Documentation

February 14, 2012

Installation and Administration Guide

Release 6.5

beyondtrust®
privilege. made simple

www.beyondtrust.com

BeyondTrust
2173 Salk Avenue
Carlsbad, California 92008
Phone: +1 818-575-4000

Revision/Update Information: February 14, 2012
Software Version: PowerBroker Identity Services Enterprise Edition 6.5
Revision Number: 1

COPYRIGHT NOTICE
Copyright © 2012 BeyondTrust Software, Inc. All rights reserved. Use of this software and/or document, as and when applicable, is also subject to the terms and conditions of the license between the licensee and BeyondTrust Software, Inc. ("BeyondTrust") or BeyondTrust's authorized remarketer, if and when applicable.

TRADE SECRET NOTICE
This software and/or documentation, as and when applicable, and the information and know-how they contain constitute the proprietary, confidential and valuable trade secret information of BeyondTrust and/or of the respective manufacturer or author, and may not be disclosed to others without the prior written permission of BeyondTrust. This software and/or documentation, as and when applicable, have been provided pursuant to an agreement that contains prohibitions against and/or restrictions on copying, modification and use.

DISCLAIMER
BeyondTrust makes no representations or warranties with respect to the contents hereof. Other than, any limited warranties expressly provided pursuant to a license agreement, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING WITHOUT LIMITATION THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE.

LIMITED RIGHTS FARS NOTICE (If Applicable)
If provided pursuant to FARS, this software and/or documentation, as and when applicable, are submitted with limited rights. This software and/or documentation, as and when applicable, may be reproduced and used by the Government with the express limitation that it will not, without the permission of BeyondTrust, be used outside the Government for the following purposes: manufacture, duplication, distribution or disclosure. (FAR 52.227.14(g)(2)(Alternate II))

LIMITED RIGHTS DFARS NOTICE (If Applicable)
If provided pursuant to DFARS, use, duplication, or disclosure of this software and/or documentation by the Government is subject to limited rights and other restrictions, as set forth in the Rights in Technical Data – Noncommercial Items clause at DFARS 252.227-7013.

TRADEMARK NOTICES
PowerBroker, PowerPassword, and PowerKeeper are registered trademarks of BeyondTrust. PowerSeries, PowerADvantage, PowerBroker Password Safe, PowerBroker Directory Integrator, PowerBroker Management Console, PowerBroker Desktops, PowerBroker Virtualization, PowerBroker Express, PowerBroker Databases, PowerBroker Windows Servers, PowerBroker Windows Desktops, and PowerBroker Identity Services are trademarks of BeyondTrust.
ssh® is a registered trademark of SSH Communications Security Corp in the United States and in certain other jurisdictions. The SSH logo, Tectia and tectia logo are trademarks of SSH Communications Security Corp and may be registered in certain jurisdictions.
This application contains software powered by PKAIP®, the leading solution for enabling efficient and secure data storage and transmission. PKAIP® is provided by PKWARE, the inventor and continuing innovator of the ZIP file format. Used with permission.

FICTITIOUS USE OF NAMES
All names of persons mentioned in this document are used fictitiously. Any resemblance to actual persons, living or dead is entirely coincidental.

OTHER NOTICES
If and when applicable the following additional provisions are so noted:
The PowerBroker Identity Services Open software is free to download and use according to the terms of the Limited GPL 2.1 for client libraries and the GPL 2 for daemons. The licenses for PowerBroker Identity Services Enterprise and for PowerBroker Identity Services UID-GID Module are different. For complete information on the software licenses and terms of use for BeyondTrust products, see www.beyondtrust.com.

Contents

Quick Start with PBIS Open **13**
Install the Agent on Linux, Join a Domain, and Log On 13
Step 1: Download PBIS Open 13
Step 2: Install PBIS Open on Linux 14
Step 3: Join Active Directory 14
Step 4: Log On with AD Credentials 16
Install the Agent on Mac OS X, Join a Domain, and Log On 16
Step 1: Download PBIS Open 16
Step 2: Install PBIS Open on a Mac 17
Step 3: Join Active Directory 17
Step 4: Log On with AD Credentials 19
Set Common Options 19
Give Your Domain Account Admin Rights 20
Upgrade to the Latest Version 21

PBIS Agent **22**
Services 22
PBIS Input-Output Service 24
PAM Options 25
Managing the PBIS Services 25
PBIS Registry 26
Ports and Libraries 26
Caches and Databases 26
Time Synchronization 28
Using a Network Time Protocol Server 29
Automatic Detection of Offline Domain Controller and Global Catalog 29
UID-GID Generation in PowerBroker Cells 30
Cached Credentials 30
Trust Support 30
Working with Trusts 31
Trusts and Cells in PBIS Enterprise 32
Integrating with Samba 33
Supported Platforms 33

Configuring Clients Before PBIS Agent Installation **34**
Configure nsswitch.conf 34
Configure resolv.conf 34
Configure Firewall Ports 35
Extend Partition Size (IBM AIX) 35
Increase Max Username Length (IBM AIX) 35
Check System Health 36

Installing the PBIS Agent **42**

Checking Your Linux Kernel Release Number 42
Package Management Commands 43
Requirements for the Agent 43
Patch Requirements 43
Other Requirements for the Agent 44
Networking Requirements 45
Disk Space Requirements 45
Memory and CPU Requirements 45
Clock Skew Requirements 46
Install the Agent on Linux or Unix with the Shell Script 46
Install the Agent on Linux in Unattended Mode 47
Install the Agent on Unix with the Command Line 47
Install the Agent on a Mac OS X Computer 48
Install the Agent on a Mac in Unattended Mode 48
Install the Agent in Solaris Zones 50
Upgrading Your Operating System 51

Joining an Active Directory Domain 52

Privileges and Permissions 53
Removing a Computer from a Domain 53
Creation of Local Accounts 53
Join Active Directory from the Command Line 54
Before Joining a Domain 55
Join a Linux or Unix Computer to Active Directory 55
Join a Mac Computer to Active Directory 56
Join a Linux or Unix Computer to an Organizational Unit 56
Join a Linux or Unix Computer to a Nested Organizational Unit 56
Domainjoin-cli Options, Commands, and Arguments 57
Basic Commands 57
Advanced Commands 58
Preview the Stages of the Domain Join for Your Computer 58
Check Required Configurations 59
View Details about a Module 60
Turn On or Turn Off Domain-Join Modules 61
Configuration and Debugging Commands 62
Join Active Directory Without Changing /etc/hosts 63
Join a Linux Computer to Active Directory 64
Join a Mac Computer to Active Directory 66
Turn Off OS X Directory Service Authentication 68
Use PBIS with a Single Organizational Unit 69
Rename a Joined Computer 70
Rename a Computer by Using the Command-Line Tool 70
Rename a Computer by Using the Domain Join Tool GUI 71
Files Modified When You Join a Domain 72
NetworkManager: Use a Wired Connection to Join a Domain 74

Logging on with Domain Credentials 75

Log on with AD Credentials ... 76
Log on with SSH ... 76
Solve Logon Problems from Windows ... 76
Solve Logon Problems on Linux or Unix ... 77
Make Sure You Are Joined to the Domain ... 77
Check Whether You Are Using a Valid Logon Form ... 78
Clear the Cache ... 78
Destroy the Kerberos Cache ... 78
Check the Status of the PBIS Authentication Service ... 78
Check Communication between the PBIS Service and AD ... 78
Verify that PBIS Can Find a User in AD ... 79
Make Sure the AD Authentication Provider Is Running ... 79
Run the id Command to Check the User ... 80
Switch User to Check PAM ... 81
Test SSH ... 81
Run the Authentication Service in Debug Mode ... 81
Check Nsswitch.Conf ... 81
On HP-UX, Escape Special Characters at the Console ... 82
Additional Diagnostic Tools ... 82
Troubleshooting SSH SSO Problems ... 82
Use NT4-style Credentials and Escape the Slash Character ... 82
Perform General Logon Troubleshooting ... 83
Get an SSH Log ... 83
After an Upgrade, Reconfigure SSH for PBIS ... 83
Verify that Port 22 Is Open ... 83
Make Sure PAM Is Enabled for SSH ... 84
Make Sure GSSAPI Is Configured for SSH ... 85
Check the Configuration of SSH for SSO ... 86
Platform-Specific Issues ... 88
More Information ... 94

Troubleshooting Domain-Join Problems ... 95

Solve Domain-Join Problems ... 96
Verify that the Name Server Can Find the Domain ... 96
Make Sure the Client Can Reach the Domain Controller ... 96
Check DNS Connectivity ... 96
Make Sure nsswitch.conf Is Configured to Check DNS for Host Names ... 96
Generate a Domain-Join Log ... 96
Ensure that DNS Queries Use the Correct Network Interface Card ... 97
Determine If DNS Server Is Configured to Return SRV Records ... 97
Make Sure that the Global Catalog Is Accessible ... 97
Verify that the Client Can Connect to the Domain on Port 123 ... 98
FreeBSD: Run ldconfig If You Cannot Restart Computer ... 98
Ignore Inaccessible Trusts ... 98
Resolve Error Messages ... 100
Configuration of Krb5 ... 100

Diagnose NTP on Port 123 ... 100
Output When There Is No NTP Service ... 101
Turn off Apache to Join a Domain ... 102

Configuring Clients After PBIS Agent Installation ... 103

Modify Settings with the Config Tool ... 103
Add Domain Accounts to Local Groups ... 104
Configure Entries in Your sudoers Files ... 105
Check a User's Canonical Name on Linux ... 106
Specify a sudoers Search Path ... 106
AIX: Create Audit Classes to Monitor Events ... 106

Troubleshooting the PBIS Agent ... 108

PBIS Services ... 108
Check the Status of the Authentication Service ... 109
Check the Status of the DCE/RPC Service ... 109
Check the Status of the Network Logon Service ... 110
Check the Status of the Input-Output Service ... 110
Restart the Authentication Service ... 111
Restart the DCE/RPC Service ... 111
Restart the Network Logon Service ... 111
Restart the Input-Output Service ... 112
Logging ... 112
Temporarily Change the Log Level and Target for a Service ... 114
Generate a Domain-Join Log ... 115
Generate a PAM Debug Log ... 116
Generate a Directory Service Log on a Mac ... 117
Generate a Network Trace ... 118
Basic Troubleshooting ... 118
Check the Version and Build Number ... 118
Determine a Computer's FQDN ... 119
Make Sure Outbound Ports Are Open ... 120
Check the File Permissions of nsswitch.conf ... 120
Configure SSH After Upgrading It ... 121
Upgrading an Operating System ... 121
Accounts ... 121
Allow Access to Account Attributes ... 121
User Settings Are Not Displayed in ADUC ... 122
Resolve an AD Alias Conflict with a Local Account ... 123
Troubleshoot with the Get Status Command ... 124
Troubleshoot User Rights with Ldp.exe and Group Policy Modeling ... 125
Fix Selective Authentication in a Trusted Domain ... 129
Cache ... 130
Clear the Authentication Cache ... 130
Clear a Corrupted SQLite Cache ... 131
Kerberos ... 133

Fix a Key Table Entry-Ticket Mismatch 133
Fix a KRB Error During SSO 134
Eliminate Logon Delays When DNS Connectivity Is Poor 136
Eliminate Kerberos Ticket Renewal Dialog 136
PAM 136
Dismiss the Network Credentials Required Message 137
OS-Specific Troubleshooting 137
Red Hat and CentOS 137
Ubuntu 139
SUSE Linux Enterprise Desktop (SLED) 139
AIX 140
FreeBSD 141
Solaris 142
Mac OS X 143

Command-Line Reference 144

Manage PBIS Services (lwsm) 144
Modify Settings (config) 145
Start the Registry Shell (regshell) 145
Export the Registry to an Editor (edit-reg) 146
Set the Log Level (set-log-level) 146
Change the Hostname in the Local Provider (set-machine-name) 146
Find a User or a Group 147
Find a User by Name 147
Find a User by UID 148
Find a User by SID 148
Find a Group by Name 148
Find a Group by ID 148
List Groups for a User (list-groups-for-user) 149
List Groups (enum-groups) 149
List Users (enum-users) 150
List the Status of Authentication Providers (get-status) 150
List the Domain 151
List Domain Controllers (get-dc-list) 151
List Domain Controller Information (get-dc-name) 152
List Domain Controller Time (get-dc-time) 152
List Computer Account Information (lsa ad-get-machine) 152
Dynamically Update DNS (update-dns) 153
Manage the AD Cache (ad-cache) 153
On Mac OS X 154
Join or Leave a Domain (domainjoin-cli) 154
Display NIS Map (ypcat) 154
Display the Value of a Key in an NIS Map (ypmatch) 155
Modify Objects in AD (adtool) 155
Using the Tool 157
Options 159

Examples 160
Copy Files Across Disparate Operating Systems (lwio-copy) 163
Modify Local Accounts 163
Add a Local User (add-user) 163
Add a Local Group Member (add-group) 164
Remove a Local User (del-user) 164
Remove a Local Group (del-group) 164
Modify a Local User (mod-user) 164
Modify the Membership of a Local Group (mod-group) 165
Kerberos Commands 165
Destroy the Kerberos Ticket Cache (kdestroy) 165
View Kerberos Tickets (klist) 165
Obtain and Cache a TGT (kinit) 166
Change a Password (kpasswd) 166
The Keytab File Maintenance Utility (ktutil) 167
Acquire a Service Ticket and Print Key Version Number (kvno) 167
Manage PBIS Enterprise from the Windows Command Line (lwopt.exe) 168

Leaving a Domain and Uninstalling the PBIS Agent 170

Leave a Domain 170
Remove the Computer Account in Active Directory 171
Remove a Linux or Unix Computer from a Domain 171
Remove a Mac from a Domain 171
Remove a Mac from a Domain from the Command Line 171
Uninstall the Domain Join GUI 172
Uninstall the Agent on a Linux or Unix Computer 172
Uninstall the Agent on a Mac 172

Monitoring Events with the Event Log 173

View the Local Event Log 174
Event Types 176
Event Sources 176
Event Source IDs 177

Single Sign-On Using PBIS 181

How PBIS Makes SSO Happen 181
How to Implement SSO with PBIS 182
Enable PAM for SSH 183
Configure PuTTY for Windows-Based SSO 185
Configure PuTTY 186
Configure the Base Linux Computer in Active Directory 186
Configure Apache for SSO 188
Prerequisites 189
Configure Apache HTTP Server 2.2 for SSO on RHEL 5 191
Control Group Access with mod_authz_unixgroup 196
Configure Firefox for SSO 196

Configure Internet Explorer for SSO 198
Troubleshooting Kerberos Authentication 200
Examples 205

Configuring PBIS with the Registry 206

The Structure of the Registry 206
Data Types 207
Modify Settings with the config Tool 208
Another Example 209
Access the Registry 210
Change a Registry Value by Using the Shell 211
Set Common Options with the Registry Shell 213
Change a Registry Value from the Command Line 214
Find a Registry Setting 214
lsass Settings 215
Log Level Value Entries 215
Turn on Event Logging 215
Turn off Network Event Logging 216
Restrict Logon Rights 216
Display an Error to Users Without Access Rights 217
Display a Message of the Day 217
Change the Domain Separator Character 218
Change Replacement Character for Spaces 218
Turn Off System Time Synchronization 219
Set the Default Domain 220
Set the Home Directory and Shell for Domain Users 220
Set the Umask for Home Directories 222
Set the Skeleton Directory 223
Force PBIS Enterprise to Work Without Cell Information 224
Refresh User Credentials 225
Turn Off K5Logon File Creation 225
Change the Duration of the Computer Password 226
Sign and Seal LDAP Traffic 227
NTLM Settings 227
Additional Subkeys 229
Add Domain 229
Control Trust Enumeration 230
Modify Smart Card Settings 231
Set the Interval for Checking the Status of a Domain 231
Set the Interval for Caching an Unknown Domain 232
lsass Cache Settings 232
Set the Cache Type 232
Cap the Size of the Memory Cache 233
Change the Duration of Cached Credentials 233
Change NSS Membership and NSS Cache Settings 234
eventlog Settings 236

Allow Users and Groups to Delete Events 236
Allow Users and Groups to Read Events 236
Allow Users and Groups to Write Events 237
Set the Maximum Disk Size 237
Set the Maximum Number of Events 238
Set the Maximum Event Timespan 238
Change the Purge Interval 238
netlogon Settings 239
Set the Negative Cache Timeout 239
Set the Ping Again Timeout 239
Set the Writable Rediscovery Timeout 240
Set the Writable Timestamp Minimum Change 240
Set CLdap Options 240
lwio Settings 241
Sign Messages If Supported 241
Lwedsplugin Settings for Mac Computers 241

Contact Technical Support 244
Before Contacting Technical Support 244
Contacting Support 246

Quick Start with PBIS Open

PowerBroker Identity Services Open Edition is an agent-based tool that allows you connect Linux, Unix, and Mac OS X computers to Microsoft Active Directory for consistent security policy across your entire environment.

To get started with PBIS Open, you need to install the PBIS agent, join a domain, and log on using Active Directory credentials. You can do so on Linux or Mac OS X, or you can refer to the instructions for joining a domain from the command line of a Unix computer.

Depending on your environment, you may also need to set common options and give your domain account admin rights.

If you already have a previous version of PBIS Open or Likewise Open installed, you should upgrade to the latest version.

Install the Agent on Linux, Join a Domain, and Log On

This topic skips system requirements and information about pre-configuring clients to cut to the chase: Installing PowerBroker Identity Services Open Edition on a Linux computer, connecting it to an Active Directory domain, and logging on with your domain credentials. (For other operating systems, see Install the Agent on Unix or Install the Agent on Mac OS X.)

Before you deploy PBIS Open in anything other than a test environment, you should read the overview of the agent, the chapter about installing the agent, the chapter about joining a domain, and the chapter about configuring the PBIS services.

Step 1: Download PBIS Open

Browse to www.beyondtrust.com and click **Free Software**. Under PowerBroker Identity Services Open Edition click **Download Free Trial**. Enter your information and submit the form.

In the email message that you receive in response to the form you submitted, click the link under the Download section to open a webpage where you can download installers for different operating systems. On the webpage, right-click the download link for your platform and then save the installer to the desktop of your Linux computer.

Step 2: Install PBIS Open on Linux

You install PBIS Open by using a shell script that contains a self-extracting executable—an SFX installer with a file name that ends in sh. Example: `PBISOpen-6.5.0.3499-linux-i386-rpm.sh`.

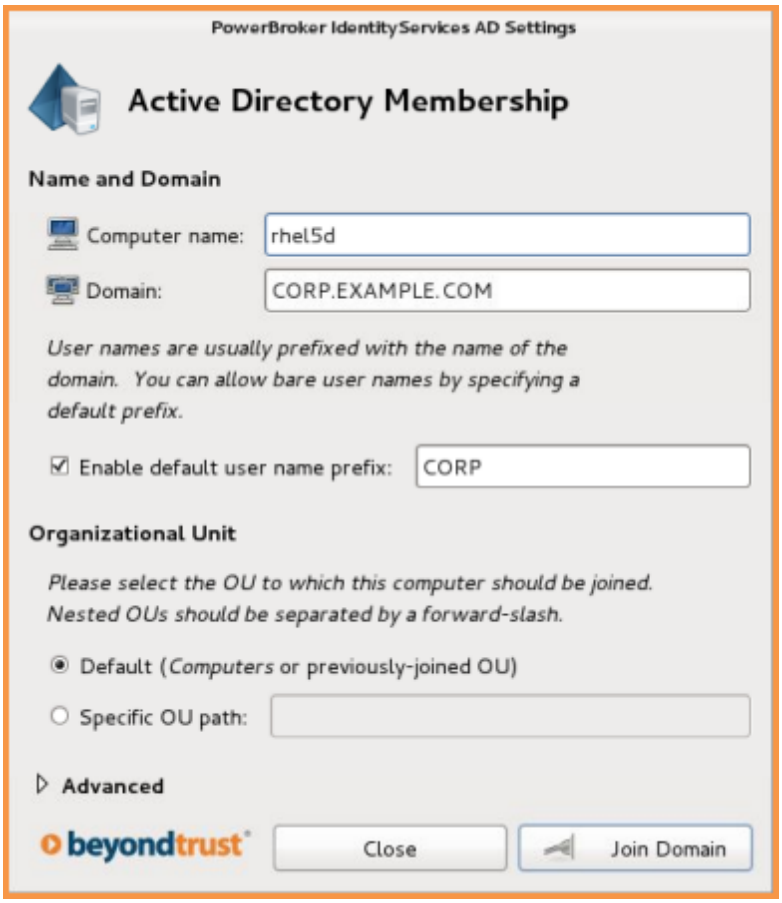
1. As root, run the installer, substituting the file name of the installer that you have selected for the one shown below:
   `sh ./PBISOpen-6.5.0.3499-linux-i386-rpm.sh`
   Alternatively, you can run the installer as a regular user:
   `sudo sh ./PBISOpen-6.5.0.3499-linux-i386-rpm.sh`
2. Follow the instructions in the installer.
   Note: On SLES and other systems on which the pager is set to less, you must exit the end user license agreement, or EULA, by typing the following command: `q`

Step 3: Join Active Directory

After the wizard finishes installing PBIS Open, the user interface for joining a domain appears. If it does not appear, see Join Active Directory with the Command Line.

To join a computer to a domain, you must use the root account and you must have the user name and password of an Active Directory account that has privileges to join computers to the domain.

1. In the **Domain** box, enter the fully qualified domain name (FQDN) of your Active Directory domain. Example: `CORP.EXAMPLE.COM`



2. To avoid typing the domain prefix before your user or group name each time you log on, select **Enable default user name prefix** and enter your domain prefix in the box. Example: `CORP`
3. Under **Organizational Unit**, you can optionally join the computer to an organizational unit (OU) by selecting **Specific OU Path** and then typing a path in the box. The OU path is from the top of the Active Directory domain down to the OU that you want. (See Use PBIS with a Single OU.)
   Or, to join the computer to the Computers container, select **Default (Computers or previously joined OU)**.
4. Click **Join Domain**.
5. Enter the user name and password of an Active Directory account that has privileges to join computers to the domain and then click **OK**.

After you join a domain for the first time, you must restart the computer before you can log on.

To solve problems, see Troubleshooting Domain-Join Problems or run this command at the command line: `domainjoin-cli --help`

Step 4: Log On with AD Credentials

After you have joined your Linux computer to a domain and restart the computer, you can log on interactively or from the text login prompt with your Active Directory credentials in the following form: `DOMAIN\username`. If you set a default domain, just use your Active Directory username.

1. Log out of the current session.
2. Log on the system console by using the name of your Active Directory user account.
   If you did not set a default domain, log on the system console by using an Active Directory user account in the form of `DOMAIN\username`, where `DOMAIN` is the Active Directory domain name. Example:

   ```
   example.com\kathy
   ```

   Important: When you log on from the command line, for example with ssh, you must use a slash to escape the slash character, making the logon form `DOMAIN\\username`.

To troubleshoot issues, see Solve Logon Problems on Linux.

Install the Agent on Mac OS X, Join a Domain, and Log On

This topic covers installing PowerBroker Identity Services Open Edition on a Mac, connecting it to an Active Directory domain, and logging on with your domain credentials. (For other operating systems, see Install the Agent on Linux or Install the Agent on Unix.)

Before you deploy PBIS Open in anything other than a test environment, you should read the overview of the agent, the chapter about installing the agent, the chapter about joining a domain, and the chapter about configuring the PBIS services.

Step 1: Download PBIS Open

Browse to www.beyondtrust.com and click **Free Software**. Under PowerBroker Identity Services Open Edition click **Download Free Trial**. Enter your information and submit the form.

In the email message that you receive in response to the form you submitted, click the link under the Download section to open a webpage where you can download installers for different operating systems. On the webpage, right-click the download link for your platform and then save the installer to the desktop of your Mac.

