords are: `solrhost`, `solrport`, `solrcafile`, `solrclientkeyfile`, `solrclientcertfile`

Post-Install when Installing with BeyondInsight

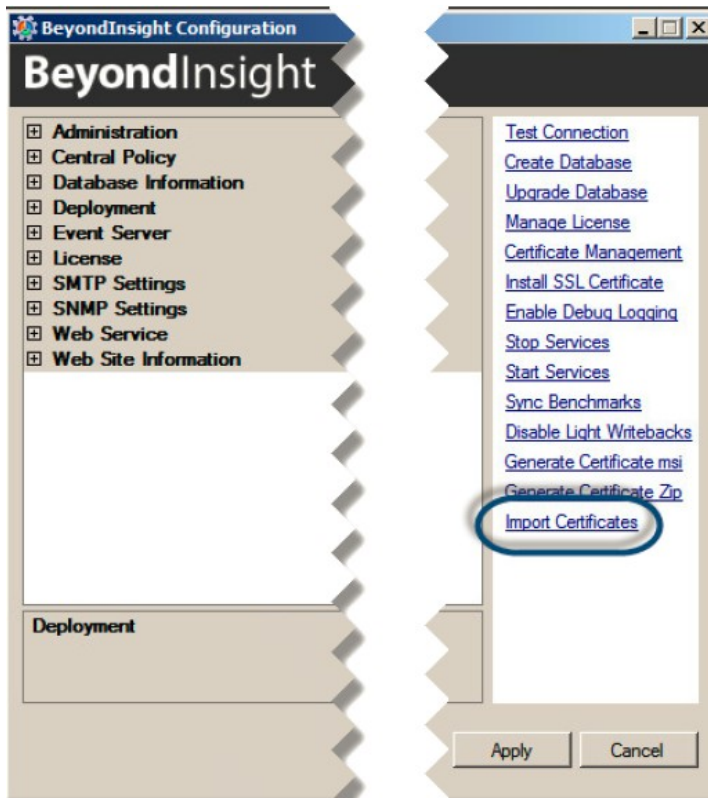
After `solrinstall` has installed and started Solr, Solr is registered with BeyondInsight.

To give the Solr Server a heartbeat, a script called `pbrcssolrupdate` is launched at the Solr installation, and with each restart of Solr services (jetty), where a Solr asset update event is sent to BeyondInsight daily.

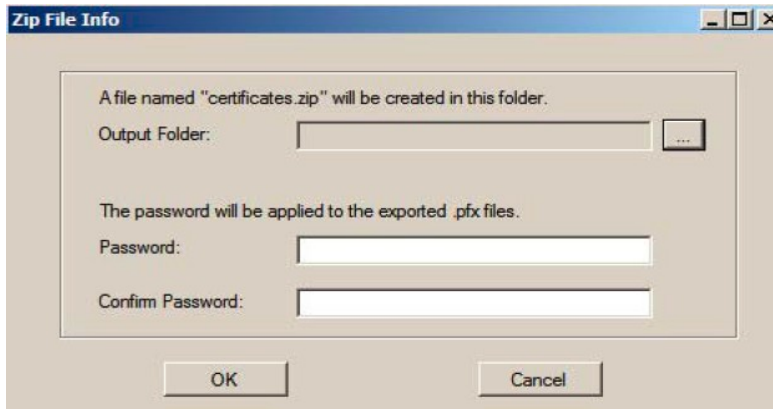
Follow the instructions as listed after a successful Solr install are displayed at the end of the installation.

In order for the logserver and Policy Server hosts to communicate with this Solr Server, for indexing PowerBroker for Unix & Linux IO log data, you will need to do the following:

1. On your BeyondInsight Windows server, start the BeyondInsight Configuration Tool.
2. Click **Import Certificates** to import the certificates created during the Solr install and grant privileges to the certificates for use by the Solr search.



3. Enter the password that you provided when you created the Certificates zip file. See [Prerequisites](#).



4. Securely copy the following files from the `/opt/pbul-solr/etc` to a secure directory on the PowerBroker for Unix & Linux Policy Server and logserver hosts:
 - `solr.<host>.client.pem`
 - `solr.<host>.ssl.CA.pem`

Note that a tarball (solr.\${shorthostname}.pbsettings.tar) is created with the certificate files and related settings, for convenient copying to other hosts. When the tarball is extracted from the root directory, the certificate files and solr.pb.settings will be placed in /etc/. The settings contained in /etc/solr.pb.settings must be manually merged into /etc/pb.settings.

5. In pb.settings of the Policy Server/Logserver hosts, add the following parameters:

```
solrhost <host>
solrport 8443
solrcafile <secure_directory>/solr.<host>.ssl.CA.pem
solrclientkeyfile <secure_directory>/solr.<host>.client.pem
solrclientcertfile <secure_directory>/solr.<host>.client.pem
```

Note that a tarball (solr.\${shorthostname}.pbsettings.tar) is created with the certificate files and related settings, for convenient copying to other hosts. When the tarball is extracted from the root directory, the certificate files and solr.pb.settings will be placed in /etc/. The settings contained in /etc/solr.pb.settings must be manually merged into /etc/pb.settings.

Re-Installation when Installing with BeyondInsight

Starting with v9.4, when re-installing solr, the installation script will recognize that certificates have already been generated, and the registration with BeyondInsight will be skipped. This prevents re-generation of certificates by BeyondInsight. In the case where re-generation of certificates is desired, the certificates must be manually cleared from BeyondInsight, and removed from the etc directory of the solr installation (default: /opt/pbul-solr/etc).

Solr Uninstall

As root:

1. Create directory /opt/beyondtrust and cd to that directory.
2. Extract the Solr installation files.


```
# gunzip -c pbul-solr_multiarch-{version}.tar.Z | tar xvf -
```
3. Navigate to the install directory:


```
# cd /opt/beyondtrust/powerbroker-solr/v7.5/install
```
4. Start the solruninstall script with the following command; solruninstall has 1 command line option:


```
# ./solruninstall
```

- or -

```
# ./solruninstall -clean
```

PowerBroker Sudo Installation

Starting with v9.1, installing PowerBroker Sudo on hosts with sudo will allow integration between sudo and PowerBroker for Unix & Linux. Sudo clients will transfer the sudoers policy to the Policy Server, and sudo will be configured to use PowerBroker for Unix & Linux sudo plugins for policy processing and/or IO logging.

When configured for PowerBroker Sudo policy processing, the PowerBroker for Unix & Linux Policy Server will store the sudoers policies in the PowerBroker sudoers database. When sudo is invoked, the PowerBroker for Unix & Linux policy plugin will contact the PowerBroker for Unix & Linux Policy Server to retrieve the latest sudoers policy for that client. The sudoers policy from the Policy Server is maintained in a cache on the client for sudo policy processing (accept/reject).

Sudo is configured to use the customized PowerBroker Sudo plugin that “reads” the sudoers policy from the client cache database. The PowerBroker Sudo client will initiate an accept event or a reject event based on the results of the sudoers policy processing.

When configured for PowerBroker Sudo IO logging, the PowerBroker for Unix & Linux iolog plugin will log to a PowerBroker for Unix & Linux Log Server.

Supported Platforms

PowerBroker Sudo is supported on various Linux, AIX, HPUX and Solaris platforms. Refer to PowerBroker for Unix & Linux ReadMe for more details on the specific platforms supported.

Unix/Linux Utilities

The PowerBroker Sudo installer requires the following Unix/Linux utilities and built-in commands:

awk	cut	getopt	ps	sort	unset
basename	date	grep	pwd	stty	vi
cat	diff	id	read	tar	wc
cd	dirname	kill	rm	tee	xargs
chmod	df	ls	rmdir	touch	
chown	echo	mkdir	sed	tr	
cksum	eval	more	set	trap	
clear	exec	mv	shift	umask	
cp	export	od	sleep	uname	

System File Modifications

PowerBroker Sudo modifies ‘sudo.conf’ and replaces ‘sudoers_policy’ and ‘sudoers_io’ plugins by the PowerBroker for Unix & Linux plugins:

Plugin sudoers_policy /usr/lib/beyondtrust/pb/pbsudoers.so

Plugin sudoers_io /usr/lib/beyondtrust/pb/pbsudoers.so

Prerequisites

PowerBroker Sudo requires sudo v1.8 or higher installed and properly configured on the host prior to PowerBroker Sudo installation. Sudo must be built with shared library support to use shared library plugins. Currently PowerBroker Sudo does not support ldap-enabled sudo. During the installation, the installer will check to see if sudo is configured to use ldap, and if so, it will exit with an error.