Important: On an Intel-based Mac, install the **i386** version of the `.dmg` package. On a Mac that does not have an Intel chip, install the **powerpc** version of the `.dmg` package. On Mac OS X 10.6 (Snow Leopard), you must use the 10.6 universal installation package.

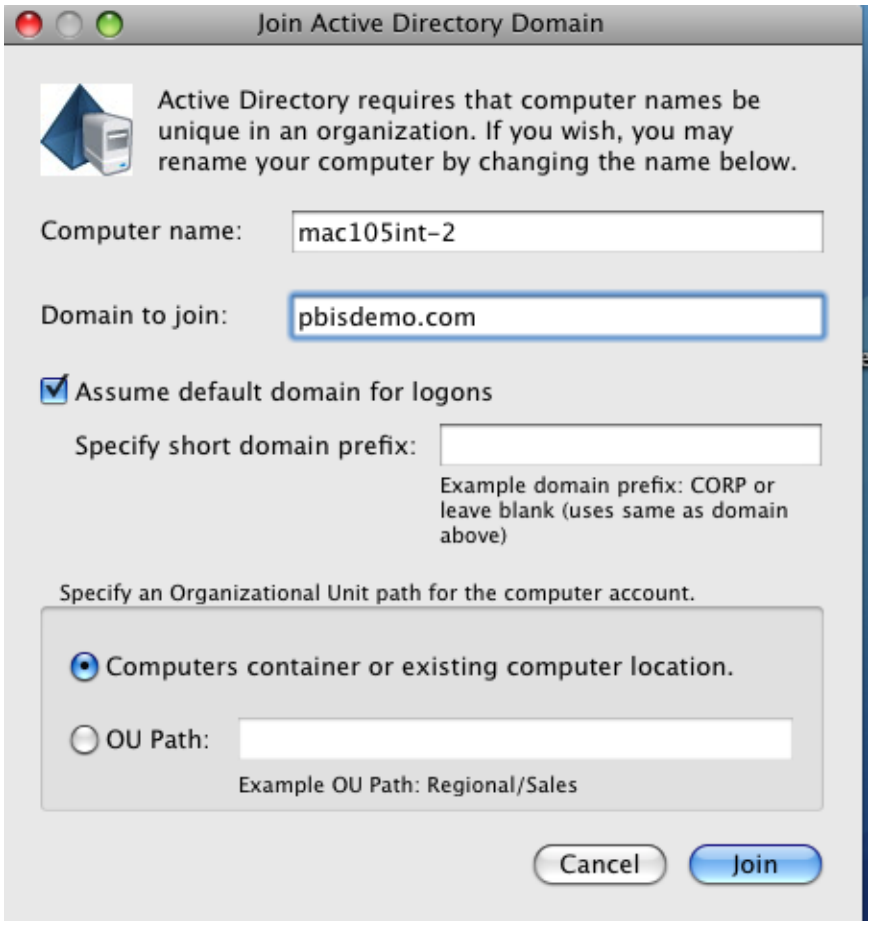
Step 2: Install PBIS Open on a Mac

To install the PBIS agent on a computer running Mac OS X, you must have administrative privileges on the Mac.

1. Log on to the Mac with a local account that has administrative privileges.
2. On the **Apple** menu , click **System Preferences**.
3. Under **Internet & Network**, click **Sharing**, and then select the **Remote Login** check box. Turning on Remote Login lets you access the Mac with SSH after you install PBIS.
4. On the Mac computer, go to the Desktop and double-click the PBIS `.dmg` file.
5. In the Finder window, double-click the PBIS `.mpkg` file.
6. Follow the instructions in the installation wizard.

Step 3: Join Active Directory

After the wizard finishes installing PBIS Open, the Join Active Directory Domain dialog is displayed. If it does not appear or if you want to join the domain later, see Join a Mac Computer to Active Directory with the GUI.



To join a computer to a domain, you must have administrative privileges on the Mac and be a member of the Domain Administrator security group or have otherwise been granted privileges on the Active Directory domain that allow you to join computers to the domain.

1. In the **Computer name** box, type the local hostname of the Mac without the `.local` extension. Because of a limitation with Active Directory, the local hostname cannot be more than 15 characters. Also, `localhost` is not a valid name.
2. In the **Domain to join** box, enter the fully qualified domain name (FQDN) of your Active Directory domain. Example: `engineering.example.com`
3. Under **Organizational Unit**, you can join the computer to an organizational unit (OU) by selecting **OU Path** and then typing a path in the **OU Path** box. The OU path is from the top of the Active Directory domain down to the OU that you want.
   Or, to join the computer to the Computers container, select **Default to "Computers" container**.
4. Click **Join**.

To solve problems, see Troubleshooting Domain-Join Problems.

Step 4: Log On with AD Credentials

After you have installed PBIS Open and joined the Mac computer to a domain, you can log on interactively with your Active Directory credentials.

1. Log out of the current session.
2. Log on to the Mac by using the name of your Active Directory user account in the form of `DOMAIN\username`, where `DOMAIN` is the Active Directory domain name.
   Example:

   `example.com\kathy`

   Important: If you log on from the command line, you must use a slash to escape the slash character, making the logon form `DOMAIN\\username`.

Set Common Options

This section shows you how to quickly modify two common PBIS settings—the default domain and the shell—by running the following `config` command-line tool as root:

/opt/pbis/bin/config

To view the settings you can change with `config`, execute the following command:

```
/opt/pbis/bin/config --list
```

The syntax to change the value of a setting is as follows, where `setting` is replaced by the PBIS option that you want to change and `value` by the new value that you want to set:

```
/opt/pbis/bin/config setting value
```

Here is an example of how to use `config` to change the `AssumeDefaultDomain` setting:

```
[root@rhel5d bin]# ./config --detail AssumeDefaultDomain
❶
Name: AssumeDefaultDomain
Description: Apply domain name prefix to account name at
logon
Type: boolean
Current Value: false
Accepted Values: true, false
Current Value is determined by local policy.
```

```
[root@rhel5d bin]# ./config AssumeDefaultDomain true ❷

[root@rhel5d bin]# ./config --show AssumeDefaultDomain ❸
boolean
true
local policy
```

❶ Use the `--detail` argument to view the setting's current value and to determine the values that it accepts.

❷ Set the value to `true`.

❸ Use the `--show` argument to confirm that the value was set to `true`.

Here is another example. To set the shell for a domain account, run `config` as root with the `LoginShellTemplate` setting followed by the path and shell that you want:

```
[root@rhel5d bin]# /opt/pbis/bin/config
LoginShellTemplate /bin/ksh
```

For more information, see Set the Home Directory and Shell for Domain Users and the section on config.

Give Your Domain Account Admin Rights

You can give your Active Directory account local administrative rights to execute commands with superuser privileges and perform tasks as a superuser.

On Ubuntu, you can simply add your domain account to the `admin` group in the `/etc/group` file by entering a line like the following as root:

```
admin:x:115:EXAMPLE\kathy
```

On other Linux systems, you can add an entry for your Active Directory group to your sudoers file—typically, `/etc/sudoers`— by editing the file with the `visudo` command as root. Editing the sudoers file, however, is recommended only for advanced users, because an improperly configured sudoers file could lock out administrators, mess up the privileges of important accounts, or undermine the system's security.

Example entry of an AD user account:

```
% EXAMPLE\\domain^admins ALL=(ALL) ALL
```

Note: The example assumes that you are a member of the Active Directory domain administrators group.

For information about how to format your sudoers file, see your computer's man page for sudo.

Upgrade to the Latest Version

With PowerBroker Identity Services Open Edition 6 or later, you can seamlessly upgrade from version 5, preserving your local configuration and maintaining your Active Directory state. Simply install PBIS Open 6 or later while version 5.3 or earlier is running and the computer is joined to a domain. It is unnecessary to leave the domain and uninstall the old version before you install the latest version. After installation, you will still be connected to your domain.

PBIS Open 6 preserves the changes you made to your local PBIS configuration. When you upgrade, a utility in PBIS Open 6 converts the configuration files from versions 5.0, 5.1, 5.2, and 5.3 into registry files and loads the files into the registry. The registry files that capture the old configuration are stored in `/tmp/upgrade`; the original configuration files in `/etc/pbis` are removed.

Although the latest Ubuntu release makes the `pbis-open` package available through the `apt-get install` command, the PBIS Open 6 installer does not support upgrading from the package. Before you upgrade from the version available through Ubuntu, it is recommended that you leave the domain, uninstall the domain join GUI package (`pbis-open-gui`), and uninstall the `pbis-open` package.

Important: If you plan to upgrade from a 4.x or earlier version to PBIS Open 6.0 or later, first contact BeyondTrust Technical Support at pbis-support@beyondtrust.com. At this time, it is recommended that you do not attempt to upgrade to a 6.x version from a 4.x version without assistance from BeyondTrust Support.

For more information about the registry and about leaving the domain, see the following topics:

- Configuring PBIS with the Registry
- Leaving a Domain and Uninstalling the PBIS Agent

PBIS Agent

The PowerBroker Identity Services (PBIS) agent is installed on a Linux, Unix, or Mac OS X computer to connect it to Microsoft Active Directory and to authenticate users with their domain credentials. The agent integrates with the core operating system to implement the mapping for any application, such as the logon process (`/bin/login`), that uses the name service (NSS) or pluggable authentication module (PAM). As such, the agent acts as a Kerberos 5 client for authentication and as an LDAP client for authorization. In PBIS Enterprise, the agent also retrieves Group Policy Objects (GPOs) to securely update local configurations, such as the sudo file.

The following topics provide more information about the PBIS agent, also known as the PBIS client software.

Services

Prior to PowerBroker Identity Services 6.5, the agent was composed of separate daemon processes (with various dependencies between them), and each was started in sequence by the operating systems at boot up. In PowerBroker Identity Services 6.5, the daemons have been replaced by libraries loaded by the service manager daemon (`/opt/pbis/sbin/lwsmd`). Beginning in version 6.5, the service `lsass` replaces the daemon `lsassd`.

At boot time, the operating system is configured to start the service manager daemon. It is then instructed by the operating system (with the command `/opt/pbis/bin/lwsm autostart`) to start all desired services. The service manager daemon keeps track of which services have already been started and sees to it that all services are started and stopped in the appropriate order.

PBIS Open and PBIS Enterprise

Both the PBIS Open agent and the PBIS Enterprise agent are composed of the service manager daemon (`/opt/pbis/sbin/lwsmd`) and include the following services:

Service	Description	Dependencies

`lsass`	Handles authentication, authorization, caching, and idmap lookups. You can check its status or restart it.
	View a diagram of the Lsass architecture.
		`netlogon`
		`lwio`
		`rdr`
		`lwreg`
		Usually `eventlog` (Can be disabled after installation.)

Service	Description	Dependencies
		Sometimes `dcerpc` (Can be enabled after installation for registering TCP/IP endpoints of various services.)
`netlogon`	Detects the optimal domain controller and global catalog and caches them.	`lwreg`
`lwio`	An input-output service that is used to communicate through DCE-RPC calls to remote computers, such as during domain join and user authentication.	`lwreg`
`rdr`	A redirector that multiplexes connections to remote systems.	`lwio` `lwreg`
`dcerpc`	Handles communication between Linux, Unix, and Mac computers and Microsoft Active Directory by mapping data to end points. By default, it is disabled.	
`eventlog`	Collects and processes data for the local event log. Can be disabled.	
`lwreg`	The registry service that holds configuration information both about the services and information provided by the services.	
`reapsysl`	The syslog reaper that scans the syslog for events of interest and records them in the eventlog.	`eventlog`

PBIS Enterprise Only

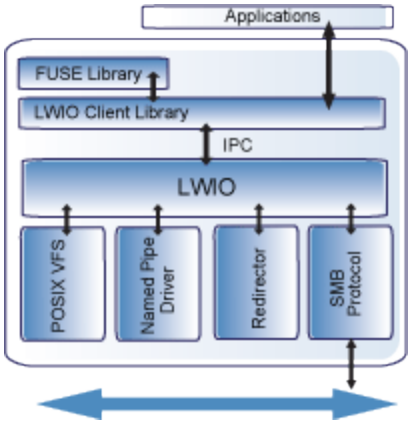
Additionally, PBIS Enterprise also includes the following services to apply Group Policy settings, handle smart cards, and monitor security events:

Service	Description	Dependencies
`gpagent`	Pulls Group Policy Objects (GPOs) from Active Directory and applies them to the computer.	`lsass, netlogon, lwio, rdr, lwreg, eventlog`

Service	Description	Dependencies
`eventfwd`	Forwards events from the local event log to a remote computer.	`eventlog`
`lwsc`	Smart card service.	`lwpkcs11`
`lwpkcs11`	Aids `lwsc` by supporting PKCS#11 API.	

PBIS Input-Output Service

The `lwio` service multiplexes input and output by using SMB1 or SMB2. The service's plugin-based architecture includes several drivers, the most significant of which is coded as `rdr`—the redirector.



The redirector multiplexes CIFS/SMB connections to remote systems. For instance, when two different processes on a local Linux computer need to perform input-output operations on a remote system by using CIFS/SMB, with either the same identity or different identities, the preferred method is to use the APIs in the `lwio` client library, which routes the calls through the redirector. In this example, the redirector maintains a single connection to the remote system and multiplexes the traffic from each client by using multiplex IDs.

The input-output service plays a key role in the PBIS architecture because PBIS makes heavy use of DCE/RPC, short for Distributed Computing Environment/Remote Procedure Calls. DCE/RPC, in turn, uses SMB: Thus, the DCE-RPC client libraries use the PBIS input-output client library, which in turn makes calls to `lwio` with Unix domain sockets.

When you join a domain, for example, PBIS uses DCE-RPC calls to establish the machine password. The PBIS authentication service periodically refreshes the machine password by using DCE-RPC calls. Authentication of users and groups in Active Directory takes place with Kerberos, not RPC. (View a data-flow diagram that shows how systems interact when you join a domain.)

In addition, when a joined computer starts up, the PBIS authentication service enumerates Active Directory trusts by using DCE-RPC calls that go through the redirector. With one-way trusts, the authentication service uses RPC to look up domain users, groups, and security identifiers. With two-way trusts, lookup takes place through LDAP, not RPC.

Because the authentication service registers trusts only when it starts up, you should restart `lsass` with the PBIS Service Manager after you modify a trust relationship.

The PBIS Group Policy agent also uses the input-output client library and the redirector when it copies files from the `sysvol` share of a domain controller.

To troubleshoot remote procedure calls that go through the input-output service and its redirector, use a Wireshark trace or a TCP dump to capture the network traffic. Wireshark, a free open-source packet analyzer, is recommended.

PAM Options

PowerBroker Identity Services uses three standard PAM options—`try_first_pass`, `use_first_pass`, and `use_authtok`—and adds three non-standard options to the PAM configuration on some systems: `unknown_ok`, `remember_chpass`, and `set_default_repository`. The `unknown_ok` option allows local users to continue down the stack (first line succeeds but second line fails) while blocking domain users who do not meet group membership requirements. On AIX systems, which have both PAM and LAM modules, the `remember_chpass` prevents the AIX computer from trying to change the password twice and prompting the user twice. On Solaris systems, the `set_default_repository` option is used to make sure password changes work as expected.

Managing the PBIS Services

The PBIS Service Manager lets you track and troubleshoot all the PBIS services with a single command-line utility. You can, for example, check the status of the services, view their dependencies, and start or stop them. The service manager is the preferred method for restarting a service because it automatically identifies a service's dependencies and restarts them in the correct order. In addition, you can use the service manager to set the logging destination and the log level.

To list status of the services, run the following command with superuser privileges at the command line:

/opt/pbis/bin/lwsm list

Example:

```
[root@rhel5d bin]# /opt/pbis/bin/lwsm list
lwreg       running (container: 1999)
dcerpc      stopped
eventlog    running (container: 2027)
lsass       running (container: 2049)
lwio        running (container: 2041)
netlogon    running (container: 2035)
rdr         running (io: 2041)
reapsysl    running (container: 2064)
```

After you change a setting in the registry, you must use the service manager to force the service to begin using the new configuration by executing the following command with super-user privileges. This example refreshes the `lsass` service:

/opt/pbis/bin/lwsm refresh lsass

PBIS Registry

Configuration information for the services is stored in the PBIS registry, which you can access and modify by using the registry shell or by executing registry commands at the command line. The registry shell is at `/opt/pbis/bin/regshell`. For more information, see Configuring the PBIS Services with the Registry.

Ports and Libraries

The agent includes a number of libraries in `/opt/pbis/lib` and uses certain ports for outbound traffic. For details about the ports, see Make Sure Outbound Ports Are Open.

View a data-flow diagram that shows how systems interact when you join a domain.

Caches and Databases

To maintain the current state and to improve performance, the PBIS authentication service (lsass) caches information about users and groups in memory. You can, however, change the cache to store the information in a SQLite database; for more information, see the chapter on configuring PBIS with the registry.

The PBIS site affinity service, `netlogon`, caches information about the optimal domain controller and global catalog in the PBIS registry.

The following files are in `/var/lib/pbis/db`:

File	Description
registry.db	The SQLite 3.0 database in which the PBIS registry service, lwreg, stores data.
sam.db	Repository managed by the local authentication provider to store information about local users and groups.
lwi_events.db	The database in which the event logging service, eventlog, records events.
lsass-adcache.db.fqdn	Cache managed by the Active Directory authentication provider to store user and group information. The file is in `/var/lib/pbis/db` only when you set the database type to be the non-default SQLite database. In the name of the file, FQDN is replaced by your fully qualified domain name.

Since the default UIDs that PBIS generates are large, the entries made by the operating system in the `lastlog` file when AD users log in make the file appear to increase to a large size. This is normal and should not cause concern. The lastlog file (typically `/var/log/lastlog`) is a sparse file that uses the UID and GID of the users as disk addresses to store the last login information. Because it is a sparse file, the actual amount of storage used by it is minimal.

With PBIS Open, you can manage the following settings for your cache by editing the PBIS registry. See Cache Settings in the lsass Branch.

- The Cache Type
- The Size of the Memory Cache
- The Duration of Cached Credentials
- The NSS Membership and NSS Cache Settings
- The Interval for Caching an Unknown Domain

With PBIS Enterprise, you can manage the settings with Group Policy settings; see the *PowerBroker Identity Services Group Policy Administration Guide*.

Additional information about a computer's Active Directory domain name, machine account, site affinity, domain controllers, forest, the computer's join state, and so forth is stored in the PBIS registry. Here is an example of the kind of information that is stored under the `Pstore` key and the `netlogon` key:

```
[HKEY_THIS_
MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory\DomainJoin\EXAMPL
"ClientModifyTimestamp"=dword:4b86d9c6
"CreationTimestamp"=dword:4b86d9c6
```

```
"DomainDnsName"="EXAMPLE.COM"
"DomainName"="EXAMPLE"
"DomainSID"="S-1-5-21-3190566242-1409930201-3490955248"
"HostDnsDomain"="example.com"
"HostName"="RHEL5D"
"MachineAccount"="RHEL5D$"
"SchannelType"=dword:00000002

[HKEY_THIS_MACHINE\Services\netlogon\cachedb\example.com-
0]
"DcInfo-ClientSiteName"="Default-First-Site-Name"
"DcInfo-DCSiteName"="Default-First-Site-Name"
"DcInfo-DnsForestName"="example.com"
"DcInfo-DomainControllerAddress"="192.168.92.20"
"DcInfo-DomainControllerAddressType"=dword:00000017
"DcInfo-DomainControllerName"="w2k3-r2.example.com"
"DcInfo-
DomainGUID"=hex:71,c1,9e,b5,18,35,f3,45,ba,15,05,95,fb,5b,62,e3
"DcInfo-Flags"=dword:000003fd
"DcInfo-FullyQualifiedDomainName"="example.com"
"DcInfo-LMToken"=dword:0000ffff
"DcInfo-NetBIOSDomainName"="EXAMPLE"
"DcInfo-NetBIOSHostName"="W2K3-R2"
"DcInfo-NTToken"=dword:0000ffff
"DcInfo-PingTime"=dword:00000006
"DcInfo-UserName"=""
"DcInfo-Version"=dword:00000005
"DnsDomainName"="example.com"
"IsBackoffToWritableDc"=dword:00000000
"LastDiscovered"=hex:c5,d9,86,4b,00,00,00,00
"LastPinged"=hex:1b,fe,86,4b,00,00,00,00
"QueryType"=dword:00000000
"SiteName"=""
```

Time Synchronization

For the PBIS agent to communicate over Kerberos with the domain controller, the clock of the client must be within the domain controller's maximum clock skew, which is 300 seconds, or 5 minutes, by default. (For more information, see http://web.mit.edu/kerberos/krb5-1.4/krb5-1.4.2/doc/krb5-admin/Clock-Skew.html.)

The clock skew tolerance is a server-side setting. When a client communicates with a domain controller, it is the domain controller's Kerberos key distribution center that determines the maximum clock skew. Since changing the maximum clock skew in a client's `krb5.conf` file does not affect the clock skew tolerance of the domain controller, the change will not allow a client outside the domain controller's tolerance to communicate with it.

The clock skew value that is set in the `/etc/pbis/krb5.conf` file of Linux, Unix, and Mac OS X computers is useful only when the computer is functioning as a server for other clients. In such cases, you can use a PBIS Group Policy setting to change the maximum tolerance; for more information, see Set the Maximum Tolerance for Kerberos Clock Skew in the *PowerBroker Identity Services Group Policy Administration Guide*.

The domain controller uses the clock skew tolerance to prevent replay attacks by keeping track of every authentication request within the maximum clock skew. Authentication requests outside the maximum clock skew are discarded. When the server receives an authentication request within the clock skew, it checks the replay cache to make sure the request is not a replay attack.

Using a Network Time Protocol Server

If you set the system time on your computer with a Network Time Protocol (NTP) server, the time value of the NTP server and the time value of the domain controller could exceed the maximum skew. As a result, you will be unable to log on your computer.

If you use an NTP server with a cron job, there will be two processes trying to synchronize the computer's time—causing a conflict that will change the computer's clock back and forth between the time of the two sources.

It is recommended that you configure your domain controller to get its time from the NTP server and configure the domain controller's clients to get their time from the domain controller.

Automatic Detection of Offline Domain Controller and Global Catalog

The PBIS authentication service—`lsass`—manages site affinity for domain controllers and global catalogs and caches the information with `netlogon`. When a computer is joined to Active Directory, `netlogon` determines the optimum domain controller and caches the information. If the primary domain controller goes down, `lass` automatically detects the failure and switches to another domain controller and another global catalog within a minute.

However, if another global catalog is unavailable within the forest, the PBIS agent will be unable to find the Unix and Linux information of users and groups. The PBIS agent must have access to the global catalog to function. Therefore, it is a recommended that each forest has redundant domain controllers and redundant global catalogs.

UID-GID Generation in PowerBroker Cells

In PBIS Open, a UID and GID are generated by hashing the user or group's security identifier, or SID, from Active Directory. With PBIS Open, you do not need to make any changes to Active Directory. A UID and GID stays the same across host machines. With PBIS Open, you cannot set UIDs and GIDs for Linux and Unix in Active Directory; using AD to set and manage UIDs and GIDs is a feature of PBIS Enterprise or the PBIS UID-GID management tool.

If your Active Directory relative identifiers, or RIDs, are a number greater than 524,287, the PBIS Open algorithm that generates UIDs and GIDs can result in UID-GID collisions among users and groups. In such cases, it is recommended that you use PBIS Enterprise or the PBIS UID-GID management tool.

The PBIS Open algorithm is the same in 4.1 and 5.0, and if you are running 4.1 on one computer and 5.0 or later on another, each user and group should have the same UID and GID on both machines.

Note: If you have UIDs and GIDs defined in Active Directory, PBIS Open will not use those UIDs and GIDs.

In PBIS Enterprise, you can specify the UIDs and GIDs that you want, including setting multiple UID and GID values for a given user based on OU membership by using PowerBroker cells. (PowerBroker cells, available only in PBIS Enterprise, provide a method for mapping Active Directory users and groups to UIDs and GIDs.) You can also set PBIS Enterprise to automatically generate UID and GID values sequentially.

Cached Credentials

Both PBIS Open and PBIS Enterprise cache credentials so users can log on when the computer is disconnected from the network or Active Directory is unavailable.

Trust Support

The PBIS agent supports the following Active Directory trusts:

Trust Type	Transitivity	Direction	PBIS Default Cell Support	PBIS Non-Default Cell Support (Named Cells)
Parent and child	Transitive	Two-way	Yes	Yes

Trust Type	Transitivity	Direction	PBIS Default Cell Support	PBIS Non-Default Cell Support (Named Cells)
External	Nontransitive	One-way	No	Yes
External	Nontransitive	Two-way	No	Yes
Forest	Transitive	One-way	No	Yes
Forest	Transitive	Two-way	Yes: Must enable default cell in both forests.	Yes

There is information on the types of trusts at http://technet.microsoft.com/en-us/library/cc775736(WS.10).aspx.

Working with Trusts

The following is general information about working with trusts.

- You must place the user or group that you want to give access to the trust in a cell other than the default cell.
- In a two-way forest or parent-child trust, PBIS merges the default cells. When merged, users in one domain can log on computers in another domain, and vice-versa.
- To put a user in a child domain but not the parent domain, you must put the user in a non-default cell, which is a cell associated with an organizational unit.
- If there is a UID conflict across two domains, one domain will be dropped.
- In a cross-forest transitive one- or two-way trust, the root of the trusted forest must have a default cell.
- In a one-way trust in which Forest A trusts Forest B, a computer in Forest A cannot get group information from Forest B, because Forest B does not trust Forest A. The computer in Forest A can obtain group information if the user logs on with a password for a domain user, but not if the user logs on with Kerberos single sign-on credentials. Only the primary group information, not the secondary group information, is obtained.

- To support a 1-way trust without duplicating user accounts, you must use a cell associated with an OU, not a default cell. If Domain A trusts Domain B (but not the reverse) and if Domain B contains all the account information in cells associated with OUs, then when a user from Domain B logs on a machine joined to Domain A, Domain B will authenticate the user and authorize access to the machine in Domain A.
  In such a scenario, you should also add a domain user from the trusted domain to an administrative group in the trusting domain so you can manage the trusting domain with the appropriate level of read access to trusted user and group information. However, before you add the domain user from the trusted domain to the trusting domain, you must first add to the trusting domain a group that includes the user because Unix and Linux computers require membership in at least one group and Active Directory does not enumerate a user's membership in foreign groups.
- If you have a network topology in which the "front" domain trusts the "back" domain, and you join a machine to the front domain using a back domain administrator, as in the following example, the attempt to join the domain will fail: `domainjoin-cli join front.example.com back\\administrator password`. However, the attempt to join the domain will succeed if you use the following nomenclature: `domainjoin-cli join front.example.com administrator@BACK.example.COM password`
- With PBIS Enterprise, aliased user names are supported in the default cell and in named cells.

Trusts and Cells in PBIS Enterprise

In PBIS Enterprise, a cell contains Unix settings, such as a UID and a GID, for an Active Directory user. When an AD user logs on a PBIS client, PBIS Enterprise searches Active Directory for the user's cell information—and must find it to operate properly. Thus, your AD topology and your trust relationships may dictate where to locate a cell in Active Directory so that your PBIS clients can access their Unix settings.

With a default cell, PBIS searches for a user or group's attributes in the default cell of the domain where the user or group resides. In a multi-domain topology, a default cell must exist in the domain where user and group objects reside in addition to the default cell that exists in the domain to which Unix, Linux, and Mac computers are joined. In a multi-domain topology, then, be sure to create a default cell in each domain.

Ideally, Unix information is stored on the user object in default cell schema mode. If the client computer does not have the access rights to read and write the information to the user object, as in an external one-way trust, the Unix information cannot be stored on the user object. It can, however, be stored locally in a named cell, that is, a cell associated with an organizational unit.

Since a named cell can be linked to the default cell, you can store Unix information on the user object in default cell schema mode when possible, and otherwise in a named cell that represents the external user. For information about cells, see the chapter on planning your PBIS Enterprise installation and deployment.

Integrating with Samba

PowerBroker Identity Services includes a tool to install the files necessary to use Samba with PBIS. Located in `/opt/pbis/bin`, the tool is named `samba-interop-install`. The *PowerBroker Identity Services Samba Guide* describes how to use the tool to integrate Samba 3.0.25, 3.2.X, or 3.5.X with PBIS Enterprise or PBIS Open.

Supported Platforms

PBIS Open and PBIS Enterprise run on a broad range of Unix, Mac OS X, and Linux platforms. BeyondTrust frequently adds new vendors and distributions. See the BeyondTrust website for the list of supported platforms.

Configuring Clients Before PBIS Agent Installation

Before you install the PBIS agent, you should configure client computers as indicated in the following topics.

Configure nsswitch.conf

Before you attempt to join an Active Directory domain, make sure the `/etc/nsswitch.conf` file contains the following line:

```
hosts: files dns
```

The `hosts` line can contain additional information, but it must include the `dns` entry, and it is recommended that the `dns` entry appear after the `files` entry.

Computers running Solaris, in particular, may not contain this line in `nsswitch.conf` until you add it.

When you use PowerBroker Identity Services with Multicast DNS 4 (mDNS4) and have a domain in your environment that ends in `.local`, you must place the `dns` entry before the `mdns4_minimal` entry and before the `mdns4` entry:

```
hosts: files dns mdns4_minimal [NOTFOUND=return] mdns4
```

The default setting for many Linux systems is to list the `mdns4` entries before the `dns` entry—a configuration that leaves PBIS unable to find the domain.

Important: For PBIS to process changes to your `nsswitch.conf` file, you must restart the PBIS input-output service (`lwio`) and the authentication service (`lsass`). Running the following command as root restarts both services:

```
/opt/pbis/bin/lwsm restart lwio
```

For PBIS to work correctly, the `nsswitch.conf` file must be readable by user, group, and world.

For more information on configuring nsswitch, see the man page for `nsswitch.conf`.

Configure resolv.conf

Before you attempt to join an Active Directory domain, make sure that `/etc/resolv.conf` on your Linux, Unix, or Mac client includes a DNS server that can resolve SRV records for your domain.

Example:

```
[root@rhel5d Desktop]# cat /etc/resolv.conf
```

```
search example.com
nameserver 192.168.100.132
```

For more information on `resolv.conf`, see your operating system's man page.

Configure Firewall Ports

The PBIS agent requires several firewall ports to be open for outbound traffic. For a list of the required ports, see Make Sure Outbound Ports Are Open.

Extend Partition Size (IBM AIX)

On AIX 5.2 and 5.3, you may need to extend the size of certain partitions to be able to complete the installation.

To do so, use IBM's `chfs` command to change the partition sizes—for example:

```
# chfs -a size=+200M /opt
```

This command increases the size of the `opt` partition by 200 megabytes, which should be sufficient for a successful installation.

Increase Max Username Length (IBM AIX)

By default, IBM AIX is not configured to support long user and group names, which might present a conflict when you try to log on with a long Active Directory username. On AIX 5.3 and AIX 6.1, the symptom is that group names, when enumerated through the `groups` command, are truncated.

To increase the max username length on AIX 5.3, use the following syntax:

```
# chdev - l sys0 -a max_logname=MaxUserNameLength+1
```

Example:

```
# chdev - l sys0 -a max_logname=255
```

This command allocates 254 characters for the user and 1 for the terminating null.

The safest value that you can set `max_logname` to is `255`.

You must reboot for the changes to take effect:

```
# shutdown - Fr
```

Note: AIX 5.2 does not support increasing the maximum user name length.

Check System Health

Members of the BeyondTrust support staff may use a shell script to check the health of a Linux or Unix computer on which you plan to install the PBIS agent. The script helps identify potential system configuration issues before you install the agent and attempt to join a Linux or Unix computer to Active Directory.

With PBIS Open, the script is unavailable, but you can manually check your computer against the list in the table below.

The name of the script is `healthchk.sh`. To execute it, copy the script to the Unix or Linux computer that you want to check, and then execute the following command from the shell prompt: `pbis-health-check.sh`

The script outputs the results of its scan to `/tmp/healthchk.out`.

The following table lists each item the script checks, describes the item, and suggests action to correct the issue.

Item Checked	Description	Corrective Action
Type of operating system	The operating system must be one of the platforms that PBIS supports. Supported platforms are listed later in this guide.	Install the agent on a computer that is running a supported operating system.
Hostname	Informational.	Not applicable.
Processor type	The processor type must be supported by the PBIS Agent. See the list of supported platforms later in this guide.	Install the agent on a computer with a supported processor.
Disk usage	Checks the disk space available to `/opt` to ensure that there is enough to install the agent and its accompanying packages.	Increase the amount of disk space available to `/opt`.

Item Checked	Description	Corrective Action
Contents of `/etc/*release` (for AIX, to determine the `oslevel`)	Displays the operating system and version number to ensure that they are supported by PBIS. See the list of supported platforms later in this guide.	Install the agent on a computer that is running a supported operating system and version.
Network interface and its status	Displays network interfaces and IP addresses to ensure that the system has network access.	Configure the computer so that it has network access and can communicate with the domain controller.
Contents of the IP routing table	To determine whether a single default gateway is defined for the computer.	If the computer does not use a single default gateway, you must define a route to a single default gateway. For example, you can run the `route -n` to view the IP routing table and set a static route. For more information, see the man pages for your system. On Solaris, you may need to create or edit `/etc/defaultrouter`. On Linux, you can set the default gateway by running the network utility for your distribution.
Connectivity to the default gateway	Pings the default gateway to ensure that the computer can connect to it. A connection to the default gateway is required.	Configure the computer and the network so that the computer can connect to the default gateway.
Contents of `nsswitch.conf` (or, for AIX, `netsvc.conf`)	Displays information about the nsswitch configuration.	The `nsswitch.conf` file must contain the following line: `hosts: files dns` Computers running Solaris, in particular, may not contain this line in `nsswitch.conf`.

Item Checked	Description	Corrective Action

FQDN	Determines the fully qualified domain name of the computer to ensure that it is set properly.	Make sure the computer's FQDN is correct in `/etc/hosts`.
		You can determine the fully qualified domain name of a computer running Linux, Unix, or Mac OS X by executing the following command:
		`ping - c 1 `hostname``
		On HP-UX:
		`ping `hostname` -n 1`
		On Solaris:
		`FQDN=`/usr/lib/mail/sh/check-hostname|cut -d" " -f7`;echo $FQDN`
		This command prompts the computer to look up the primary host entry for its hostname. In most cases, it looks for its hostname in `/etc/hosts`, returning the first FQDN name on the same line. So, for the hostname `qaserver`, here is an example of a correct entry in `/etc/hosts`:
		`10.100.10.10 qaserver.corpqa.example.com qaserver`
		If, however, the entry in `/etc/hosts` incorrectly lists the hostname (or anything else) before the FQDN, the computer's FQDN becomes, using the malformed example below, `qaserver`:
		`10.100.10.10 qaserver qaserver.corpqa.example.com`

Item Checked	Description	Corrective Action
		If the host entry cannot be found in `/etc/hosts`, the computer looks for the results in DNS instead. This means that the computer must have a correct A record in DNS. If the DNS information is wrong and you cannot correct it, add an entry to `/etc/hosts`.
IP address of local NIC	Determines whether the IP address of the local network card matches the IP address returned by DNS for the computer. The IP address of the local NIC must match the IP address for the computer in DNS.	Either update DNS or change the local IP address so that the IP address of the local network card matches the IP address returned by DNS for the computer.
Contents of `resolv.conf`	Returns the address for the `nameserver` set in `resolv.conf`. The address of `nameserver` must point to a DNS server that can resolve the Active Directory domain name and return the SRV records for the domain controllers. The SRV record is a DNS resource record that is used to identify computers that host specific services. SRV resource records are used to locate domain controllers for Active Directory.	Compare against the results of the items checked next.

Item Checked	Description	Corrective Action
DNS query results for system (hostname and IP)	The IP address for the host name from DNS must match the IP address of the computer's local NIC.	Either update DNS or change the local IP address so that the IP address of the local network card matches the IP address returned by DNS for the computer.
DNS name resolution and connectivity to specified domain controller	Pings the domain name to get the IP address.	Correct `resolv.conf` so that the `nameserver` points to a DNS server that can resolve the Active Directory domain name—typically the domain controller running DNS.
SRV records from DNS	Performs a DNS lookup for the SRV records to get the IP addresses for the domain controller.	Correct `resolv.conf` so that the `nameserver` points to a DNS server that can resolve the SRV records.
Connectivity to the Internet	Informational. Although connectivity to the Internet is optional, it makes it easier to download the installer for the agent installer.	Not applicable.
Location and version information for sudo, openssl, bash, rpm, and ssh	Checks whether required utilities are installed and are in expected locations.	PBIS requires the following utilities: ssh and openssl. The other utilities are optional but may be useful.
Selected firewall settings (Kerberos, NetBIOS, and LDAP)	Tests whether the computer can connect to ports on the domain controller to make sure that a firewall will not block the computer's attempt to join the domain.	Reconfigure the firewall to allow the computer to access the domain controller.
Listing of files in `/etc/pam.d`	Lists other software that requires PAM.	Not applicable. Save this information for BeyondTrust support staff in case they need to troubleshoot the installation.

Item Checked	Description	Corrective Action
Contents of selected pam files (pam.conf, common-auth, system-auth)	May reveal installation of other applications that are incompatible with the installer.	Not applicable. Save this information for BeyondTrust support staff in case they need to troubleshoot the installation.
Contents of `/etc/krb5.conf`	Shows Kerberos 5 configuration.	Not applicable. Save this information for BeyondTrust support staff in case they need to troubleshoot the installation.
DHCP	Checks whether DHCP is in use. When the PBIS Agent joins the computer to the domain, the agent restarts the computer. DHCP can then change the contents of `/etc/resolv.conf`, `/etc/hosts`, and other files, causing the computer to fail to join the domain.	Set the computer to a static IP address or configure DHCP so that it does not update such files as `/etc/resolv.conf` and `/etc/hosts`.
ISA type	Returns 32-bit or 64-bit information.	Use the installer for your ISA type.
Read-only filespaces	Checks whether `/opt` is mounted as readonly.	Make sure that `/opt` is writable.
AIX TL levels	Determines the AIX TL level.	Not all TL levels are supported. For AIX, check with BeyondTrust support to make sure that PBIS is compatible with the TL level you are using.

Installing the PBIS Agent

You must install the PBIS agent—the identity service that authenticates users—on each Linux, Unix, or Mac OS X computer that you want to connect to Active Directory. To obtain the installer or to view a list of supported platforms, see www.beyondtrust.com. You can download the PBIS Open installation package for free from the BeyondTrust website. If you are using PBIS Enterprise, make sure you install the PBIS Enterprise version of the agent.

Important: Before you install the agent, it is recommended that you upgrade your system with the latest security patches. Patch requirements for Unix systems are listed below.

The procedure for installing the PBIS Open agent or the PBIS Enterprise agent depends on the operating system of your target computer or virtual machine. Each procedure is documented in a separate section of this chapter.

Operating System	Procedure by Title
Linux platforms running kernel release number 2.6 or later are supported by PBIS 6.1 or later. Linux platforms runing kernel release number 2.4 or later are supported by PBIS 6.0 or earlier.	Install the Agent on Linux or Unix with the Shell Script
Unix: Sun Solaris, HP-UX, IBM AIX	Install the Agent on Unix with the Command Line
VMware ESX 3.0 and 3.5 (hypervisor)	Install the Agent on Linux or Unix with the Shell Script
Mac OS X 10.4 or later, including 10.5 and 10.6	Install the Agent on a Mac Computer

You also have the option of installing the agent in unattended mode; see Install the Agent on Linux in Unattended or Text Mode and Install the Agent on a Mac in Unattended Mode.

Checking Your Linux Kernel Release Number

To determine the release number of the kernel on your Linux machine, run the following command:

uname -r

For the Linux machine to be supported by PBIS, the kernel release number must be 2.6 or later.

Package Management Commands

For an overview of commands such as `rpm` and `dpkg` that can help you manage PBIS on Linux and Unix platforms, see *PowerBroker Identity Services Package Management Commands*.

Requirements for the Agent

This section lists requirements for installing and running the PBIS agent. Requirements for the BeyondTrust Management Console, which is part of PBIS Enterprise, are detailed in the chapter on installing the console. PBIS Open does not include the BeyondTrust Management Console, and that chapter is not included in the *PowerBroker Identity Services Open Installation and Administration Guide*.

Before you install the PBIS agent, make sure that the following environmental variables are not set: `LD_LIBRARY_PATH`, `LIBPATH`, `SHLIB_PATH`, `LD_PRELOAD`. Setting any of these environmental variables violates best practices for managing Unix and Linux computers because it causes PBIS to use non-PBIS libraries for its services. For more information on best practices, see http://linuxmafia.com/faq/Admin/ld-lib-path.html. PBIS does not support installations that use these environmental variables. If joining the domain fails with an error message that one of these environmental variables is set, stop all the PBIS services, clear the environmental variable, make sure it is not automatically set when the computer restarts, and then try to join the domain again.

If you must set `LD_LIBRARY_PATH`, `LIBPATH`, or `SHLIB_PATH` for another program, put the PBIS library path (`/opt/pbis/lib` or `/opt/pbis/lib64`) before any other path—but keep in mind that doing so may result in side effects for your other programs, as they will now use PBIS libraries for their services.

Patch Requirements

It is recommended that you apply the latest patches for your operating system before you install PBIS. Known patch requirements are listed below.

Sun Solaris

All Solaris versions require the md5sum utility, which can be found on the companion CD.

Sun Solaris 10 requires update 5 or later. The Solaris 10 05/08 (or later) patch bundle is available at http://www.sun.com/. Solaris 10_x86 requires the patch for nscd, either patch ID number 138047-02 or the patch that supercedes it, number 138264-02. This patch available for SPARC as patch 138046.

Solaris 8 Sparc should be fully patched according to Sun's recommendations. PBIS depends on the latest patch for `libuuid`. On Sparc systems, the patch for `libuuid` is 115831. Sun patch 110934-28 for Solaris 5.8 is also required for Solaris 8.

Solaris 8 Intel systems also require the latest patch for libuuid: `115832-01`. Sun patches 110403-06 and 110935-26 are also required. Patch 110403-06 must be installed before you install patch 110935-26.

Solaris 9 requires Sun patch 113713-28 for Solaris 5.9.

OpenSolaris is compatible with PBIS without any patches.

HP-UX

Secure Shell: For all HP-UX platforms, it is recommended that a recent version of HP's Secure Shell be installed. It is recommended that you use HP-UX Secure Shell A.05.00.014 or later.

Sudo: By default, the versions of sudo available from the HP-UX Porting Center do not include the Pluggable Authentication Module, or PAM, which PBIS requires to allow domain users to execute sudo commands with super-user credentials. It is recommended that you download sudo from the HP-UX Porting Center and make sure that you use the `with-pam` configuration option when you build it.

HP-UX 11iv1 requires the following patches: PHCO_36229, PHSS_35381, PHKL_34805, PHCO_31923, PHCO_31903, and PHKL_29243. Although these patches may be superceded by subsequent patches, these patches represent the minimum patch level for proper operation.

Kerberos client libraries: For single sign-on with HP-UX 11.11 and 11.23, you must download and install the latest KRB5-Client libraries from the HP Software Depot. (By default, HP-UX 11.31 includes the libraries.)

Other Requirements for the Agent

AIX

On AIX computers, PAM must be enabled. LAM is supported only on AIX 5.x. PAM must be used exclusively on AIX 6.x.

Secure Shell

To properly process logon events with PBIS, your SSH server or client must support the `UsePam yes` option. For single sign-on, both the SSH server and the SSH client must support GSSAPI authentication.

Other Software

Telnet, rsh, rcp, rlogin, and other programs that uses PAM for processing authentication requests are compatible with PBIS.

Networking Requirements

Each Unix, Linux, or Mac computer must have fully routed network connectivity to all the domain controllers that service the computer's Active Directory site. Each computer must be able to resolve A, PTR, and SRV records for the Active Directory domain, including at least the following:

- `A domain.tld`
- `SRV _kerberos._tcp.domain.tld`
- `SRV _ldap._tcp.domain.tld`
- `SRV _kerberos._udp.sitename.Sites._msdcs.domain.tld`
- `A domaincontroller.domain.tld`

In addition, several ports must be open; see Make Sure Outbound Ports Are Open.

Disk Space Requirements

The PBIS agent requires 100 MB of disk space in the `/opt` mount point. The agent also creates configuration files in `/etc/pbis` and offline logon information in `/var/lib/pbis`. In addition, the PBIS Enterprise agent caches Group Policy Objects (GPOs) in `/var/cache/pbis`.

Memory and CPU Requirements

The agent consists of several services and daemons that typically use between 9 MB and 14 MB of RAM. Memory utilization of the authentication service on a 300-user mail server is typically 7 MB; the other services and daemons require between 500 KB and 2 MB each. CPU utilization on a 2.0 gigahertz single-core processor under heavy load with authentication requests is about 2 percent. For a description of the PBIS services and daemons, see PBIS Agent.

Clock Skew Requirements

For the PBIS agent to communicate over Kerberos with the domain controller's Kerberos key distribution center, the clock of the client must be within the domain controller's maximum clock skew, which is 300 seconds, or 5 minutes, by default. For more information on time synchronization, see PBIS Agent.

Install the Agent on Linux or Unix with the Shell Script

You install the PBIS Enterprise agent by using a shell script that contains a self-extracting executable. The file name of the SFX installer ends in `sh`. Example: `PBISEnterprise-6.5.0.3499-linux-i386-rpm.sh`.

The examples shown are for Linux RPM-based platforms. For other Linux and Unix platforms—such as Debian, HP-UX, AIX, and Solaris—simply substitute the right installer. The installer's name includes the product name, version and build numbers, operating system, computer type, and platform type.

Perform the following procedure with the **root** account. To view information about the installer or to view a list of command-line options, run the following command, replacing `6.5.0.3499` with the version and build number indicated in the file name of the SFX installer that you have:
`./PBISEnterprise-6.5.0.3499-linux-i386-rpm.sh --help`

After the wizard finishes, the user interface for joining a domain appears. To suppress it, you can run the installer with its `--dont-join` argument. In the following procedure, replace `6.5.0.3499` with the version and build number indicated in the file name of the SFX installer that you have available.

1. Download or copy the shell script to your Linux or Unix computer's desktop.
   Important: If you FTP the file to the desktop of the target Linux or Unix computer, you must select binary, or BIN, for the transfer. Most FTP clients default to AUTO or ASCII, but the installer includes some binary code that becomes corrupted in AUTO or ASCII mode.

2. Change directories to the desktop.

3. As root, change the mode of the installer to executable.
   `chmod a+x PBISEnterprise-6.5.0.3499-linux-i386-rpm.sh`

   On Ubuntu, execute the `sudo` command before you execute the `chmod` command:

   `sudo chmod a+x PBISEnterprise-6.5.0.3499-linux-i386-rpm.sh`

4. As root, run the installer:
   `./PBISEnterprise-6.5.0.3499-linux-i386-rpm.sh`

5. Follow the instructions in the installer.
   Note: On SLES and other systems on which the pager is set to less, you must exit the end user license agreement, or EULA, by typing the following command: `q`

Install the Agent on Linux in Unattended Mode

You can install the agent in unattended mode by using the `install` command. Replace `6.5.0.3499` with the version and build number indicated in the file name of the SFX installer that you have available.

`./PBISEnterprise-6.1.0.67-linux-i386-rpm.sh install`

Install the Agent on Unix with the Command Line

You install the PBIS Open agent or the PBIS Enterprise agent on Sun Solaris, HP-UX, and IBM AIX by using a shell script that contains a self-extracting executable—an SFX installer with a file name that ends in `sh`. Example: `PBISEnterprise-6.5.0.70-solaris-sparc-pkg.sh`.