PowerBroker Sudo version 9.4 requires PowerBroker for Unix & Linux Policy Server v9.4.

PowerBroker Sudo installation utilizes the “client registration” capabilities, and requires an Application ID, Application Key, and Client Profile name as well as the hostname and port for a PowerBroker for Unix & Linux Policy Server/REST server. The PowerBroker for Unix & Linux Policy Server installation automatically creates two related Application IDs and keys: PBSUDOADMIN and PBSDUOREAD for administration and read-only access, respectively. The PBSUDOADMIN Application ID can be used when installing the PowerBroker Sudo client. Other Application IDs can be used as well, as long as the Application ID has the appropriate administration rights.

If, during the installation or upgrade of the PowerBroker Unix & Linux Policy Server/Log Server, the option ‘Integrate with sudo’ was set to yes, the install created a default registration profile ‘sudodefauit’ that can be used during the installation of PowerBroker Sudo. The install also created a file called ‘/etc/pbsudo.settings.default’ stored as ‘/etc/pbsudo.settings’ in ‘sudodefauit’ profile.

Although ‘sudodefauit’ registration profile created by pbininstall on the Policy Server is perfectly adequate to use, you can also create your own registration profile. Refer to “Sudo Integration” section in the Administration Guide for more information on how to create a registration profile.

Prior to running ‘pbsudoinstall’, you will need to create an Application ID and Key on the Policy Server.

Run the following command on the policy server:

```
pbdbutil --rest -g <appid>
```

For example:

```
# pbdbutil --rest -g sudoappid
{ "appkey": "934bbab5-503e-4c40-8486-90c748142431" }
```

Make sure you copy the value of the appkey generated in a secure, safe file. This information cannot be retrieved after it is generated.

The pbsudoinstall “default install” option `-d` can be used to automatically select the default port 24351, the default profile name “sudodefauit”, the default selections of using Power BrokerSudo for both Policy processing and I/O Logging, and to automatically execute the generated installation script.

PowerBroker Sudo Host Aliases (not to be confused with sudoers host aliases) can be used to group sudo client hosts that use the same sudoers policy.

Host aliases can be created on the Policy Server, or during pbsudoinstall. If a host alias is created, and the sudo client host is added to that host alias on the Policy Server prior to installing the client, that client will automatically detect that the alias is to be used.

If the client does not already belong to a host alias, the interactive installation will normally ask whether a host alias should be created or joined. The pbsudoinstall commandline option `-C` can be used to create an alias, and the `-J` commandline option can be used to join an alias (thus skipping the question during interactive installation).

When not using an alias, the first time the pbsudo client is installed on a host, that host’s existing sudoers policy file (and any included files) are uploaded to the Policy Server. Any subsequent re-installations do not normally re-upload the sudoers file(s). the `-U` and `-F` commandline options used together will force re-uploading the sudoers file(s).

Installation

PowerBroker Sudo is provided as a tarball named 'pbsudo{arch}-{version}.tar.Z'.

Prior to running the install script, please make sure the path where 'sudo' binary is located is in the environment variable PATH, and you can successfully run 'sudo -V'.

As root:

1. Create directory /opt/beyondtrust and cd to that directory.
2. Extract the PowerBroker Sudo installation files:
gunzip -c pbsudo{arch}-{version}.tar.Z | tar xvf -
3. Navigate to the install directory:
cd pbsudo/{version}/pbsudo{arch}-{version}/install
4. Start the pbsudoinstall script with the following command
./pbsudoinstall

The pbsudoinstall menu displays options similar to the following:

```
Client Registration provides a method of automatic configuration based upon
a profile provided by your Primary Policy Server.
```

```
To use this functionality you will need to know specific parameters from
your Primary Policy Server setup. Please see the installation guide for
details.
```

```
Enter the Application ID generated on the Primary Policy Server:
```