The examples shown below are for Solaris Sparc systems. For other Unix platforms, simply substitute the right installer. The installer's name includes the product name, version and build numbers, operating system, computer type, and platform type.

Note: The name of a Unix installer for PBIS Enterprise on installation media might be truncated to an eight-character file name with an extension. For example, `l3499sus.sh` is the truncated version of `PBISEnterprise-6.5.0.3499-solaris-sparc-pkg.sh`.

Perform the following procedure with the root account. Replace `6.5.0.70` with the version and build number indicated in the file name of the SFX installer that you have available.

1. Download or copy the installer to the Unix computer's desktop.
2. Change directories to the desktop.

3. As root, change the mode of the installer to executable:
   `chmod a+x PBISEnterprise-6.5.0.70-solaris-sparc-pkg.sh`

   > Tip:
   >
   > To view a list of command-line options, run the following command:
   >
   > `./PBISEnterprise-6.5.0.70-solaris-sparc-pkg.sh --help`

4. As root, run the installer:
   `./PBISEnterprise-6.5.0.70-solaris-sparc-pkg.sh`

5. Follow the instructions in the installer.

Install the Agent on a Mac OS X Computer

To install the PBIS agent on a computer running Mac OS X, you must have administrative privileges on the Mac. PBIS supports Mac OS X 10.4 or later.

1. Obtain the PBIS agent installation package for your Mac from BeyondTrust Software, Inc., and save it to your desktop.
2. Log on to the Mac with a local account that has administrative privileges.
3. On the **Apple** menu , click **System Preferences**.
4. Under **Internet & Network**, click **Sharing**, and then select the **Remote Login** check box. Turning on Remote Login lets you access the Mac with SSH after you install PBIS.
5. On the Mac computer, go to the Desktop and double-click the PBIS `.dmg` file.
6. In the Finder window, double-click the PBIS `.mpkg` file.
7. Follow the instructions in the installation wizard.

When the wizard finishes installing the package, you are ready to join the Mac computer to an Active Directory domain.

Install the Agent on a Mac in Unattended Mode

The PBIS command-line tools can remotely deploy the shell version of the PBIS agent to multiple Mac OS X computers, and you can automate the installation of the agent by using the installation command in unattended mode.

The commands in this procedure require administrative privileges. Replace `6.5.0.3628` with the version and build number indicated in the file name of the SFX installer that you have available.

1. Use SSH to connect to the target Mac OS X computer and then use SCP to copy the `.dmg` installation file to the desktop of the Mac or to a location that can be accessed remotely. The rest of this procedure assumes that you copied the installation file to the desktop.
2. On the target Mac, open Terminal and then use the `hdiutil mount` command to mount the `.dmg` file under `Volumes`:
   ```
   /usr/bin/hdiutil mount Desktop/PBISEnterprise-6.1.0.3628.dmg
   ```
3. Execute the following command to open the `.mpkg` volume:
   ```
   /usr/bin/open Volumes/PBISEnterprise-6.5.0.3628
   ```
4. Execute the following command to install the agent:
   ```
   sudo installer -pkg /Volumes/PBISEnterprise-6.5.0.3628/PBISEnterprise-6.5.0.3628.mpkg -target LocalSystem
   ```

   Note: For more information about the `installer` command, in Terminal execute the following command:

   ```
   man installer
   ```
5. To join the domain, execute the following command in the Terminal, replacing `domainName` with the FQDN of the domain that you want to join and `joinAccount` with the user name of an account that has privileges to join computers to the domain:
   ```
   sudo /opt/pbis/bin/domainjoin-cli join domainName joinAccount
   ```

   Example: `sudo /opt/pbis/bin/domainjoin-cli join example.com Administrator`

   Terminal prompts you for two passwords: The first is for a user account on the Mac that has admin privileges; the second is for the user account in Active Directory that you specified in the join command.

   Note: You can also add the password for joining the domain to the command, but it is recommended that you do not use this approach because another user could view and intercept the full command that you are running, including the password:

   ```
   sudo /opt/pbis/bin/domainjoin-cli join domainName joinAccount joinPassword
   ```

   Example: `sudo /opt/pbis/bin/domainjoin-cli join example.com Administrator YourPasswordHere`

Install the Agent in Solaris Zones

Solaris Zones are a virtualization technology created by Sun Microsystems to consolidate servers. Primarily used to isolate an application, Solaris Zones act as isolated virtual servers running on a single operating system, making each application in a collection of applications seem as though it is running on its own server. A Solaris Container combines system resource controls with the virtual isolation provided by zones.

Every zone server contains a global zone that retains visibility and control in any installed non-global zones. By default, the non-global zones share certain directories, including `/usr`, which are mounted read-only. The shared directories are writable only for the global zone.

By default, installing PBIS in the global zone results in it being installed in all the non-global zones. You can, however, control the target of the installation by using the following options of the SFX installer. Replace `6.5.0.97` with the version and build number indicated in the file name of the SFX installer that you have available.

```
./PBISEnterprise-6.5.0.97-solaris-i386-pkg.sh --help
...
--all-zones           (Solaris) Install to all zones
(default)
--current-zone        (Solaris) Install only to current
zone
```

After a new child zone is installed, booted, and configured, you must run the following command as root to complete the installation:

/opt/pbis/bin/postinstall.sh

You cannot join zones to Active Directory as a group. Each zone, including the global zone, must be joined to the domain independently of the other zones.

Caveats

There are some caveats when using PBIS with Solaris Zones:

- When you join a non-global zone to AD, you will receive an error as PBIS attempts to synchronize the Solaris clock with AD. The error occurs because the root user of the non-global zone does not have root access to the underlying global system and thus cannot set the system clock. If the clocks are within the 5-minute clock skew permitted by Kerberos, the error will not be an issue. Otherwise, you can resolve the issue by manually setting the clock in the global zone to match AD or by

joining the global zone to AD before joining the non-global zone.

- Some Group Policy settings may log PAM errors in the non-global zones even though they function as expected. The cron Group Policy setting is one example:

```
Wed Nov 7 16:26:02 PST 2009 Running Cronjob 1 (sh)
Nov 7 16:26:01 zone01 last message repeated 1 time
Nov 7 16:27:00 zone01 cron[19781]: pam_lsass(cron):
request failed
```

  Depending on the Group Policy setting, these errors may result from file access permissions, attempts to write to read-only directories, or both.

- By default, Solaris displays `auth.notice` syslog messages on the system console. Some versions of PBIS generate significant authentication traffic on this facility-priority level, which may lead to an undesirable amount of chatter on the console or clutter on the screen.

To redirect the traffic to a file instead of displaying it on the console, edit your `/etc/syslog.conf` file as follows:

Change this:

```
*.err;kern.notice;auth.notice /dev/sysmsg
```

To this:

```
*.err;kern.notice /dev/sysmsg
auth.notice /var/adm/authlog
```

Important: Make sure that you use **tabs**, not spaces, to separate the facility.priority information (on the left) from the action field (on the right). Using spaces will cue syslog to ignore the entire line.

Upgrading Your Operating System

Before you upgrade your operating system, you must leave the domain, uninstall the domain join GUI, and uninstall the agent. Then, make sure you are using the correct agent for the new version of your operating system, install it, and rejoin the domain.

If, for example, you plan to upgrade your operating system from Mac OS X 10.5 (Leopard) to Mac OS X 10.6 (Snow Leopard), you must first leave the domain and uninstall the current agent. Then, after upgrading your operating system, install the correct agent for the new version of the operating system and join the domain again. See Uninstall the Agent on a Mac.

Joining an Active Directory Domain

When PBIS joins a computer to an Active Directory domain, it uses the hostname of the computer to create the name of the computer object in Active Directory. From the hostname, the PBIS domain join tool attempts to derive a fully qualified domain name. By default, the PBIS domain join tool creates the Linux and Unix computer accounts in the default Computers container in Active Directory.

You can, however, choose to pre-create computer accounts in Active Directory before you join your computers to the domain. When you join a computer to a domain, PBIS associates the computer with the pre-existing computer account when PBIS can find it. To locate the computer account, PBIS first looks for a computer account with a DNS hostname that matches the hostname of the computer. If the DNS hostname is not set, PBIS then looks for the name of a computer account that matches the computer's hostname, but only when the computer's hostname is 15 characters or less. Therefore, when the hostname of your computer is more than 15 characters, you should set the DNS hostname for the computer account to ensure that the correct computer account is found. If no match is found, PBIS creates a computer account.

The location of the domain join command-line utility is as follows:

/opt/pbis/bin/domainjoin-cli

After you join a domain for the first time, you must restart the computer before you can log on. If you cannot restart the computer, you must restart each service or daemon that looks up users or groups through the standard nsswitch interface, which includes most services that authenticate users, groups, or computers. You must, for instance, restart the services that use Kerberos, such as `sshd`.

For Linux computers, there is an optional graphical version of the PBIS domain join tool. It is installed on Linux platforms that are running GTK+ version 2.6 or later. For more information, see Join a Linux Computer to Active Directory with the GUI.

Important: On Linux computers running NetworkManager—which is often used for wireless connections—you must make sure before you join a domain that the computer has a non-wireless network connection and that the non-wireless connection is configured to start when the networking cable is plugged in. You must continue to use the non-wireless network connection during the post-join process of restarting your computer and logging on for the first time with your Active Directory domain credentials. For more information, see NetworkManager: Use a Wired Connection to Join a Domain.

Privileges and Permissions

To join a computer to a domain, you must have the user name and password of an Active Directory account that has privileges to join computers to the domain and the full name of the domain that you want to join. For instructions on how to delegate rights to join a computer to a domain, see http://support.microsoft.com/kb/932455. The level of privileges that you need is set by Microsoft Active Directory and is typically the same as performing the corresponding action on a Windows computer.

For more information about Active Directory privileges, permissions, and security groups, see the following references on the Microsoft TechNet website:

- Active Directory Privileges
- Active Directory Object Permissions
- Active Directory Users, Computers, and Groups
- Securing Active Directory Administrative Groups and Accounts

Removing a Computer from a Domain

You can remove a computer from the domain either by removing the computer's account from Active Directory Users and Computers or by running the domain join tool on the Unix, Linux, or Mac OS X computer that you want to remove; see Leave a Domain.

Creation of Local Accounts

After you join a domain, PBIS creates two local user accounts in the following form: `ComputerName\Administrator` and `ComputerName\Guest`. The administrator account is disabled until you enable it by running the `mod-user` command with the root account. You will be prompted to reset the password the first time you use the account.

You can view information about these accounts by executing the following command:

```
/opt/pbis/bin/enum-users
```

Example output:

```
User info (Level-2):
====================
Name:                        EXAMPLE-01\Administrator
UPN:                         Administrator@EXAMPLE-01
Generated UPN:               YES
```

```
Uid:                        1500
Gid:                        1544
Gecos:                      <null>Shell: /bin/sh
Home dir:                   /
LMHash length:              0
NTHash length:              0
Local User:                 YES
Account disabled:           TRUE
Account Expired:            FALSE
Account Locked:             FALSE
Password never expires:     FALSE
Password Expired:           TRUE
Prompt for password change: YES
User can change password:   NO
Days till password expires: -149314


User info (Level-2):
====================
Name:                       EXAMPLE-01\Guest
UPN:                        Guest@EXAMPLE-01
Generated UPN:              YES
Uid:                        1501
Gid:                        1546
Gecos:                      <null>Shell: /bin/sh
Home dir:                   /tmp
LMHash length:              0
NTHash length:              0
Local User:                 YES
Account disabled:           TRUE
Account Expired:            FALSE
Account Locked:             TRUE
Password never expires:     FALSE
Password Expired:           FALSE
Prompt for password change: YES
User can change password:   NO
Days till password expires: -149314
```

Join Active Directory from the Command Line

On Linux, Unix, and Mac OS X computers, the location of the domain join command-line utility is as follows:

/opt/pbis/bin/domainjoin-cli

Important: To run the command-line utility, you must use a **root** account. To join a computer to a domain, you must have the user name and password of an Active Directory account that has privileges to join computers to the domain and the full name of the domain that you want to join. Instructions on how to delegate rights to join a computer to a domain are at http://support.microsoft.com/kb/932455. After you join a domain for the first time, you must restart the computer before you can log on with your domain account.

When you join a domain by using the command-line utility, PBIS uses the hostname of the computer to derive a fully qualified domain name (FQDN) and then automatically sets the FQDN in the `/etc/hosts` file. You can also join a domain without changing the `/etc/hosts` file; see Join Active Directory Without Changing /etc/hosts.

Before Joining a Domain

To join a domain, the computer's name server must be able to find the domain and the computer must be able to reach the domain controller. You can make sure the name server can find the domain by running this command:

`nslookup` *`domainName`*

You can verify that your computer can reach the domain controller by pinging it:

`ping` *`domainName`*

If either of these tests fails, see Check System Health Before Installing the Agent and Troubleshooting Domain-Join Problems.

Join a Linux or Unix Computer to Active Directory

Execute the following command as root, replacing *`domainName`* with the FQDN of the domain that you want to join and *`joinAccount`* with the user name of an account that has privileges to join computers to the domain:

```
/opt/pbis/bin/domainjoin-cli join domainName joinAccount
```

Example: `/opt/pbis/bin/domainjoin-cli join` *`example.com Administrator`*

Tip: On Ubuntu, execute the `sudo su` command before you run the `domainjoin-cli` command.

Join a Mac Computer to Active Directory

Using `sudo`, execute the following command in Terminal, replacing *`domainName`* with the FQDN of the domain that you want to join and *`joinAccount`* with the user name of an account that has privileges to join computers to the domain:

```
sudo /opt/pbis/bin/domainjoin-cli join domainName
joinAccount
```

Example: `sudo /opt/pbis/bin/domainjoin-cli join` *`example.com Administrator`*

The terminal prompts you for two passwords: The first is for a user account on the Mac that has administrative privileges; the second is for the account in Active Directory that you specified in the join command.

Join a Linux or Unix Computer to an Organizational Unit

Execute the following command as root, replacing *`organizationalUnitName`* with the path and name of the organizational unit that you want to join, *`domainName`* with the FQDN of the domain, and *`joinAccount`* with the user name of an account that has privileges to join computers to the domain:

```
/opt/pbis/bin/domainjoin-cli join --ou
organizationalUnitName domainName joinAccount
```

Example: `/opt/pbis/bin/domainjoin-cli join --ou Engineering example.com Administrator`

Join a Linux or Unix Computer to a Nested Organizational Unit

Execute the following command as root, replacing `path` with the AD path to the OU from the top down, with each node separated by a forward slash (/). In addition, replace *`organizationalUnitName`* with the name of the organizational unit that you want to join. Replace *`domainName`* with the FQDN of the domain and *`joinAccount`* with the user name of an AD account that has privileges to join computers to the target OU:

```
/opt/pbis/bin/domainjoin-cli join --ou
path/organizationalUnitName domainName joinAccount
```

Here is an example of how to join a deeply nested OU:

```
domainjoin-cli join --ou
topLevelOU/middleLevelOU/LowerLevelOU/TargetOU example.com
Administrator
```

`Domainjoin-cli` Options, Commands, and Arguments

The `domainjoin-cli` command-line interface includes the following options:

Option	Description	Example
`--help`	Displays the command-line options and commands.	`domainjoin-cli --help`
`--help-internal`	Displays a list of the internal debugging and configuration commands.	`domainjoin-cli --help-internal`
`--logfile {.| path}`	Generates a log file or prints the log to the console.	`domainjoin-cli --logfile /var/log/domainjoin.log join example.com Administrator` `domainjoin-cli --logfile . join example.com Administrator`

Basic Commands

The domain join command-line interface includes the following basic commands:

Command	Description	Example
`query`	Displays the hostname, current domain, and distinguished name, which includes the OU to which the computer belongs. If the computer is not joined to a domain, it displays only the hostname.	`domainjoin-cli query`
`setname computerName`	Renames the computer and modifies the `/etc/hosts` file with the name that you specify.	`domainjoin-cli setname RHEL44ID`
`fixfqdn`	Fixes a computer's fully qualified domain name.	`domainjoin-cli fixfqdn`
`join [--ou organizationalUnit ]domainName userName`	Joins the computer to the domain that you specify by using the account that you specify.	`domainjoin-cli join --ou Engineering example.com Administrator`

Command	Description	Example
	You can use the `--ou` option to join the computer to an OU within the domain by specifying the path to the OU and the OU's name. When you use this option, you must use an account that has membership in the Domain Administrators security group. The path to the OU is top down.	
`join --notimesync`	Joins the computer to the domain without synchronizing the computer's time with the domain controller's. When you use this option, the `sync-system-time` value for lsass is set to `no`.	`domainjoin-cli join -- notimesync example.com Administrator`
`leave [userName]`	Removes the computer from the Active Directory domain. If the `userName` is provided, the computer account is disabled in Active Directory.	`domainjoin-cli leave` `domainjoin-cli leave smithy@example.com`

Advanced Commands

The command-line interface includes advanced commands that you can use to preview the stages of joining or leaving a domain, find out which configurations are required for your system, view information about a module that will be changed, configure a module such as nsswitch, and enable or disable a module. The advanced commands provide a potent tool for troubleshooting issues while configuring a Linux or Unix computer to interoperate with Active Directory.

View a data-flow diagram that shows how systems interact when you join a domain.

Preview the Stages of the Domain Join for Your Computer

To preview the domain, DNS name, and configuration stages that will be used to join a computer to a domain, execute the following command at the command line:

```
domainjoin-cli join --preview domainName
```

Example: `domainjoin-cli join --preview example.com`

Here is an example of the results, which can vary by computer:

```
[root@rhel4d bin]# domainjoin-cli join --preview
example.com
Joining to AD Domain:   example.com
With Computer DNS Name: rhel4d.example.com

The following stages are currently configured to be run
during the domain join:
join           - join computer to AD
krb5           - configure krb5.conf
nsswitch       - enable/disable PowerBroker Identity
Services nsswitch module
start          - start daemons
pam            - configure pam.d/pam.conf
ssh            - configure ssh and sshd
```

Check Required Configurations

To see a full listing of the modules that apply to your operating system, including those modules that will not be run, execute either the following join or leave command:

`domainjoin-cli join --advanced --preview` domainName

`domainjoin-cli leave --advanced --preview` domainName

Example: `domainjoin-cli join --advanced --preview example.com`

The result varies by computer:

```
[root@rhel4d bin]# domainjoin-cli join --advanced --
preview example.com
Joining to AD Domain:   example.com
With Computer DNS Name: rhel4d.example.com
    [F] stop           - stop daemons
    [F] hostname       - set computer hostname
    [F] firewall       - open ports to DC
    [F] keytab         - initialize kerberos keytab
[X] [N] join           - join computer to AD
[X] [N] krb5           - configure krb5.conf
[X] [N] nsswitch       - enable/disable PowerBroker
Identity Services nsswitch module
[X] [N] start          - start daemons
    [F] gdm            - fix gdm presession script for
spaces in usernames
[X] [N] pam            - configure pam.d/pam.conf
[X] [S] ssh            - configure ssh and sshd

Key to flags
[F]ully configured        - the system is already
configured for this step
[S]ufficiently configured - the system meets the minimum
```

```
configuration
                                  requirements for this step
[N]ecessary                     - this step must be run or
manually performed.
[X]                             - this step is enabled and will
make changes
[ ]                             - this step is disabled and
will not make changes
```

View Details about a Module

The PBIS domain join tool includes the following modules—the components and services that the tool must configure before it can join a computer to a domain:

Module	Description
join	Joins the computer to Active Directory
leave	Deletes the machine account in Active Directory
dsplugin	Enables the PBIS directory services plugin on a Mac computer
stop	Stops services so that the system can be configured
start	Starts services after configuration
firewall	Opens ports to the domain controller
hostname	sets the computer hostname
krb5	Configures `krb5.conf`
pam-mode	Switches authentication from LAM to PAM
nsswitch	Enables or disables PBIS nsswitch module
pam	Configures pam.d and pam.conf
lam-auth	Configures LAM for Active Directory authentication
ssh	Configures ssh and sshd
bash	Fixes the bash prompt for backslashes in usernames
gdm	Fixes gdm presession script for spaces in usernames

As the previous section illustrated, you can see the modules that must be configured on your computer by executing the following command:

```
domainjoin-cli join --advanced --preview domainName
```

You can further bore down into the details of the changes that a module will make by using either the following join or leave command:

```
domainjoin-cli join --details module domainName joinAccount
```

```
domainjoin-cli leave --details module domainName joinAccount
```

Example: `domainjoin-cli join --details nsswitch example.com Administrator`

The result varies depending on your system's configuration:

```
domainjoin-cli join --details nsswitch example.com
Administrator
[X] [N] nsswitch          - enable/disable PowerBroker
Identity Services nsswitch module

Key to flags
[F]ully configured        - the system is already
configured for this step
[S]ufficiently configured - the system meets the minimum
configuration
                            requirements for this step
[N]ecessary               - this step must be run or
manually performed.
[X]                       - this step is enabled and will
make changes
[ ]                       - this step is disabled and
will not make changes

Details for 'enable/disable PowerBroker Identity Services
nsswitch module':
The following steps are required and can be performed
automatically:
    * Edit nsswitch apparmor profile to allow libraries
in the /opt/pbis/lib
        and /opt/pbis/lib64 directories
    * List lwidentity module in
/usr/lib/security/methods.cfg (AIX only)
    * Add lwidentity to passwd and group/groups line
/etc/nsswitch.conf or
        /etc/netsvc.conf

If any changes are performed, then the following services
must be restarted:
    * GDM
    * XDM
    * Cron
    * Dbus
    * Nscd
```

Turn On or Turn Off Domain-Join Modules

You can explicitly enable or disable a module when you join or leave a domain. Disabling a module can be useful in cases where a module has been manually configured or in cases where you must ensure that certain system files will not be modified.

Note: If you disable a necessary module and you have not manually configured it, the domain join utility will not join your computer to the domain.

The following command, with either `join` or `leave`, can be used to disable a module:

```
domainjoin-cli join --disable module domainName
accountName
domainjoin-cli leave --disable module domainName
accountName
```

Example: `domainjoin-cli join --disable pam example.com Administrator`

To enable a module, execute the following command at the command line:

```
domainjoin-cli join --enable module domainName
accountName
```

Example: `domainjoin-cli join --enable pam example.com Administrator`

Configuration and Debugging Commands

The `domainjoin-cli` tool includes commands for debugging the domain-join process and for configuring or preconfiguring a module. You can, for example, run the `configure` command to preconfigure a system before you join a domain—a useful strategy when you are deploying PBIS in a virtual environment and you need to preconfigure the nsswitch, ssh, or PAM module of the target computers to avoid having to restart them after they are added to the domain. Here is an example with nsswitch:

domainjoin-cli configure --enable nsswitch

The following commands, viewable by running `domainjoin-cli --help-internal`, are available:

```
    fixfqdn
    configure { --enable | --disable } pam [--testprefix
<dir>]
    configure { --enable | --disable } nsswitch [--
testprefix <dir>]
    configure { --enable | --disable } ssh [--testprefix
<dir>]
    configure { --enable | --disable } [--testprefix
```

```
<dir>]
                [--long <longdomain>] [--short
<shortdomain>] krb5
    configure { --enable | --disable } firewall [--
testprefix <dir>]
    configure { --enable | --disable } eventfwd
    configure { --enable | --disable } reapsysl
    get_os_type
    get_arch
    get_distro
    get_distro_version
    raise_error <error code | error name | 0xhex error
code>
```

Join Active Directory Without Changing /etc/hosts

When you join a computer to a domain by using the PBIS domain join tool, PBIS uses the hostname of the computer to derive a fully qualified domain name (FQDN) and automatically sets the computer's FQDN in the `/etc/hosts` file.

To join a Linux computer to the domain without changing the `/etc/hosts` file, execute the following command as **root**, replacing `domainName` with the FQDN of the domain that you want to join and `joinAccount` with the user name of an account that has privileges to join computers to the domain:

`/opt/pbis/bin/domainjoin-cli join --disable hostname domainName joinAccount`

Example: `/opt/pbis/bin/domainjoin-cli join --disable hostname example.com Administrator`

After you join a domain for the first time, you must restart the computer before you can log on.

If the Computer Fails to Join the Domain

Make sure the computer's FQDN is correct in `/etc/hosts`. For the computer to process tickets in compliance with the Kerberos protocol and to function properly when it uses cached credentials in offline mode or when its DNS server is offline, there must be a correct FQDN in `/etc/hosts`. For more information on GSS-API requirements, see RFC 2743.

You can determine the fully qualified domain name of a computer running Linux, Unix, or Mac OS X by executing the following command:

```
ping -c 1 `hostname`
```

When you execute this command, the computer looks up the primary host entry for its hostname. In most cases, this means that it looks for its hostname in `/etc/hosts`, returning the first FQDN name on the same line. So, for the hostname `qaserver`, here is an example of a correct entry in `/etc/hosts`:

```
10.100.10.10 qaserver.corpqa.example.com qaserver
```

If, however, the entry in `/etc/hosts` incorrectly lists the hostname (or anything else) before the FQDN, the computer's FQDN becomes, using the malformed example below, `qaserver`:

```
10.100.10.10 qaserver qaserver.corpqa.example.com
```

If the host entry cannot be found in `/etc/hosts`, the computer looks for the results in DNS instead. This means that the computer must have a correct A record in DNS. If the DNS information is wrong and you cannot correct it, add an entry to `/etc/hosts`.

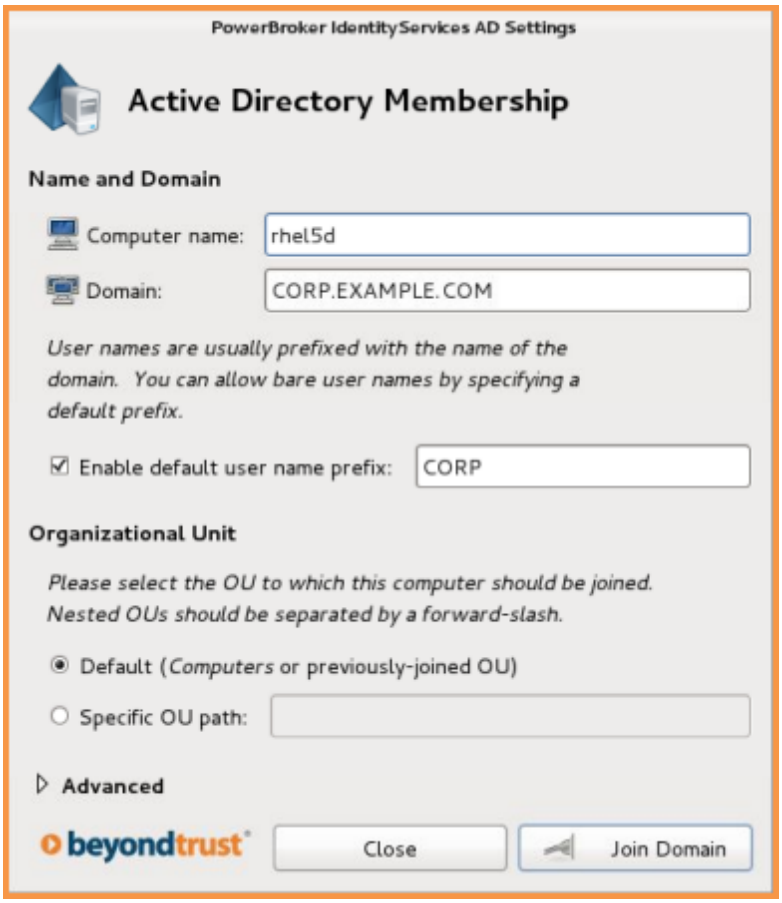
Join a Linux Computer to Active Directory

A graphical user interface for joining a domain is included when you install the PBIS agent.

Important: To join a computer to a domain, you must have the user name and password of a user who has privileges to join computers to a domain and the full name of the domain that you want to join.

1. With **root** privileges, run the following command at the shell prompt of a Linux computer:
   `/opt/pbis/bin/domainjoin-gui`

2. Continuing as root, in the **Domain** box, enter the Fully Qualified Domain Name (FQDN) of your Active Directory domain. Example: `CORP.EXAMPLE.COM`



Note: The domain join tool automatically sets the computer's FQDN by modifying the `/etc/hosts` file. For example, if your computer's name is `qaserver` and the domain is `corpqa.example.com`, the domain join tool adds the following entry to the `/etc/hosts` file: `qaserver.corpqa.example.com`. To manually set the computer's FQDN, see Join Active Directory Without Changing /etc/hosts.

3. To avoid typing the domain prefix before your user or group name each time you log on—that is, to force the computer to assume the default domain—select **Enable default user name prefix** and enter your domain prefix in the box. Example: `CORP`
4. Under **Organizational Unit**, you can optionally join the computer to an OU by selecting **Specific OU path** and then typing a path in the box. The OU path is from the top of the Active Directory domain down to the OU that you want.
   Or, to join the computer to the Computers container, select **Default**.
5. Click **Join Domain**.

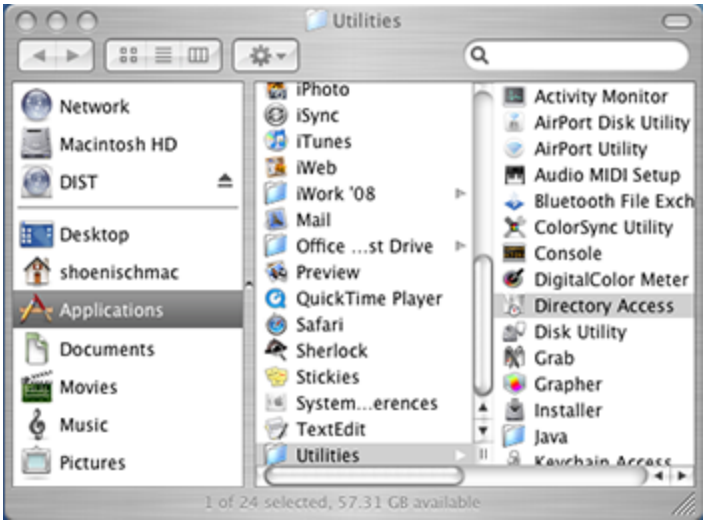
6. Enter the user name and password of an Active Directory account that has privileges to join computers to the domain and then click **OK**.
   Note: If you do not use an Active Directory Domain Administrator account, you might not have sufficient privileges to change a machine object in Active Directory.

After you join a domain for the first time, you must restart the computer before you can log on.

Join a Mac Computer to Active Directory

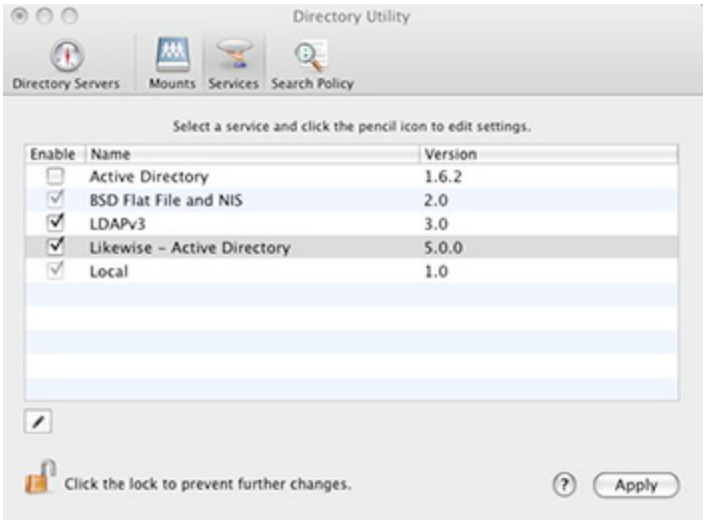
To join a computer running Mac OS X 10.4 or later to an Active Directory domain, you must have administrative privileges on the Mac and privileges on the Active Directory domain that allow you to join a computer.

1. In Finder, click **Applications**. In the list of applications, double-click **Utilities**, and then double-click **Directory Access** in OS X 10.4 or **Directory Utility** in OS X 10.5. In Mac OS X 10.6 (Snow Leopard), you gain access to Directory Utility by using the **Apple** menu to view the system preferences for accounts; for instructions, see your Mac OS X 10.6 documentation.



2. On Mac OS X 10.5, click **Show Advanced Settings**.
3. On the **Services** tab, click the lock and enter an administrator name and password to unlock it.
4. In the list, make sure that the check box for **Active Directory** is not selected.
   Important: Active Directory, Apple's built-in service for interoperating with AD, must be disabled for PBIS to work properly.

5. In the list, click **Likewise - Active Directory**, make sure the **Enable** check box for **Likewise - Active Directory** is selected, and then click **Configure** in OS X 10.4 or double-click **Likewise – Active Directory** in OS X 10.5 and later.
   Note: On Mac OS X 10.6, if **Likewise - Active Directory** does not appear in the list, restart your computer.



6. Enter a name and password of a local machine account with administrative privileges.
7. On the menu bar at the top of the screen, click the **Domain Join** menu, and then click **Join or Leave Domain**.
8. In the **Computer name** box, type the local hostname of the Mac without the `.local` extension. Because of a limitation with Active Directory, the local hostname cannot be more than 15 characters. Also: `localhost` is not a valid name.

> **Tip**
>
> To find the local hostname of a Mac, on the **Apple** menu , click **System Preferences**, and then click **Sharing**. Under the **Computer Name** box, click **Edit**. Your Mac's local hostname is displayed.

9. In the **Domain to join** box, type the fully qualified domain name of the Active Directory domain that you want to join.

10. Under **Organizational Unit**, you can join the computer to an OU in the domain by selecting **OU Path** and then typing a path in the **OU Path** box.
    Note: To join the computer to an OU, you must be a member of the Domain Administrator security group.

    Or, to join the computer to the Computers container, select **Default to "Computers" container**.

11. Click **Join**.
12. After you are joined to the domain, you can set the display login window preference on the Mac: On the **Apple** menu , click **System Preferences**, and then under **System**, click **Accounts**.
13. Click the lock and enter an administrator's name and password to unlock it.
14. Click **Login Options**, and then under **Display login window as**, select **Name and password.**

With PBIS Enterprise, the domain join utility includes a tool to migrate a Mac user's profile from a local user account to the home directory specified for the user in Active Directory; see Migrate a User Profile on a Mac.

Turn Off OS X Directory Service Authentication

If you are migrating from Open Directory or Active Directory and you had set authentication from the command line with `dsconfigad` or `dsconfigldap`, you must run the following commands to stop the computer from trying to use the built-in directory service even if the Mac is not bound to it:

```
dscl . -delete /Computers
dscl /Search -delete / CSPSearchPath
/LDAPv3/FQDNforYourDomainController
dscl /Search -delete / CSPSearchPath /Active\
Directory/All\ Domains
dscl /Search/Contacts -delete / CSPSearchPath /Active\
Directory/All\ Domains
dscl /Search/Contacts -delete / CSPSearchPath
/LDAPv3/FQDNforYourDomainController
```

Use PBIS with a Single Organizational Unit

If you have write privileges only for an organizational unit (OU) in Active Directory (AD), you can still use PBIS. Your AD rights to create objects in an OU allow you to join Linux and Unix computers to the OU even though you do not have Active Directory Domain Administrator or Enterprise Administrator privileges.

There are additional limitations to this approach:

- You must join the computer to a specific OU, and you must know the path to that OU.
- You cannot use PBIS Enterprise in schema mode unless you have Enterprise Administrator privileges, which are required to upgrade the schema.

Join a Linux Computer to an Organizational Unit

To join a computer to a domain, you must have the user name and password of an account that has privileges to join computers to the OU and the full name of the domain that you want to join. The OU path is from the top OU down to the OU that you want.

As root, execute the following command, replacing `organizationalUnitName` with the path and name of the organizational unit that you want to join, `domainName` with the FQDN of the domain, and `joinAccount` with the user name of an account that has privileges to join computers to the domain:

```
/opt/pbis/bin/domainjoin-cli join --ou
organizationalUnitName domainName joinAccount
```

Example: `/opt/pbis/bin/domainjoin-cli join --ou Engineering example.com Administrator`

Example of how to join a nested OU:

`domainjoin-cli join --ou topLevelOU/middleLevelOU/LowerLevelOU/TargetOU example.com Administrator`

After you join a domain for the first time, you must restart the computer before you can log on.

Rename a Joined Computer

To rename a computer that has been joined to Active Directory, you must first leave the domain. You can then rename the computer by using the domain join command-line interface. After you rename the computer, you must rejoin it to the domain. Renaming a joined computer requires the user name and password of a user with privileges to join a computer to a domain.

Important: Do not change the name of a Linux, Unix, or Mac computer by using the `hostname` command because some distributions do not permanently apply the changes.

Rename a Computer by Using the Command-Line Tool

The following procedure removes a Unix or Linux computer from the domain, renames the computer, and then rejoins it to the domain.

1. With root privileges, at the shell prompt of a Unix computer, execute the following command:
   `/opt/pbis/bin/domainjoin-cli leave`

2. To rename the computer in `/etc/hosts`, execute the following command, replacing `computerName` with the new name of the computer:
   `/opt/pbis/bin/domainjoin-cli setname computerName`

   Example: `/opt/pbis/bin/domainjoin-cli setname RHEL44ID`

3. To rejoin the renamed computer to the domain, execute the following command at the shell prompt, replacing DomainName with the name of the domain that you want to join and UserName with the user name of a user who has privileges to join a domain:
   `/opt/pbis/bin/domainjoin-cli join DomainName UserName`

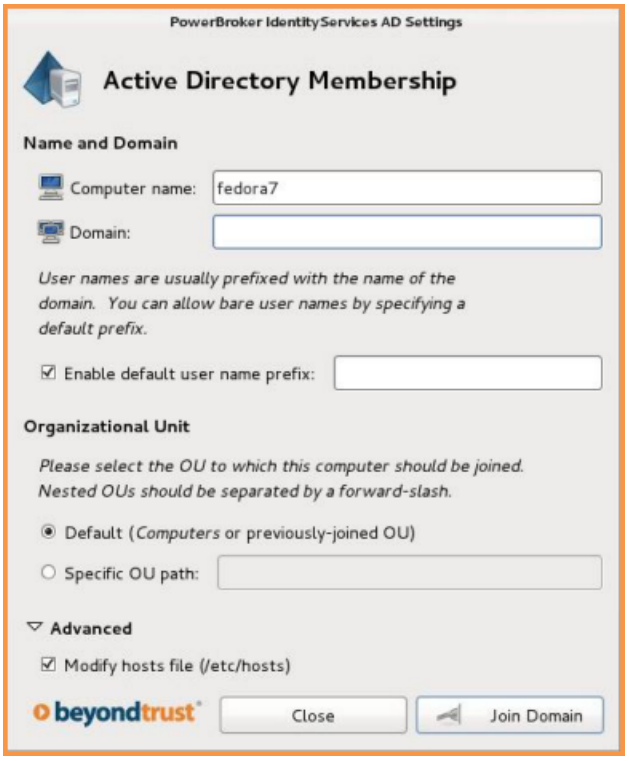
   Example: `/opt/pbis/bin/domainjoin-cli join example.com Administrator`

   It may take a few moments before the computer is joined to the domain.

4. After you change the hostname of a computer, you must also change the name in the PBIS local provider database so that the local PBIS accounts use the correct prefix. To do so, execute the following command as root, replacing `hostName` with the name that you want:
   `/opt/pbis/bin/set-machine-name hostName`

Rename a Computer by Using the Domain Join Tool GUI

1. From the desktop with root privileges, double-click the PBIS Domain Join Tool, or at the shell prompt of a Linux computer, type the following command:
   `/opt/pbis/bin/domainjoin-gui`
2. Click **Leave**, and then click **OK**.
3. Start the domain join tool again by double-clicking the PBIS Domain Join Tool on the desktop, or by typing the following command at the shell prompt of a Linux computer:
   `/opt/pbis/bin/domainjoin-gui`
4. Click **Next**.
5. In the **Computer name** box, rename the computer by typing a new name.



6. In the **Domain** box, enter the Fully Qualified Domain Name (FQDN) of the Active Directory domain.
7. Under **Organizational Unit**, you can join the computer to an OU in the domain by selecting **OU Path** and then typing a path in the **Specific OU path** box.
   Or, to join the computer to the Computers container, select **Default**.
8. Click **Next**.

9. Enter the user name and password of an Active Directory user with authority to join a machine to the Active Directory domain, and then click **OK**.
   The computer's name in `/etc/hosts` has been changed to the name that you specified and the computer has been joined to the Active Directory domain with the new name.

10. After you change the hostname of a computer, you must also change the name in the PBIS local provider database so that the local PBIS accounts use the correct prefix. To do so, execute the following command as root, replacing `hostName` with the name that you want:
    `/opt/pbis/bin/set-machine-name hostName`

Files Modified When You Join a Domain

When PBIS adds a computer to a domain, it modifies some system files. The files that are modified depend on the platform, the distribution, and the system's configuration. The following files might be modified.

To see a listing of the changes that joining a domain will make to your operating system, execute the following join command:

domainjoin-cli join --advanced --preview domainName

Note: Not all the following files are present on all computers.

- /etc/nsswitch.conf (On AIX, the file is /etc/netsvcs.conf.)
- /etc/pam.conf on AIX, HP-UX, and Solaris
- /etc/pam.d/* on Linux
- /etc/ssh/{ssh_config,sshd_config} (or wherever sshd configuration is located)
- /etc/hosts (To join a domain without modifying /etc/hosts, see Join Active Directory Without Changing /etc/hosts.)
- /etc/apparmor.d/abstractions/nameservice
- /etc/X11/gdm/PreSession/Default
- /etc/vmware/firewall/services.xml
- /usr/lib/security/methods.cfg
- /etc/security/user
- /etc/security/login.cfg
- /etc/netsvc.conf
- /etc/krb5.conf
- /etc/krb5/krb5.conf
- /etc/rc.config.d/netconf
- /etc/nodename

- /etc/{hostname,HOSTNAME,hostname.*}
- /etc/sysconfig/network/config
- /etc/sysconfig/network/dhcp
- /etc/sysconfig/network/ifcfg-*
- /etc/sysconfig/network-scripts/ifcfg-*
- /etc/init.d or /sbin/init.d
- /etc/rcX.d/ (new files and links created)
- /etc/inet/ipnodes

As an example, the following table lists the files that are modified for the *default configuration* of the operating system of a few selected platforms.

Mofified Files	Solaris 9	Solaris 10	AIX 5.3	AIX 6.1	Red Hat Enterprises Linux 5
/etc/nsswitch.conf (On AIX, the file is /etc/netsvcs.conf.)	Modified	Modified			Modified
/etc/pam.conf on AIX, HP-UX, and Solaris	Modified	Modified	Modified	Modified	
/etc/pam.d/* on Linux					Modified
/etc/ssh/{ssh_config,sshd_config} (or wherever sshd configuration is located)		Modified	Modified		Modified
/etc/hosts	Modified	Modified	Modified	Modified	Modified
/etc/apparmor.d/abstractions/nameservice					
/etc/X11/gdm/PreSession/Default					
/etc/vmware/firewall/services.xml					
/usr/lib/security/methods.cfg			Modified	Modified	
/etc/security/user			Modified	Modified	
/etc/security/login.cfg			Modified		
/etc/netsvc.conf			Modified	Modified	
/etc/krb5.conf			Modified	Modified	Modified
/etc/krb5/krb5.conf	Modified	Modified			
/etc/rc.config.d/netconf					
/etc/nodename	Modified	Modified			
/etc/{hostname, HOSTNAME, hostname.*}	Modified				
/etc/sysconfig/network/config					
/etc/sysconfig/network/dhcp					
/etc/sysconfig/network/ifcfg-*					
/etc/sysconfig/network-scripts/ifcfg-*					
/etc/init.d or /sbin/init.d					
/etc/rcX.d/ (new files and links created)				Modified	
/etc/inet/ipnodes	Modified	Modified			

NetworkManager: Use a Wired Connection to Join a Domain

On Linux computers running NetworkManager—which is often used for wireless connections—you must make sure before you join a domain that the computer has a non-wireless network connection and that the non-wireless connection is configured to start when the networking cable is plugged in. You must continue to use the non-wireless network connection during the post-join process of restarting your computer and logging on with your Active Directory domain credentials.

After you have joined the domain and logged on for the first time with your AD domain credentials by using a non-wireless connection, you can then revert to using your wireless connection because your AD logon credentials are cached. (You will not, however, be notified when your AD password is set to expire until you either run a sudo command or log on by using a non-wireless connection.)

If, instead, you attempt to use a wireless connection when you join the domain, you will be unable to log on your computer with AD domain credentials after your computer restarts.

Here is why: NetworkManager is composed of a daemon that runs at startup and a user-mode application that runs only after you log on. NetworkManager is typically configured to auto-start wired network connections when they are plugged in and wireless connections when they are detected. The problem is that the wireless network is not detected until the user-mode application starts—which occurs only after you have logged on.

Information about NetworkManager is available at http://projects.gnome.org/NetworkManager/.

Logging on with Domain Credentials

PBIS includes the following logon options:

- Full domain credentials—example: `example.com\\hoenstiv`
- Single domain user name—example: `example\\hoenstiv`
- Alias—example: `stiv`
- Cached credentials

Important: When you log on from the command line, you must use a slash to escape the slash character, making the logon form `DOMAIN\\username`.

To use UPN names, you must raise your Active Directory forest functional level to Windows Server 2003, but raising the forest functional level to Windows Server 2003 will exclude Windows 2000 domain controllers from the domain.

When you log on a Linux, Unix, or Mac OS X computer by using your domain credentials, PBIS uses the Kerberos protocol to connect to Active Directory's key distribution center, or KDC, to establish a key and to request a Kerberos ticket granting ticket (TGT). The TGT lets you log on to other computers joined to Active Directory or applications provisioned with a service principal name and be automatically authenticated with Kerberos and authorized for access through Active Directory.

After logon, PBIS stores the password in memory and securely backs it up on disk. You can, however, configure PBIS to store logon information in a SQLite database, but it is not the default method. The password is used to refresh the user's Kerberos TGT and to provide NTLM-based single sign-on through the PBIS GSSAPI library. In addition, the NTLM verifier hash—a hash of the NTLM hash—is stored to disk to handle offline logons by comparing the password with the cached credentials.

PBIS stores an NTLM hash and LM hash only for accounts in PBIS's local provider. The hashes are used to authenticate users over CIFS. Since PBIS does not support offline logons for domain users over CIFS, it does not store the LM hash for domain users.

See Also

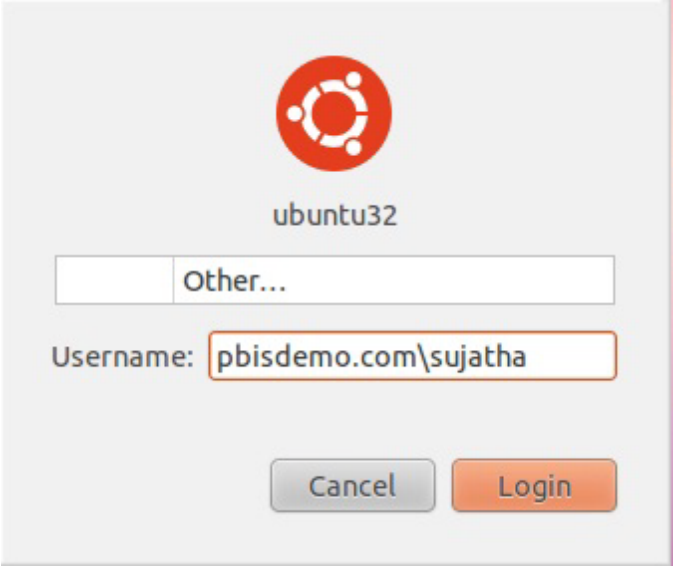
Using PBIS for Single Sign-On

Configure PuTTY for Windows-Based SSO

Log on with AD Credentials

After the PBIS agent has been installed and the Linux or Unix computer has been joined to a domain, you can log on with your Active Directory credentials, either from the command line or interactively through the system console. After you join a domain for the first time, you must reboot your computer before you can log on interactively through the console.