5. For a fresh install, enter the Application ID created on the Policy Server, as well as the Application Key, the name of the host where the Policy Server is installed, the REST port (pbrestport) and the registration profile name (default 'sudodefauit') as well as whether to configure sudo for policy processing and/or IO logging:
Enter the Application ID generated on the Primary Policy Server: PBSUDOADMIN
Enter the Application Key generated on the Primary Policy Server: cefd039d-966f-44e2-a2f8-c56804009cfb
Enter the Primary Policy Server address/domain name for registering clients: host1
Enter the Primary Policy Server REST TCP/IP port [24351]:
Enter the Registration Client Profile name [sudodefauit]:
Configure sudo to use PBUL Policy processing [yes]:
Configure sudo to use PBUL I/O log processing [yes]:
6. After Client registration, if the client host is not already a member of a host alias on the Policy Server, the install will ask if you wish to join or create a Host Alias on the Policy Server for this host:
A PBSudo Host Alias, defined in the Policy Server database, provides a way to group clients that must share a common set of sudoers policies.
Would you like to join an existing alias (j), create a new alias (c), or skip creating an alias (s) [s]:
If "join" is selected, a list of existing aliases is presented. Followed by:
Please enter the PBSudo Host Alias name to join:
If "create" is selected, the installer prompts for the alias name:
Please enter the PBSudo Host Alias name to create:
If "skip" (the default) is selected, or if the host alias requires a sudoers policy, and the client's sudoers policy cannot be located, the installer prompts for the sudoers location:

Enter the path of the primary sudoers policy [e.g. /etc/sudoers]:

Alternatively, for a fresh install, you can run 'pbsudoinstall' with command line options providing the above values (in batch mode '-b' or interactive mode to get the default values of the above set to the command line arguments). For example:

```
./pbsudoinstall -A sudoappid -D host1 -K b3d6e2c0-ae6-493f-87a5-
d7900d963028 -P 24351 -N sudodefauit -S sudo_alias1 -b
```

7. For an upgrade, where 'sudoers' file does not need to be re-imported, answer 'no' to the prompt:


```
Do you wish to utilize Client Registration which will overwrite
/etc/pbsudo.settings and re-import the sudoers file? [no]?
```
8. A fresh install will get the files /etc/pb.settings, /etc/pb.key, /etc/pbssl.pem from the Policy Server and place them in /etc.

It will also import the sudoers file (sudoers and all the included files specified in #includedir and #include) to the Sudoers database on the Policy Server.

It will then replace the 'Plugin' variables in 'sudo.conf' by PowerBroker for Unix & Linux plugins:

```
Plugin sudoers_policy /usr/lib/beyondtrust/pb/pbsudoers.so
```

```
Plugin sudoers_io /usr/lib/beyondtrust/pb/pbsudoers.so
```

PowerBroker Sudo Uninstall

PowerBroker Sudo can be uninstalled by running 'pbsudouninstall' located in pbsudo/{version}/pbsudo {arch}-{version}/install directory.

'pbsudouninstall' will remove all files installed and remove pbsudoers.so plugins from sudo.conf.

pbsudouninstall will normally restore the current sudoers policy (and included policy files) from the Policy Server, and if not using a host alias, remove the sudoers from the Policy Server's database. The -P commandline option can be used to skip this step, thus preserving any local files. Note that if there are not any local files (pbsudoinstall renames the original), this option will leave sudo in an un-usable state.

Example of a pbsudouninstall:

```
BeyondTrust PowerBroker Installation Removal
Exporting latest /etc/pbsudo.settings from /etc/pb.db

This script will remove PowerBroker
PBSudo programs and files from the system.

Hit return to continue

Trying /etc/pbsudo.settings
Updating policy files:
/etc/sudoers
Removing PBUL plugin definitions (if any) from /etc/sudo.conf.
Removing plugin definitions (if any) from /etc/sudo.conf.
Removing /usr/sbin/pbdbutil...
Moving /etc/pb.rest.key to /tmp/beyondtrust_pbinstall
Moving /opt/pbul/dbs/pbsvccache.db to /tmp/beyondtrust_pbinstall
```



```
BeyondTrust PowerBroker PBSudo  
Installation Removal was successful  
PowerBroker PBSudo configuration files and logs were moved to  
  /tmp/beyondtrust_pbinstall for removal
```