- Log on from the command line, but make sure you use a slash character to escape the slash, making the logon form `DOMAIN\\username`. Example with ssh: `ssh example.com\\hoenstiv@localhost`
- Log on the system console or the text login prompt by using an Active Directory user account in the form of `DOMAIN\username`, where `DOMAIN` is the Active Directory short name. Example on Ubuntu:



Log on with SSH

You can log on with SSH by executing the `ssh` command at the shell prompt in the following format:

```
ssh DOMAIN\\username@localhost
```

Example: `ssh example.com\\hoenstiv@localhost`

Solve Logon Problems from Windows

To troubleshoot a problem with a user who cannot log on a to Linux or Unix computer, perform the following series of diagnostic tests sequentially.

1. On a Windows computer, log off and then log on again with the problem user's AD credentials to verify that the password is correct and that the account is not locked or disabled.
2. Try to SSH to the target Linux or Unix computer again with the user's full NT4-style credentials and password, not just the user's alias. In your SSH command, make sure to use a slash character to escape the slash.
3. If you are using PBIS Enterprise, make sure that the user's computer is in the correct PowerBroker cell.
4. Make sure that the user is enabled to log on the computer, either by being enabled in the cell (with PBIS Enterprise) or by being in a group allowed to access the computer. Then try to log on the target computer again.
5. Ensure that the PBIS client can communicate with the Active Directory domain controller.
6. Make sure that the shell specified for the user account in Active Directory is available on the target computer. Specifying a shell that is unavailable will block the user account from logging on.
7. Verify that the home directory is set and can be created. A home directory that cannot be created because the path is incorrect or the permissions are insufficient can block an attempt to log on.
8. Make sure there are no logon restrictions in place—for example, the Group Policy setting that restricts logon to certain users or groups—that prevent the user account from logging on the computer.
9. Log on the computer with a different user account—one that is enabled for access to the computer.

Solve Logon Problems on Linux or Unix

To troubleshoot problems logging on a Linux computer with Active Directory credentials after you joined the computer to a domain, perform the following series of diagnostic tests sequentially with a root account. The tests can also be used to troubleshoot logon problems on a Unix or Mac OS X computer; however, the syntax of the commands on Unix and Mac might be slightly different.

Make Sure You Are Joined to the Domain

Execute the following command:

```
/opt/pbis/bin/domainjoin-cli query
```

If you are not joined, see Join Active Directory with the Command Line.

Check Whether You Are Using a Valid Logon Form

When troubleshooting a logon problem, use your full domain credentials: `DOMAIN\username`. Example: `example.com\hoenstiv`.

When logging on from the command line, you must escape the slash character with a slash character, making the logon form `DOMAIN\\username`. Example: `example.com\\hoenstiv`.

To view a list of logon options, see Logging On with Domain Credentials.

Clear the Cache

You may need to clear the cache to ensure that the client computer recognizes the user's ID. See Clear the Authentication Cache.

Destroy the Kerberos Cache

Clear the PBIS Kerberos cache to make sure there is not an issue with a user's Kerberos tickets. Execute the following command with the user account that you are troubleshooting:

```
/opt/pbis/bin/kdestroy
```

Check the Status of the PBIS Authentication Service

Check the status of the authentication service on a Unix or Linux computer running the PBIS Agent by executing the following command as the root user:

```
/opt/pbis/bin/lwsm status lsass
```

If	Do This
The result looks like this: `lsass is stopped`	Restart the service.
The result looks like this: `lsass (pid 1783) is running...`	Proceed to the next test.

Check Communication between the PBIS Service and AD

Verify that the PBIS service can exchange data with AD by executing this command:

```
/opt/pbis/bin/get-dc-name FullDomainName
```

Example: `/opt/pbis/bin/get-dc-name example.com`

If	Do This
The result does not show the name and IP address of your domain controller	1. Make sure the domain controller is online and operational. 2. Check network connectivity between the client and the domain controller. 3. Join the domain again. 4. View log files.
The result shows the correct domain controller name and IP address	Proceed to the next test.

Verify that PBIS Can Find a User in AD

Verify that the PBIS agent can find your user by executing the following command, substituting the name of a valid AD domain for *domainName* and a valid user for *ADuserName*:

`/opt/pbis/bin/find-user-by-name` *domainName*`\\`*ADuserName*

Example: `/opt/pbis/bin/find-user-by-name example\\hab`

If	Do This
The command fails to find the user	1. Check whether the computer is joined to the domain by executing the following command as root: `domainjoin-cli query` Displays the hostname, current domain, and distinguished name, which includes the OU to which the computer belongs. Make sure the OU is correct. If the computer is not joined to a domain, it displays only the hostname. 2. Check Active Directory to make sure the user has an account. 3. Check whether the same user is in the `/etc/passwd` file. 4. Make sure the AD authentication provider is running by proceeding to the next test.
The user is found	Proceed to the PAM test later in this topic.

Make Sure the AD Authentication Provider Is Running

PBIS includes two authentication providers:

1. The local provider
2. The Active Directory provider

If the AD provider is not online, users are unable to log on with their AD credentials. To check the status of the authentication providers, execute the following command as root:

```
/opt/pbis/bin/get-status
```

A healthy result should look like this:

```
LSA Server Status:
Agent version: 6.5.0
Uptime:        2 days 21 hours 16 minutes 29 seconds
[Authentication provider: lsa-local-provider]
        Status:   Online
        Mode:     Local system
[Authentication provider: lsa-activedirectory-provider]
        Status:   Online
        Mode:     Un-provisioned
        Domain:   example.com
        Forest:   example.com
        Site:     Default-First-Site-Name
```

An unhealthy result will not include the AD authentication provider or will indicate that it is offline. If the AD authentication provider is not listed in the results, restart the authentication service.

If the result looks like the line below, check the status of the PBIS services to make sure they are running.

```
Failed to query status from LSA service.
The LSASS server is not responding.
```

Run the `id` Command to Check the User

Run the following `id` command to check whether nsswitch is properly configured to handle AD user account information:

```
id DOMAIN\\username
```

Example: `id example\\kathy`

If the command does not show information for the user, check whether the `/etc/nsswitch.conf` file is properly configured for passwd and group: Both entries should include the `lsass` parameter.

If `/etc/nsswitch.conf` is properly configured, the PBIS name service libraries might be missing or misplaced. It is also possible that the `LD_PRELOAD` or `LD_LIBRARY_PATH` variables are defined without including the PBIS libraries.

Switch User to Check PAM

Verify that a user's password can be validated through PAM by using the switch user service. Either switch from a non-root user to a domain user or from root to a domain user. If you switch from root to a domain user, run the command below twice so that you are prompted for the domain user's password:

`su DOMAIN\\username`

Example: `su example\\hoenstiv`

If	Do This

The switch user command fails to validate the user	Generate a PAM debug log.
	Also, check the following log files for error messages (the location of the log files varies by operating system):
	`/var/log/messages`
	`/var/log/secure`

Test SSH

Check whether you can log on with SSH by executing the following command:

`ssh DOMAIN\\username@localhost`

Example: `ssh example.com\\hoenstiv@localhost`

If you believe the issue might be specific to SSH, see Troubleshooting SSH SSO Problems.

Run the Authentication Service in Debug Mode

To troubleshoot the lookup of a user or group ID, you can set the PBIS authentication service to run in debug mode and show the log in the console by executing this command:

/opt/pbis/sbin/lsass --loglevel debug

Check Nsswitch.Conf

Make sure /etc/nsswitch.conf is configured correctly to work with PBIS. For more information, see Configuring Clients Before Agent Installation.

On HP-UX, Escape Special Characters at the Console

When you log on to the console on some versions of HP-UX, such as 11.23, you might need to escape special characters, such as `@` and `#`, by preceding them with a slash (\). For more information, see your HP-UX documentation.

Additional Diagnostic Tools

There are additional command-line utilities that you can use to troubleshoot logon problems in the following directory:

`/opt/pbis/bin`

See Also

Resolve an AD Alias Conflict with a Local Account

Troubleshooting SSH SSO Problems

Solve problems logging on with SSH to Linux, Unix, and Mac OS X computers running PBIS.

Before you begin troubleshooting

Make sure you are joined to the domain by executing the following command as root:

/opt/pbis/bin/domainjoin-cli query

If you are not joined, see Join Active Directory with the Command Line.

You can use the following steps to troubleshoot problems logging on to Linux, Unix, and Mac OS X computers with ssh. It is assumed that the computer is connected to Microsoft Active Directory with PBIS Open or PBIS Enterprise and that you are trying to log on with an Active Directory account.

Use NT4-style Credentials and Escape the Slash Character

Try to SSH to the target Linux or Unix computer again with the user's full NT4-style credentials, not the user's alias. In your SSH command, make sure to use a slash character to escape the slash.

Here is an example:

ssh example.com\\kathy@localhost

Perform General Logon Troubleshooting

If you cannot logon after you escaped the slash character in your full NT4-style credentials and used your password, execute the general logon troubleshooting steps in Solve Logon Problems from Windows and Solve Logon Problems on Linux and Unix. If those steps do not help solve the problem, return to this page and perform the following PBIS-specific ssh troubleshooting steps in the order listed.

This document contains little general SSH troubleshooting information. If you believe your issue is not specific to PBIS or if the information here does not solve your problem, see *SSH: The Secure Shell: The Definitive Guide*, published by O'Reilly. See especially the sections on troubleshooting, logging and debugging, and password authentication.

Get an SSH Log

You should obtain debug logs for the PBIS authentication service (lsass), PAM, and sshd. To generate PAM and lsass logs, see the section on Logging.

To get an ssh log, locate `sshd` and then start it in a separate terminal window with the following options:

`which sshd` -vvv -p 9999 >/tmp/sshd.log 2>&1

The command starts an instance of `sshd` listening on Port 9999 and routes logging information to a log file in `/tmp/sshd.log`.

Now try to ssh to the localhost at that port:

ssh -ddd -p 9999 yourADuserName@localhost

When the logon fails, kill ssh; the `sshd` session will stop as well.

Finally, check the log file at `/tmp/sshd.log` for information that might help you resolve the issue. In addition, check the log files for lsass and PAM. For more information on how to generate a log for SSH, see logging and debugging or the man page for ssh.

After an Upgrade, Reconfigure SSH for PBIS

If SSH was recently upgraded, run the following command as root to make sure that the `sshd_config` file is set up properly to work with PBIS:

domainjoin-cli configure --enable ssh

Verify that Port 22 Is Open

A common problem is that a firewall is blocking the port used by SSH. Take a moment to verify that Port 22, which SSH typically connects to, is available by telneting to it. Failure looks like this:

```
root@example:~# telnet 10.0.0.17 22
Trying 10.0.0.18...
telnet: Unable to connect to remote host: Connection
refused
```

Success looks like this:

```
root@example:~# telnet 10.0.0.17 22
Trying 10.0.0.17...
Connected to 10.0.0.17.
Escape character is '^]'.
SSH-2.0-OpenSSH_5.1p1 Debian-5
```

Make Sure PAM Is Enabled for SSH

If your Active Directory account is not working with SSH, make sure that `UsePAM` is enabled in `sshd_config` and make sure that your `sshd` application is linked to the PAM libraries.

1. Determine which sshd is running by executing the following command:

```
bash-3.2# ps -ef | grep sshd
    root  8199     1  0  Feb  6  ?         0:00
/opt/ssh/sbin/sshd
    root  2987  8199  0  Mar  3  ?         0:04 sshd:
root@notty
    root 24864  8199  0 12:16:25 ?         0:00 sshd:
root@pts/0
    root  2998  8199  0  Mar  3  ?         0:05 sshd:
root@notty
    root 24882 24880  0 12:16:54 pts/0     0:00 grep sshd
```

2. Either use `lsof` to find out which configuration file it is reading or start it up with debugging to figure out the default path. Example:

```
username@computer:~$ /usr/sbin/sshd -dd -t
   debug2: load_server_config: filename /etc/ssh/sshd_
config
   debug2: load_server_config: done config len = 664
   debug2: parse_server_config: config /etc/ssh/sshd_
config len 664
   debug1: sshd version OpenSSH_5.1p1 Debian-3ubuntu1
   Could not load host key: /etc/ssh/ssh_host_rsa_key
   Could not load host key: /etc/ssh/ssh_host_dsa_key
```

3. Verify that `UsePAM` is enabled in the config file. As a best practice, make a backup copy of the configuration file before you change it.

4. Run `ldd` on `sshd` to make sure it links with `libpam`. Here is an example from an IA64 HP system:

```
bash-3.2# ldd /opt/ssh/sbin/sshd
        libpam.so.1 =>  /usr/lib/hpux64/libpam.so.1
        libdl.so.1 =>   /usr/lib/hpux64/libdl.so.1
        libnsl.so.1 =>  /usr/lib/hpux64/libnsl.so.1
        libxnet.so.1 => /usr/lib/hpux64/libxnet.so.1
        libsec.so.1 =>  /usr/lib/hpux64/libsec.so.1
        libgssapi_krb5.so =>
/usr/lib/hpux64/libgssapi_krb5.so
        libkrb5.so =>   /usr/lib/hpux64/libkrb5.so
        libpthread.so.1 =>
/usr/lib/hpux64/libpthread.so.1
        libc.so.1 =>    /usr/lib/hpux64/libc.so.1
        libxti.so.1 =>  /usr/lib/hpux64/libxti.so.1
        libxti.so.1 =>  /usr/lib/hpux64/libxti.so.1
        libm.so.1 =>    /usr/lib/hpux64/libm.so.1
        libk5crypto.so =>
/usr/lib/hpux64/libk5crypto.so
        libcom_err.so =>        /usr/lib/hpux64/libcom_
err.so
        libk5crypto.so =>
/usr/lib/hpux64/libk5crypto.so
        libcom_err.so =>        /usr/lib/hpux64/libcom_
err.so
        libdl.so.1 =>   /usr/lib/hpux64/libdl.so.1
```

Make Sure GSSAPI Is Configured for SSH

Logging onto a system with keys does not provide that system with the means of getting a PAC from the domain controller. Without a PAC there is no group membership information for the user. Automated Kerberos ticket renewal will also be unavailable. So, when the ssh login hits the login restrictions in the account phase as it tests for the group memberships, it will not find the user's group information, causing an ssh error like this:

```
Not in an Allowed Group!
```

A workaround is to have each user log in once with a password. Subsequent logins with keys should work until the AD cache is flushed, after which the user will have to log in again.

Check the Configuration of SSH for SSO

Although PBIS automatically configures OpenSSH to support SSO through Kerberos using GSSAPI, it is worthwhile to review how PBIS does. Since you might need to configure or troubleshoot other applications for SSO, understanding the process will make it easier to apply the technique to other applications.

Note: Not all versions of OpenSSH support Kerberos. Versions older than 4.2p1 might not work or might work improperly. For important information on Kerberos and GSSAPI support in OpenSSH, see http://www.sxw.org.uk/computing/patches/openssh.html.

SSH Service Principal Name

The first thing that needs to be considered is the Kerberos service principal name (SPN) used by SSH and SSHD. The SPN is a string that identifies the service for which an authentication ticket is to be generated. In the case of SSH, the SPN has the form:

```
host/<server name>@<REALMNAME>
```

For example, when a user uses ssh to connect to a computer named `fozzie.mycorp.com`, the ssh program requests a service ticket for the SPN:

```
host/fozzie.example.com@EXAMPLE.COM
```

The Kerberos realm is the computer's domain name in uppercase letters.

System Keytab Generation

In order for Microsoft Active Directory to generate a Kerberos ticket for this SPN, a service account must exist for it. Additionally, a keytab must be created for the service account and placed on the sshd server. PBIS completely automates this operation. When a Linux or Unix computer is joined to AD, a machine account is created for the computer. If the computer is called `fozzie`, a machine account called `fozzie$` is created in AD. PBIS then automatically creates a keytab for the SPN and places it in the standard system location (typically, `/etc/krb5.keytab`).

User Keytab Generation

When the user runs the ssh program and OpenSSH determines that it will use Kerberos authentication, it will need to access a keytab for the user so that it can obtain a service ticket for the service/computer to which it is trying to connect. This keytab must be created using the user's account name and password. Manually, this can be performed by using the `kinit` utility. PBIS, however, does it automatically when the user logs on the computer. On most systems, the user keytab is placed in the `/tmp` directory and named `krb5cc_UID` where `UID` is the numeric user ID assigned by the system.

Configuring OpenSSH

PBIS automatically configures OpenSSH at both the client and server computer. On the client, the `ssh_config` file (typically in `/etc/ssh/ssh_config`) is modified. On the server, `sshd_config` (typically in `/etc/ssh/sshd_config`) is modified. PBIS adds the following lines of code to the right files if they are not already present and if they are required by the system's version of sshd:

In the server, the following lines must be present in `sshd_config`—if you are troubleshooting, make sure these lines are there:

```
GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
```

On the client, the following line must be present in `ssh_config`:

```
GSSAPIAuthentication yes
```

On the client, `GSSAPIDelegateCredentials yes` is an optional setting that instructs the ssh client to delegate the krb5 TGT to the destination machine when SSH single sign-on is used.

In addition, if any of the following options are valid for the system's version of sshd, they are required and configured by PBIS:

```
ChallengeResponseAuthentication yes
UsePAM yes
PAMAuthenticationViaKBDInt yes
KbdInteractiveAuthentication yes
```

Setting these options to `yes` instructs SSH to use the kbdinteractive ssh authentication mechanism and allows that mechanism to use PAM—settings that are required for PBIS to function properly.

For more information, see the man pages for ssh, sshd, and the comments in the ssh and sshd configuration files.

Testing SSO

With OpenSSH properly configured, demonstrating SSO support is simple: Log on a Linux or Unix machine running PBIS by using your Active Directory credentials and then use ssh to connect to another machine that is also running PBIS. OpenSSH should establish a connection without prompting for a username or password.

Platform-Specific Issues

If you are using Red Hat, CentOS, Fedora, FreeBSD, or AIX operating systems, review any of the following sections that are relevant for your operating system.

Red Hat and CentOS: Solve the SSO Problem

There is a known bug with some versions of Red Hat and CentOS that prevents SSO from working with SSH, SSHD, and PuTTY. The following versions are known to be affected:

- CentOS 5
- Red Hat Enterprise Linux 5

The system incorrectly concatenates the Kerberos ticket's service principal name on the target Linux computer. For example, in the final entry of the results of the `klist` command below, the full name of the service principal is cut off after the `@` symbol:

```
[EXAMPLE\fanthony@centos52 ~]$ /opt/pbis/bin/klist
```

```
Ticket cache: FILE:/tmp/krb5cc_1689257039
```

```
Default principal: fanthony@EXAMPLE.COM
```

```
Valid starting     Expires            Service principal
```

```
07/31/08 09:25:13  07/31/08 19:25:31
 krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

```
    renew until 08/07/08 09:25:13
```

```
07/31/08 09:25:31  07/31/08 19:25:31
 CENTOS52$@EXAMPLE.COM
```

```
    renew until 08/07/08 09:25:13
```

```
07/31/08 09:30:04  07/31/08 19:25:31
 host/centos52.example.com@
```

```
   renew until 08/07/08 09:25:13
```

To determine whether you need to implement the solution below on your Red Hat or CentOS computer, execute the following series of tests:

1. Connect to your target machine with SSH by using PuTTY and a valid Active Directory user. Be sure to use the FQDN of the host.

2. Execute the following command:

   `/opt/pbis/bin/klist`

   The results should look like this:

```
EXAMPLE\fanthony@centos52 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_1689257039
Default principal: fanthony@EXAMPLE.COM
Valid starting     Expires            Service
principal
07/31/08 09:25:13  07/31/08 19:25:31
 krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 08/07/08 09:25:13
07/31/08 09:25:31  07/31/08 19:25:31
 CENTOS52$@EXAMPLE.COM
renew until 08/07/08 09:25:13
```

3. SSH again to the same host and when prompted for the password, type CTRL+C.

4. Execute the `klist` command again:

   `/opt/pbis/bin/klist`

5. Check the results to determine whether there is an incorrectly concatenated service principal, as there is in the following output:

```
[EXAMPLE\fanthony@centos52 ~]$ klist
Ticket cache: FILE:/tmp/krb5cc_1689257039
Default principal: fanthony@EXAMPLE.COM
Valid starting     Expires            Service
principal
07/31/08 09:25:13  07/31/08 19:25:31
krbtgt/EXAMPLE.COM@EXAMPLE.COM
renew until 08/07/08 09:25:13
07/31/08 09:25:31  07/31/08 19:25:31
 CENTOS52$@EXAMPLE.COM
renew until 08/07/08 09:25:13
07/31/08 09:30:04  07/31/08 19:25:31
 host/centos52.example.com@
renew until 08/07/08 09:25:13
```

If the tests confirm that the problem exists, implement the following solution:

1. On Red Hat Enterprise Linux 5, make sure that the reverse PTR host definitions are defined in DNS.
2. On the target Linux computer, add the following line to `/etc/krb5.conf` under the `[domain_realm]` entry of the file:

   `.yourdomainname.com = YOURDOMAINNAME.COM`

   Example:

   ```
   [domain_realm]
   .example.com = EXAMPLE.COM
   ```

3. Restart SSHD by running the following command at the shell prompt:

   `/sbin/service sshd restart`

Red Hat and Fedora: Solve SSH Config Problem

On Fedora 14 and Red Hat 5, there is an issue with the configuration of the platform that blocks SSH SSO. You must either use a workaround to connect to the client or modify the `sshd_config` file on the server side. This section illustrates the problem and shows you how to connect to the client or fix the server.

After you join a domain with PBIS, Network Manager restarts and leaves the `/etc/hosts` file looking like this:

```
 [root@nile-fedora14 etc]# cat /etc/hosts
 10.100.0.26 nile-fedora14.nile-domain.example.com nile-
fedora14 # Added by NetworkManager
 127.0.0.1 localhost.localdomain localhost localhost4
 ::1 nile-fedora14.nile-domain.example.com nile-fedora14
localhost6 nile-fedora14.ramp.example.com
```

It should, however, look like this, but Network Manager keeps resetting it:

```
 10.100.0.26  nile-fedora14.nile-domain.example.com
nile-fedora14   # Added by NetworkManager
 127.0.0.1  nile-fedoar14.nile-domain.example.com nile-
fedora14 localhost.localdomain  localhost  localhost4
 ::1  nile-fedora14.nile-domain.example.com  nile-
fedora14   localhost6.localdomain6 localhost6
```

The configuration set by Network Manager blocks SSO because it ends up restricting reverse name lookups to ipv4 only.

When using the client, you can work around the problem by connecting by the external IP address. In other words, instead of using `ssh -l user nile-fedora14.nile-domain.example.com` to connect, use the following form:

```
ssh -l user 10.100.0.26
```

Alternatively, to fix the problem, you can turn off `GSSAPIStrictAcceptorCheck` in `sshd_config` on the server, but such a resolution might be unavailable when you do not have administrative access to the server or when doing so might cause intractable side effects or security holes.

Another, possibly cleaner way to fix the problem is to turn off reverse DNS lookups in Kerberos—but again, such a solution might result in side effects that block other applications or operations.

FreeBSD: Invalid Argument with SSHD

On FreeBSD, user names that are longer than 16 characters, including the domain name, exceed the FreeBSD username length limit. Attempts to connect by ssh with a user name that exceeds the limit can result in the following notification:

```
bvt-fbs72-64# ssh testuser1@localhost
Password:
Connection to localhost closed by remote host.
Connection to localhost closed.
```

The log for sshd, meanwhile, might show an error that looks something like this:

```
Oct  7 18:22:57 vermont02 sshd[66387]:
setlogin(EXAMPLE\adm.kathy):
Invalid argument
Oct  7 18:25:02 vermont02 sshd[66521]:
setlogin(EXAMPLE\adm.kathy):
Invalid argument
```

Although `testuser1` is less than 16 characters, when you use the `id` command to check the account, something longer than 16 characters is returned:

```
[root@bvt-fbs72-64 /home/testuser]# id testuser1
uid=1100(BVT-FBS72-64\testuser1) gid=1801(BVT-FBS72-
64\testgrp)
groups=1801(BVT-FBS72-64\testgrp)
```

The result of the `id` command exceeds the FreeBSD username length limit. There are several solutions: set the default domain, change the user name to 16 characters or less, or with PBIS Enterprise use aliases. Keep in mind, though, that aliases will not solve the problem in relation to the PBIS local provider.

AIX and Red Hat: Set Reverse PTR Host Definitions for SSO

For single sign-on with SSH to work on Red Hat Enterprise Linux 5 and AIX, reverse PTR host definitions must be set in DNS.

AIX: Configure for Outbound Single Sign-On

On AIX 5.3, client-side SSH is not set up by default. Here is how to configure it so that it will work with PBIS.

1. On your AIX 5.3 computer, make sure the network authentication service, version 1.4.0.8, is installed; example:

   ```
   -bash-3.00$ lslpp -l | grep krb
   krb5.client.rte 1.4.0.8 COMMITTED Network
   Authentication Service
   ```

   If it is not installed, obtain it from the IBM AIX website at http://www.ibm.com/developerworks/aix/library/au-nas_relatedtech/index.html and install it.

2. After joining an Active Directory domain with PBIS, append the following lines to the end of `/etc/krb5/krb5.conf`:

   ```
   [domain_realm]
   .demo.example.com = DEMO.EXAMPLE.COM
   demo.example.com = DEMO.EXAMPLE.COM
   ```

3. Make sure that `/etc/krb5/krb5.conf` links to `/etc/krb5.conf`.

4. Also make sure that `/etc/krb5/krb5.keytab` links to `/etc/krb5.keytab`.
5. Make a backup of the credentials directory by executing the following command as root:

   ```
   mv /var/krb5/security/creds /var/krb5/security/creds_old
   ```

6. As root, make a symbolic link to the `/tmp` directory so that the AIX Kerberos libraries can access the directory in which PBIS stores its credential caches:

   ```
   ln -s /tmp /var/krb5/security/creds
   ```

7. Open `/etc/environment`—which contains the list of environmental variables that are set when a user logs on—and add the following line to the end of it:

   ```
   KRB5_CONFIG=/var/lib/pbis/krb5-
   affinity.conf:/etc/krb5.conf
   ```

8. If you are logged on the machine whose environmental variable you changed, you must log off and log on again for the change to take effect.

More Information

For additional troubleshooting information, see the following:

- Solve Logon Problems on Linux or Unix
- Troubleshooting Domain-Join Problems

For information about troubleshooting integration with Samba, see the *Samba Integration Guide for PBIS*.

For an overview of commands such as `rpm` and `dpkg` that can help troubleshoot PBIS packages on Linux and Unix platforms, see *Package Management Commands*.

Troubleshooting Domain-Join Problems

Here are the top 10 reasons that an attempt to join a domain fails:

1. Root was not used to run the domain-join command (or to run the domain-join graphical user interface).
2. The user name or password of the account used to join the domain is incorrect.
3. The name of the domain is mistyped.
4. The name of the OU is mistyped.
5. The local hostname is invalid.
6. The domain controller is unreachable from the client because of a firewall or because the NTP service is not running on the domain controller. (See Make Sure Outbound Ports Are Open and Diagnose NTP on Port 123.)
7. The client is running RHEL 2.1 and has an old version of SSH.
8. On SUSE, GDM (dbus) must be restarted. This daemon cannot be automatically restarted if the user logged on with the graphical user interface.
9. On HP-UX and Solaris, `dtlogin` must be restarted. This daemon cannot be automatically restarted if the user logged on with the HP-UX or Solaris graphical user interface. To restart `dtlogin`, run the following command: `/sbin/init.d/dtlogin.rc start`
10. SELinux is turned on by being set to either `enforcing` or `permissive`—which is especially likely on Fedora and some versions of Red Hat. SELinux must be set to `disabled` before the computer can be joined to the domain.
    To turn off SELinux, edit the following file, which is the primary configuration file for enabling and disabling SELinux:

    `/etc/sysconfig/selinux`

    or

    `/etc/selinux/config`

    For instructions on how to edit the file to disable SELinux, see the SELinux man page.

See Also

Generate a Domain-Join Log

Solve Domain-Join Problems

To troubleshoot problems with joining a Linux computer to a domain, perform the following series of diagnostic tests sequentially on the Linux computer with a root account. The tests can also be used to troubleshoot domain-join problems on a Unix or Mac OS X computer; however, the syntax of the commands on Unix and Mac might be slightly different.

The procedures in this topic assume that you have already checked whether the problem falls under the Top 10 Reasons Domain Join Fails. It is also recommended that you generate a domain-join log.

Verify that the Name Server Can Find the Domain

Run the following command as root:

```
nslookup YourADrootDomain.com
```

Make Sure the Client Can Reach the Domain Controller

You can verify that your computer can reach the domain controller by pinging it:

```
ping YourDomainName
```

Check DNS Connectivity

The computer might be using the wrong DNS server or none at all. Make sure the nameserver entry in `/etc/resolv.conf` contains the IP address of a DNS server that can resolve the name of the domain you are trying to join. The IP address is likely to be that of one of your domain controllers.

Make Sure `nsswitch.conf` Is Configured to Check DNS for Host Names

The `/etc/nsswitch.conf` file must contain the following line. (On AIX, the file is `/etc/netsvc.conf`.)

```
hosts: files dns
```

Computers running Solaris, in particular, may not contain this line in `nsswitch.conf` until you add it.

Generate a Domain-Join Log

To log information about your attempt to join a domain, you can use the command-line utility's `log` option with the `join` command. The `log` option captures information about the attempt to join the domain on the screen or in a file.

- To display the information in the terminal, execute the following command; the dot after the `logfile` option denotes that the information is to be shown in the console:
  `domainjoin-cli --logfile . join` *`domainName userName`*
- To save the information in a log file, execute the following command:
  `domainjoin-cli --logfile path join` *`domainName userName`*
  Example:

  ```
  domainjoin-cli --logfile /var/log/domainjoin.log join
     example.com Administrator
  ```

After you generate a log, review it for information that might help solve the problem.

Ensure that DNS Queries Use the Correct Network Interface Card

If the computer is multi-homed, the DNS queries might be going out the wrong network interface card. Temporarily disable all the NICs except for the card on the same subnet as your domain controller or DNS server and then test DNS lookups to the AD domain. If this works, re-enable all the NICs and edit the local or network routing tables so that the AD domain controllers are accessible from the host.

Determine If DNS Server Is Configured to Return SRV Records

Your DNS server must be set to return SRV records so the domain controller can be located. It is common for non-Windows (bind) DNS servers to not be configured to return SRV records.

Diagnose it by executing the following command:

`nslookup -q=srv _ldap._tcp.` *`ADdomainToJoin.com`*

Make Sure that the Global Catalog Is Accessible

The global catalog for Active Directory must be accessible. A global catalog in a different zone might not show up in DNS. Diagnose it by executing the following command:

`nslookup -q=srv _ldap._tcp.gc._msdcs.` *`ADrootDomain.com`*

From the list of IP addresses in the results, choose one or more addresses and test whether they are accessible on Port 3268 by using telnet.

```
telnet 192.168.100.20 3268
Trying 192.168.100.20... Connected to sales-dc.example.com
   (192.168.100.20). Escape character is '^]'. Press the
   Enter key to close the connection: Connection closed by
   foreign host.
```

Verify that the Client Can Connect to the Domain on Port 123

The following test checks whether the client can connect to the domain controller on Port 123 and whether the Network Time Protocol (NTP) service is running on the domain controller. For the client to join the domain, NTP—the Windows time service—must be running on the domain controller.

On a Linux computer, run the following command as root:

`ntpdate -d -u DC_hostname`
Example: `ntpdate -d -u sales-dc`

For more information, see Diagnose NTP on Port 123.

In addition, check the logs on the domain controller for errors from the source named `w32tm`, which is the Windows time service.

FreeBSD: Run `ldconfig` If You Cannot Restart Computer

When installing PBIS on a new FreeBSD computer with nothing in `/usr/local`, run `/etc/rc.d/ldconfig start` after the installation if you cannot restart the computer. Otherwise, `/usr/local/lib` will not be in the library search path.

Ignore Inaccessible Trusts

An inaccessible trust can block you from successfully joining a domain. If you know that there are inaccessible trusts in your Active Directory network, you can set PowerBroker Identity Services to ignore all the trusts before you try to join a domain. To do so, use the `config` tool to modify the values of the `DomainManagerIgnoreAllTrusts` setting.

First, list the available trust settings:

/opt/pbis/bin/config --list | grep -i trust

The results will look something like this. The setting at issue is `DomainManagerIgnoreAllTrusts`.

```
DomainManagerIgnoreAllTrusts
DomainManagerIncludeTrustsList
DomainManagerExcludeTrustsList
```

Second, list the details of the `DomainManagerIgnoreAllTrusts` setting to see the values it accepts:

```
[root@rhel5d bin]# ./config --details
DomainManagerIgnoreAllTrusts
Name: DomainManagerIgnoreAllTrusts
Description: When true, ignore all trusts during domain
enumeration.
Type: boolean
Current Value: false
Accepted Values: true, false
Current Value is determined by local policy.
```

Third, change the setting to `true` so that PBIS will ignore trusts when you try to join a domain.

```
[root@rhel5d bin]# ./config DomainManagerIgnoreAllTrusts
true
```

Finally, check to make sure the change took effect:

```
[root@rhel5d bin]# ./config --show
DomainManagerIgnoreAllTrusts
boolean
true
local policy
```

Now try to join the domain again. If successful, keep in mind that only users and groups who are in the local domain will be able to log on the computer.

In the example output above that shows the setting's current values, `local policy` is listed—meaning that the setting is managed locally through `config` because a PBIS Group Policy setting is not managing the setting. Typically, with PBIS Enterprise, you would manage the `DomainManagerIgnoreAllTrusts` setting by using the corresponding Group Policy setting, but you cannot apply Group Policy Objects (GPOs) to the computer until after it is added to the domain. The corresponding PBIS policy setting is named `Lsass: Ignore all trusts during domain enumeration`. For more information on the domain manager policy settings to configure whitelists and blacklists for trusts, see the *PowerBroker Identity Services Group Policy Administration Guide*.

For information on the arguments of config, run the following command:

/opt/pbis/bin/config --help

Resolve Error Messages

This section lists solutions to common errors that can occur when you try to join a domain.

Configuration of Krb5

Error Message:

```
Warning: A resumable error occurred while processing a
module.
Even though the configuration of 'krb5' was executed, the
configuration did not
fully complete. Please contact BeyondTrust support.
```

Solution:

Delete `/etc/krb5.conf` and try to join the domain again.

Diagnose NTP on Port 123

When you use the PBIS domain-join utility to join a Linux or Unix client to a domain, the utility might be unable to contact the domain controller on Port 123 with UDP. The PBIS agent requires that Port 123 be open on the client so that it can receive NTP data from the domain controller. In addition, the time service must be running on the domain controller.

You can diagnose NTP connectivity by executing the following command as root at the shell prompt of your Linux computer:

`ntpdate -d -u DC_hostname`

Example: `ntpdate -d -u sales-dc`

If all is well, the result should look like this:

```
[root@rhel44id ~]# ntpdate -d -u sales-dc
2 May 14:19:20 ntpdate[20232]: ntpdate 4.2.0a@1.1190-r
Thu Apr 20 11:28:37 EDT 2006 (1)
Looking for host sales-dc and service ntp
host found : sales-dc.example.com
transmit(192.168.100.20)
receive(192.168.100.20)
transmit(192.168.100.20)
receive(192.168.100.20)
transmit(192.168.100.20)
receive(192.168.100.20)
transmit(192.168.100.20)
```

```
receive(192.168.100.20)
transmit(192.168.100.20)
server 192.168.100.20, port 123
stratum 1, precision -6, leap 00, trust 000
refid [LOCL], delay 0.04173, dispersion 0.00182
transmitted 4, in filter 4
reference time:    cbc5d3b8.b7439581  Fri, May  2 2008
10:54:00.715
originate timestamp: cbc603d8.df333333  Fri, May  2 2008
14:19:20.871
transmit timestamp:  cbc603d8.dda43782  Fri, May  2 2008
14:19:20.865
filter delay:  0.04207  0.04173  0.04335  0.04178
 0.00000  0.00000  0.00000  0.00000
filter offset: 0.009522 0.008734 0.007347 0.005818
 0.000000 0.000000 0.000000 0.000000
delay 0.04173, dispersion 0.00182
offset 0.008734
2 May 14:19:20 ntpdate[20232]: adjust time server
192.168.100.20 offset 0.008734 sec
```

Output When There Is No NTP Service

If the domain controller is not running NTP on Port 123, the command returns a response such as `no server suitable for synchronization found`, as in the following output:

```
5 May 16:00:41 ntpdate[8557]: ntpdate 4.2.0a@1.1190-r Thu
Apr 20 11:28:37 EDT 2006 (1)
Looking for host RHEL44ID and service ntp
host found : rhel44id.example.com
transmit(127.0.0.1)
transmit(127.0.0.1)
transmit(127.0.0.1)
transmit(127.0.0.1)
transmit(127.0.0.1)
127.0.0.1: Server dropped: no data
server 127.0.0.1, port 123
stratum 0, precision 0, leap 00, trust 000
refid [127.0.0.1], delay 0.00000, dispersion 64.00000
transmitted 4, in filter 4
reference time:    00000000.00000000  Wed, Feb  6 2036
22:28:16.000
originate timestamp: 00000000.00000000  Wed, Feb  6 2036
22:28:16.000
transmit timestamp:  cbca101c.914a2b9d  Mon, May  5 2008
16:00:44.567
filter delay:  0.00000  0.00000  0.00000  0.00000
 0.00000  0.00000  0.00000  0.00000
filter offset: 0.000000 0.000000 0.000000 0.000000
```

```
 0.000000 0.000000 0.000000 0.000000
delay 0.00000, dispersion 64.00000
offset 0.000000
5 May 16:00:45 ntpdate[8557]: no server suitable for
synchronization found
```

Turn off Apache to Join a Domain

The Apache web server locks the keytab file, which can block an attempt to join a domain. If the computer is running Apache, stop Apache, join the domain, and then restart Apache.

Configuring Clients After PBIS Agent Installation

After you have installed the PBIS agent on client computers, you can configure end-user settings for the agent, add domain accounts to local groups, and add Active Directory entries to your sudoers file. If PBIS is not finding your sudoers file automatically, you can specify a search path for the file. On AIX computers, after you have installed the PBIS agent, you can configure the computer to monitor users who log on with Active Directory credentials.

Modify Settings with the Config Tool

To quickly change an end-user setting for the PBIS agent, you can run the `config` command-line tool as root:

/opt/pbis/bin/config

The syntax to change the value of a setting is as follows, where `setting` is replaced by the registry entry that you want to change and `value` by the new value that you want to set:

```
/opt/pbis/bin/config setting value
```

Here is an example of how to use `config` to change the `AssumeDefaultDomain` setting:

```
[root@rhel5d bin]# ./config --detail AssumeDefaultDomain
❶
Name: AssumeDefaultDomain
Description: Apply domain name prefix to account name at
logon
Type: boolean
Current Value: false
Accepted Values: true, false
Current Value is determined by local policy.

[root@rhel5d bin]# ./config AssumeDefaultDomain true ❷

[root@rhel5d bin]# ./config --show AssumeDefaultDomain ❸
boolean
true
local policy
```

❶ Use the `--detail` option to view the setting's current value and to determine the values that it accepts.

❷ Set the value to `true`.

❸ Use the `--show` option to confirm that the value was set to `true`.

To view the settings that you can change with `config`, execute the following command:

```
/opt/pbis/bin/config --list
```

You can also import and apply a number of settings with a single command by using the `--file` option combined with a text file that contains the settings that you want to change followed by the values that you want to set. Each setting-value pair must be on a single line. For example, the contents of my flat file, named `newRegistryValuesFile` and saved to the desktop of my Red Hat computer, looks like this:

```
AssumeDefaultDomain true
RequireMembershipOf "example\\support"
"example\\domain^admins"
HomeDirPrefix /home/ludwig
LoginShellTemplate /bash/sh
```

To import the file and automatically change the settings listed in the file to the new values, I would execute the following command as root:

```
/opt/pbis/bin/config --file
/root/Desktop/newRegistryValuesFile
```

Add Domain Accounts to Local Groups

You can add domain users to your local groups on a Linux, Unix, and Mac OS X computer by placing an entry for the user or group in the `/etc/group` file. Adding an entry for an Active Directory user to your local groups can give the user local administrative rights. The entries must adhere to the following rules:

- Use the correct case; entries are case sensitive.
- Use a user or group's alias if the user or group has one in Active Directory.
- If the user or group does not have an alias, you must set the user or group in the PBIS canonical name format of `NetBIOSdomainName\SAMaccountName`.
  Note: For users or groups with an alias, the PBIS canonical name format is the alias, which you must use; you cannot use the format of `NetBIOS domain name\SAM account name`.

So, for users and groups without an alias, the form of an entry is as follows:

```
root:x:0:EXAMPLE\kristeva
```

For users and groups with an alias, the form of an entry is as follows:

```
root:x:0:kris
```

In `/etc/group`, the slash character separating the domain name from the account name does not typically need to be escaped.

Tip: On Ubuntu, you can give a domain user administrative privileges by adding the user to the `admin` group as follows:

```
admin:x:119:EXAMPLE\bakhtin
```

On a Mac OS X computer, you can AD users to a local group with Apple's directory service command-line utility: dscl. In dscl, go to the `/Local/Default/Groups` directory and then add users to a group by using the `append` command.

Configure Entries in Your sudoers Files

When you add Active Directory entries to your sudoers file—typically, `/etc/sudoers`—you must adhere to at least the following rules:

- `ALL` must be in uppercase letters.
- Use a slash character to escape the slash that separates the Active Directory domain from the user or group name.
- Use the correct case; entries are case sensitive.
- Use a user or group's alias if the user or group has one in Active Directory.
- If the user or group does not have an alias, you must set the user or group in the PBIS canonical name format of `NetBIOSdomainName\SAMaccountName` (and escape the slash character).
  Note: For users or groups with an alias, the PBIS canonical name format is the alias, which you must use; you cannot use the format of `NetBIOS domain name\SAM account name`.

So, for users and groups without an alias, the form of an entry in the sudoers file is as follows:

```
DOMAIN\\username
```

```
DOMAIN\\groupname
```

Example entry of a group:

```
% EXAMPLE\\LinuxFullAdmins ALL=(ALL) ALL
```

Example entry of a user with an alias:

```
kyle ALL=(ALL) ALL
```

For more information about how to format your sudoers file, see your computer's man page for sudo.

Check a User's Canonical Name on Linux

To determine the canonical name of a PBIS user on Linux, execute the following command, replacing the domain and user in the example with your domain and user:

```
getent passwd example.com\\hab
```

```
EXAMPLE\hab:x:593495196:593494529: Jurgen
Habermas:/home/local/ EXAMPLE/ hab:/bin/ sh
```

In the results, the user's PBIS canonical name is the first field.

Specify a sudoers Search Path

Although PowerBroker Identity Services searches a number of common locations for your sudoers file, on some platforms PBIS might not find it. In such cases, you can specify the location of your sudoers file by adding the following line to the Sudo GP Extension section of `/etc/pbis/grouppolicy.conf`:

```
SudoersSearchPath = /your/search/path
```

Example: `SudoersSearchPath = "/opt/sfw/etc";`

Here is an example in the context of the `/etc/pbis/grouppolicy.conf` file:

```
[{20D139DE-D892-419f-96E5-0C3A997CB9C4}]
Name = "PBIS Enterprise Sudo GP Extension";
DllName = "liblwisudo.so";
EnableAsynchronousProcessing = 0;
NoBackgroundPolicy = 0;
NoGPOListChanges = 1;
NoMachinePolicy = 0;
NoSlowLink = 1;
NoUserPolicy = 1;
PerUserLocalSettings = 0;
ProcessGroupPolicy = "ProcessSudoGroupPolicy";
ResetGroupPolicy = "ResetSudoGroupPolicy";
RequireSuccessfulRegistry = 1;
SudoersSearchPath = "/opt/sfw/etc";
```

AIX: Create Audit Classes to Monitor Events

On AIX, you can create audit classes to monitor the activities of users who log on with their Active Directory credentials. The file named `/etc/pbis/auditclasses.sample` is a template that you can use to create audit classes for AD users.

To create and configure an audit class, make a copy of the file, name it `/etc/pbis/auditclasses`, and then edit the file to specify the audit classes that you want.

After you configure audit classes for a user, the auditing will take place the next time the user logs in.

The sample PBIS auditclasses file looks like this:

```
#
# Sample auditclasses file.
#
# A line with no label specifies the default audit
classes for
# users that are not explicitly listed:
#
general, files
#
# A line starting with a username specifies the audit
classes for
# that AD user.  The username must be specified as the
"canonical"
# name for the user: either "DOMAIN\username" or just
"username"
# if "--assumeDefaultDomain yes" was passed to
domainjoin-cli
# with "--userDomainPrefix DOMAIN".  In PBIS Enterprise,
if
# the user has an alias specified in the cell the alias
name must
# be used here.
#
DOMAIN\user1: general, files, tcpip
user2: general, cron
#
# A line starting with an @ specifies the audit classes
for members
# of an AD group.  These classes are added to the audit
classes
# for the user (or the default, if the user is not listed
here).
# Whether to specify "DOMAIN\groupname" or just
"groupname" follows
# the same rules as for users.
#
@DOMAIN\mail_users: mail
group2: cron
```

For information on AIX audit classes, see the IBM documentation for your version of AIX.

Troubleshooting the PBIS Agent

This chapter contains information on how to troubleshoot the PBIS agent, including the authentication service, the input-output service, and the network logon service.

In addition to the information in this chapter, refer to the following topics for information about specific issues:

- Troubleshooting Domain Join Problems
- Solve Logon Problems on Linux, Unix, or Mac
- Solve Logon Problems from Windows
- Troubleshooting SSH SSO Problems
- Monitoring Events with the Event Log

For information about how to use specific commands, refer to the Command-Line Reference.

Troubleshooting guidance related to specific subjects is also provided in other guides:

- For information about troubleshooting the Group Policy Agent, see the *PowerBroker Identity Services Group Policy Administration Guide*.
- For information about troubleshooting Samba integration, see the *PowerBroker Identity Services Samba Guide*.
- For an overview of commands such as `rpm` and `dpkg` that can help troubleshoot PBIS packages on Linux and Unix platforms, see *PowerBroker Identity Services Package Management Commands*.

PBIS Services

The PBIS Service Manager lets you troubleshoot all the PBIS services from a single command-line utility. You can, for example, check the status of the services and start or stop them. The service manager is the preferred method for restarting a service because it automatically identifies a service's dependencies and restarts them in the right order.

To list the status of the services, run the following command with superuser privileges at the command line:

/opt/pbis/bin/lwsm list

Here is an example:

```
[root@rhel5d bin]# /opt/pbis/bin/lwsm list
lwreg       running (standalone: 1920)
dcerpc      running (standalone: 2544)
eventlog    running (standalone: 2589)
lsass       running (standalone: 2202)
lwio        running (standalone: 2191)
netlogon    running (standalone: 2181)
rdr         running (io: 2191)
```

To restart the `lsass` service, run the following command with superuser privileges:

/opt/pbis/bin/lwsm restart lsass

To view all the service manager's commands and arguments, execute the following command:

/opt/pbis/bin/lwsm --help

For more about PBIS services, see Services.

Check the Status of the Authentication Service

You can check the status of the authentication service on a Unix or Linux computer running the PBIS agent by executing the following command at the shell prompt as the root user:

```
/opt/pbis/bin/lwsm status lsass
```

If the service is not running, execute the following command:

```
/opt/pbis/bin/lwsm start lsass
```

Check the Status of the DCE/RPC Service

The PBIS DCE/RPC service handles communication between PBIS clients and Microsoft Active Directory.

On Linux and Unix

You can check the status of `dcerpc` on a Unix or Linux computer running the PBIS agent by executing the following command as the root user:

```
/opt/pbis/bin/lwsm status dcerpc
```

If the service is not running, execute the following command:

```
/opt/pbis/bin/lwsm start dcerpc
```

On Mac OS X

On a Mac OS X computer, you cannot use the `status` command, but you can monitor the service by using Activity Monitor:

1. In Finder, click **Applications**, click **Utilities**, and then click **Activity Monitor**.
2. In the list under **Process Name**, make sure `dcerpc` appears. If the process does not appear in the list, you might need to start it.
3. To monitor the status of the process, in the list under **Process Name**, click the process, and then click **Inspect**.

Check the Status of the Network Logon Service

The `netlogon` service detects the optimal domain controller and global catalog and caches the data.

On Linux and Unix

You can check the status of `netlogon` on a Unix or Linux computer running the PBIS agent by executing the following command as the root user:

```
/opt/pbis/bin/lwsm status netlogon
```

If the service is not running, execute the following command:

```
/opt/pbis/bin/lwsm start netlogon
```

On Mac OS X

On a Mac OS X computer, you cannot use the `status` command, but you can monitor the service by using Activity Monitor:

1. In Finder, click **Applications**, click **Utilities**, and then click **Activity Monitor**.
2. In the list under **Process Name**, make sure `netlogon` appears. If the process does not appear in the list, you might need to start it.
3. To monitor the status of the process, in the list under **Process Name**, click the process, and then click **Inspect**.

Check the Status of the Input-Output Service

The PBIS input-output service—`lwio`—communicates over SMB with external SMB servers and internal processes.

You can check the status of `lwio` on a Unix, Linux, or Mac computer running the PBIS agent by executing the following command as the root user:

```
/opt/pbis/bin/lwsm status lwio
```

If the service is not running, execute the following command:

```
/opt/pbis/bin/lwsm start lwio
```

Restart the Authentication Service

The authentication service handles authentication, authorization, caching, and idmap lookups. For more information, see PBIS Agent.

You can restart the PBIS authentication service by executing the following command at the shell prompt:

```
/opt/pbis/bin/lwsm restart lsass
```

To stop the service, type this command:

```
/opt/pbis/bin/lwsm stop lsass
```

To start the service, type this command:

```
/opt/pbis/bin/lwsm start lsass
```

Restart the DCE/RPC Service

The PBIS DCE/RPC service helps route remote procedure calls between computers on a network by serving as an end-point mapper. For more information, see PBIS Agent.

You can restart the PBIS DCE/RPC service by executing the following command at the shell prompt:

```
/opt/pbis/bin/lwsm restart dcerpc
```

To stop the service, type this command:

```
/opt/pbis/bin/lwsm stop dcerpc
```

To start the service, type this command:

```
/opt/pbis/bin/lwsm start dcerpc
```

Restart the Network Logon Service

The `netlogon` service determines the optimal domain controller and global catalog and caches the data. For more information and a list of start-order dependencies, see PBIS Agent.

You can restart the PBIS network logon service by executing the following command at the shell prompt:

```
/opt/pbis/bin/lwsm restart netlogon
```

To stop the service, type this command:

```
/opt/pbis/bin/lwsm stop netlogon
```

To start the service, type this command:

```
/opt/pbis/bin/lwsm start netlogon
```

Restart the Input-Output Service

The PBIS input-output service—`lwio`—communicates over SMB with SMB servers; authentication is with Kerberos 5.

You can restart the input-output service by executing the following command at the shell prompt:

```
/opt/pbis/bin/lwsm restart lwio
```

To stop the service, type this command:

```
/opt/pbis/bin/lwsm stop lwio
```

To start the service, type this command:

```
/opt/pbis/bin/lwsm start lwio
```

Note: If you start the `lwio` service and the rdr service does not also start, use the following command to start the `rdr` service:
`/opt/pbis/bin/lwsm start rdr`

Logging

Logging can help identify and solve problems. There are debug logs for the following services in PBIS Open and PBIS Enterprise:

- **`lsass`** - The authentication service. Generate a debug log for `lsass` when you need to troubleshoot authentication errors or failures.
- **PAM** - The pluggable authentication modules used by PBIS. Create a debug log for PAM when you need to troubleshoot logon or authentication problems.
- **`netlogon`** - The site affinity service that detects the optimal domain controller and global catalog. Generate a debug log for `netlogon` when you need to troubleshoot problems with sending requests to domain controllers or getting information from the global catalog.
- **`lwio`** - The input-output service that manages interprocess communication.
- **`eventlog`** - The event collection service. Generate a debug log for `eventlog` to troubleshoot the collection and processing of security events.
- **`lwreg`** - The PBIS registry service. Generate a debug log for `lwreg` to troubleshoot ill-fated configuration changes to the registry.
- **`lwsm`** - The service manager.
- **`reapsysl`** - Part of the data collection service. Capture a debug log for `reapsysl` to investigate the collection and processing of events.
- **Mac OS X directory service plug-in**

In addition, the following services are part of PBIS Enterprise only:

- **`gpagent`** - The Group Policy agent. Generate a debug log for `gpagent` to troubleshoot the application or processing of Group Policy Objects (GPOs).
- **`eventfwd`** - The event forwarding service. Generate a debug log to verify that the service is receiving events and forwarding them to a collector server.
- **`lwsc`** - The smart card service. Gather logging information for the smart card service when card-insertion or card-removal behavior is other than expected.
- **`lwpkcs11`** - A service that aids in logging on and logging off with a smart card. Gather logging information about it when there is a problem logging on or logging off with a smart card.

By default, log messages are processed by syslog, typically through the daemon facility. Although the path and file name of the log vary by platform, they typically appear in a subdirectory of `/var/log`. Note that when you change the log level of a PBIS service to `debug`, you may also need to update syslog configuration (typically `/etc/syslog.conf`) with the following command and then restart the syslog service:

```
*.debug /tmp/debug.log
```

Alternatively, you can log directly to a file, as the procedure to Change the Target illustrates.

Log levels can be changed temporarily or permanently. The following log levels are available for most PBIS services: `always`, `debug`, `error`, `warning`, `info`, `verbose`, and `trace`. The default is `error`. To troubleshoot, it is recommended that you change the level to `debug`. To conserve disk space, it is recommended that you set the log level back to the default level when you finish troubleshooting.

To temporarily change the log level, you can use `/opt/pbis/bin/lwsm` to specify the log level and whether to log to the syslog or directly to a file. To permanently change the log level, you must modify the service's entry in the PBIS registry.

Tip: Ignore errors caused by `reapsysl` service

The following are the pipes by which `su`, `sudo`, and local user (root) sshd logons are captured with the PBIS auditing system. They are system pipes created by the `reapsysl` service. PBIS cannot start the `reapsysl` service before syslog starts because of a complex series of dependencies on the system. Therefore, these errors are generated and should be ignored. `Reapsysl` will recreate the pipes as necessary.

```
robbie@example:~$ sudo ls -la /var/lib/pbis/syslog-
   reaper/ total 28
drwx------ 2 root root 4096 Mar 7 12:54 .
drwxr-xr-x 8 root root 4096 May 10 13:27 ..
prwx------ 1 root root 0 Mar 7 12:54 error
prwx------ 1 root root 0 Mar 7 12:54 information
prwx------ 1 root root 0 Mar 7 12:54 warning
```

Temporarily Change the Log Level and Target for a Service

The service manager supports per-service, per-facility logging. Each service has a default log target (syslog) and level (WARNING).

Change the Target

You can use the following command to change the log target for a particular service and facility to log to a file:

```
/opt/pbis/bin/lwsm set-log-target <service> <facility> file
   <path>
```

You can use the following command to change the log target for particular service and facility to the syslog:

```
/opt/pbis/bin/lwsm set-log-target <service> <facility>
   syslog
```

The service can be any PBIS service except `dcerpc`, which has its own logging mechanism.

The facility is a portion of the service and the default facility is accessed as -. For example, to target the logging messages from default facility of `lsass` to a file `/var/log/lsass.log`:

```
/opt/pbis/bin/lwsm set-log-target lsass - file
   /var/log/lsass.log
```

If you want to debug the interprocess communications of `lsass` (something rarely required), you can use the `lsass-ipc` facility:

```
/opt/pbis/bin/lwsm set-log-target lsass lsass-ipc file
   /tmp/lsass-ipc.log
```

Change the Log Level

To change the level of logging in the default facility of `lsass` to debug:

```
/opt/pbis/bin/lwsm set-log-level lsass - debug
```

The supported log levels are `always`, `error`, `warning`, `info`, `verbose`, `debug`, `trace`.

Changing the log level temporarily can help you isolate and capture information when a command or operation fails. For example, if you run a command and it fails, you can change the log level and then run the command again to get information about the failure.

View Log Settings

To view the current level and target of logging of a service, enter the following command:

```
/opt/pbis/bin/lwsm get-log <service>
```

For example, entering the following command

```
/opt/pbis/bin/lwsm get-log lsass
```

produces the following result

```
<default>: syslog LOG_DAEMON at ERROR
```

This indicates that the `lsass` service's default log level is `error` and is directed to syslog's daemon facility.

Generate a Domain-Join Log

To help troubleshoot problems with joining a domain, you can use the command-line utility's `logfile` option with the `join` command. The `logfile` option captures information about the attempt to join the domain on the screen or in a file. When an attempt to join a domain fails, a log is generated by default at `/var/log/domainjoin-cli.log` or `/var/adm/domainjoin-cli.log`.

- To display the information in the terminal, execute the following command; the dot after the `logfile` option denotes that the information is to be shown in the console:
  ```
  domainjoin-cli --logfile . join domainName userName
  ```

- To save the information in a log file, execute the following command:
  `domainjoin-cli --logfile path join domainName userName`

  Example:

  `domainjoin-cli --logfile /var/log/domainjoin.log join example.com Administrator`

Generate a PAM Debug Log

You can set the level of reporting in the PAM debug log for the PBIS authentication service on a Linux or Unix computer. PAM stands for pluggable authentication modules.

The log levels are disabled, error, warning, info, and verbose. The logged data is sent to your system's syslog message repository for security and authentication. The location of the repository varies by operating system. Here are the typical locations for a few platforms:

- Ubuntu: `/var/log/auth.log`
- Red Hat: `/var/log/secure`
- Solaris: `/var/log/authlog`
- Mac OS X: `/var/log/secure.log`

The following procedure demonstrates how to change the value of the PAM key's `LogLevel` entry with the `config` command-line utility.

First, use the `details` option to list the values that the `DomainManagerIgnoreAllTrusts` setting accepts:

```
/opt/pbis/bin/config --details PAMLogLevel
Name: PAMLogLevel
Description: Configure PAM lsass logging detail level
Type: string
Current Value: "disabled"
Acceptable Value: "disabled"
Acceptable Value: "error"
Acceptable Value: "warning"
Acceptable Value: "info"
Acceptable Value: "verbose"
Current Value is determined by local policy.
```

Now, as root change the setting to `error` so that PBIS will log PAM errors:

/opt/pbis/bin/config PAMLogLevel error

Finally, confirm that the change took effect:

```
/opt/pbis/bin/config --show PAMLogLevel
string
error
local policy
```

For more information on the arguments of config, run the following command:

/opt/pbis/bin/config --help

Generate a Directory Service Log on a Mac

To troubleshoot logon failures on a Mac OS X computer, you can generate a debug-level directory service log. For information on turning on debug-level logs, see Enabling Directory Service Debug Logging on the Apple support website.

Using the `killall -USR1` command that Apple suggests, however, puts the directory service into debug logging mode for only about 5 minutes. Instead, try using the following commands:

```
sudo touch
/Library/Preferences/DirectoryService/.DSLogDebugAtStart
sudo killall DirectoryService
```

Reproduce the error and then scan the logs named `DirectoryService.debug.log` in `/Library/Logs/DirectoryService`. Look for messages containing the string `LWEDS`, which indicates that they are produced by the PBIS directory service plug-in.

Examine the logs from the time the user entered a password. If the logs suggest that there may be a networking issue, obtain a tcpdump from the time the password is entered until you notice the logon failure:

```
tcpdump -s0 -wnetwork.pcap
```

When you are done troubleshooting, turn off debug logging and restart the directory service by issuing the following commands:

```
sudo rm
/Library/Preferences/DirectoryService/.DSLogDebugAtStart
sudo killall DirectoryService
```

On Mac OS X Lion

On the Mac OS X Lion operating system, use the following command to enable logging:

```
sudo odutil set log debug
```

Logs are stored in `/var/log/opendirectoryd.log`.

You can revert to standard logging by using the following command:

```
odutil set log default
```

Generate a Network Trace

Execute the following command in a separate session to dump network traffic as the root user and interrupt the trace with CTRL-C:

```
tcpdump -s 0 -i eth0 -w trace.pcap
```

The result should look something like this:

```
tcpdump: listening on eth0
28 packets received by filter
0 packets dropped by kernel
```

Basic Troubleshooting

The following are basic steps for troubleshooting issues related to the PBIS agent.

Check the Version and Build Number

You can check the version and build number of the PBIS agent from computers that are running Linux, Unix, or Mac OS X, or from a computer that is connected to the domain controller and is running Windows.

Check From Linux, Unix, or Mac OS X

To check the version number of the PBIS agent from a computer running Linux, Unix, or Mac OS X, execute the following command:

cat /opt/pbis/data/VERSION

Another option is to execute the following command:

/opt/pbis/bin/get-status

On Linux distributions that support RPM—for example, Red Hat Enterprise Linux, Fedora, SUSE Linux Enterprise, OpenSUSE, and CentOS—you can determine the version and build number of the agent (6.5.0.xxxx in the examples below) by executing the following command at the shell prompt:

```
rpm -qa | grep pbis
```

The result shows the build version after the version number:

```
pbis-enterprise-6.5.0-2.62055.503
pbis-enterprise-gui-6.5.0-2.62055.503
```

On Unix computers and Linux distributions that do not support RPM, the command to check the build number varies by platform:

Platform	Command
Debian and Ubuntu	`dpkg -S /opt/pbis/`
Solaris	`pkginfo | grep -i pbis`
AIX	`lslpp -l | grep pbis`
HP-UX	`swlist | grep -i pbis`

Check From Windows

To check the version and build number of the PBIS agent from a Windows administration workstation that is connected to your domain controller:

1. In Active Directory Users and Computers, right-click the Linux, Unix, or Mac computer that you want, and then click **Properties**.
2. Click the **Operating System** tab. The build number is shown in the **Service pack** box.

Determine a Computer's FQDN

You can determine the fully qualified domain name of a computer running Linux, Unix, or Mac OS X by executing the following command at the shell prompt:

```
ping -c 1 `hostname`
```

On HP-UX

The command is different on HP-UX:

```
ping `hostname` -n 1
```

On Solaris

On Sun Solaris, you can find the FQDN by executing the following command (the computer's configuration can affect the results):

```
FQDN=`/ usr/lib/mail/ sh/ check-hostname|cut - d" " -
f7`;echo $FQDN
```

See Also

Join Active Directory Without Changing /etc/hosts

Make Sure Outbound Ports Are Open

If you are using local firewall settings, such as `iptables`, on a computer running the PBIS agent, make sure the following ports are open for outbound traffic.

Note: The PBIS agent is a client only; it does not listen on any ports.

Port	Protocol	Use
53	UDP/ TCP	DNS
88	UDP/TCP	Kerberos 5
123	UDP	NTP
389	UDP/TCP	LDAP
445	TCP	SMB over TCP
464	UDP/TCP	Computer password changes (typically after 30 days)
1433	TCP	Connection to SQL Server (Whatever port you are using for SQL must be open. The default port for SQL is 1433.)
3268	TCP	Global Catalog search

Tip: To view the firewall rules on a Linux computer using `iptables`, execute the following command:

```
iptables - nL
```

Check the File Permissions of nsswitch.conf

For PowerBroker Identity Services to work correctly, the `/etc/nsswitch.conf` file must be readable by user, group, and world. The following symptoms indicate that you should check the permissions of `nsswitch.conf`:

- Running the `id` command with an AD account as the argument (example: `id example.com\\kathy`) works when it is executed as root, but when the same command is executed by the AD user, it returns only a UID and GID without a name.
- Getting an "I have no name!" or "intruder alert" error message for non-root users.
- On HP-UX, running the `whoami` command with an AD user account returns “Intruder alert.”

Configure SSH After Upgrading It

After SSH is upgraded, run the following command as root to make sure that the `sshd_config` file is set up properly to work with PowerBroker Identity Services:

```
domainjoin-cli configure --enable ssh
```

Upgrading an Operating System

After upgrading an operating system or installing a kernel patch, you should rerun the domain-join command to make sure that the files related to the operating system, such as PAM and nsswitch, are configured properly to work with PowerBroker Identity Services. Re-executing the domain-join command also updates the `operatingSystemVersion` value and the `operatingSystemServicePack` value in Active Directory so the PBIS reporting tool reflects the correct version numbers.

Another suggestion, nearly universal in scope, is to apply updates to test systems before you apply updates to production systems, giving you the opportunity to identify and resolve potential issues before they can affect production machines.

Accounts

The following topics provide help with troubleshooting account issues.

Allow Access to Account Attributes

PBIS Enterprise is compatible with Small Business Server 2003. However, because the server locks down several user account values by default, you must create a group in Active Directory for your Unix computers, add each PBIS client computer to it, and configure the group to read all user information.

On other versions of Windows Server, the user account values are available by default. If, however, you use an AD security setting to lock them down, they will be unavailable to the PBIS agent.

To find Unix account information, the PBIS agent requires that the AD computer account for the machine running PBIS can access the attributes in the following table.

Attribute	Requirement
uid	Required when you use PBIS Enterprise in schema mode.

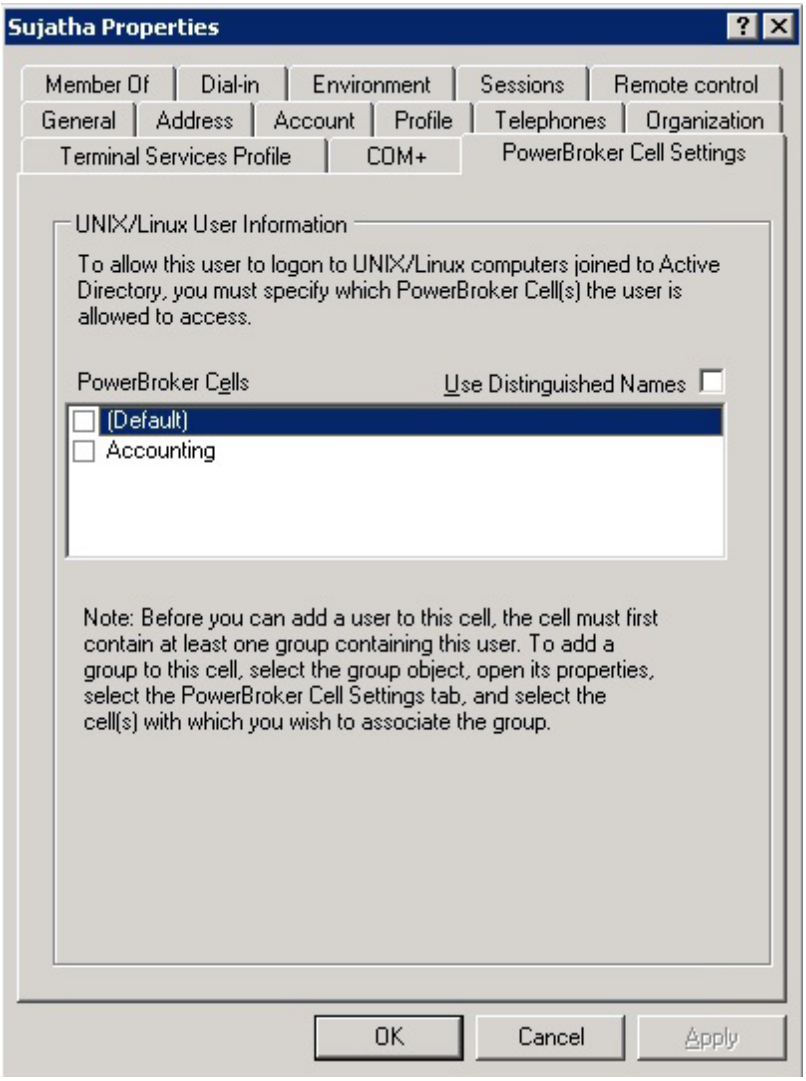
Attribute	Requirement
uidNumber	Required when you use PBIS Enterprise in schema mode.
gidNumber	Required when you use PBIS Enterprise in schema mode.
userAccountControl	Required for schema mode and non-schema mode. It is also required for unprovisioned mode, which means that you have not created a PowerBroker cell in Active Directory, as will be the case if you are using **PBIS Open**.

To allow access to account attributes:

1. In Active Directory Users and Computers, create a group named `Unix Computers`.
2. Add each PBIS client computer to the group.
3. In the console tree, right-click the domain, choose **Delegate Control**, click Next, click **Add**, and then enter the group named `Unix Computers`.
4. Click **Next**, select **Delegate the following common tasks**, and then in the list select **Read all user information**.
5. Click **Next**, and then click **Finish**.
6. On the target Unix, Linux, or Mac computer, restart the PBIS agent to reinitialize the computer account's logon to Active Directory and to get the new information about group membership.
7. Run `/opt/pbis/enum-users` to verify that you can read user information.

User Settings Are Not Displayed in ADUC

If there is no group in a cell that can serve as the user's primary GID—for instance, because the default primary group, domain users, has been removed from the cell—the **PBIS Settings** tab for a user in ADUC will not display the user or group settings, as shown in the screen shot below. To display the settings, enable a group that the user is a member of.



Resolve an AD Alias Conflict with a Local Account

When you use PowerBroker Identity Services to set an Active Directory alias for a user, the user can have a file-ownership conflict under the following conditions if the user logs on with the AD account:

- The AD alias is the same alias as the original local account name.
- The home directory assigned to the user in Active Directory is the same as the local user's home directory.
- The owner UID-GID of the AD account is different from that of the local account.

To avoid such conflicts, by default PBIS includes the short AD domain name in each user's home directory. If the conflict nevertheless occurs, there are two options to resolve it:

1. Make sure that the UID assigned to the user's AD alias is the same as that of the user's local account.
2. Log on as root and use the `chown` command to recursively change the ownership of the local account's resources to the AD user alias.

Change Ownership

Log on the computer as root and execute the following commands:

```
cd <users home directory root>
```

```
chown -R <AD user UID>:<AD primary group ID> *.*
```

Or: `chown -R <short domain name>\\<account name>:<short domain name>\\<AD group name> *.*`

Troubleshoot with the Get Status Command

The `/opt/pbis/bin/get-status` command shows whether the domain or the PBIS AD provider is offline. The results of the command include information useful for general troubleshooting.

/opt/pbis/bin/get-status

Here is an example of the information the command returns:

```
[root@rhel5d bin]# /opt/pbis/bin/get-status
LSA Server Status:
Compiled daemon version: 6.1.272.54796
Packaged product version: 6.1.272.54796
Uptime:        15 days 21 hours 24 minutes 1 seconds

[Authentication provider: lsa-activedirectory-provider]

        Status:        Online
        Mode:          Un-provisioned
        Domain:        EXAMPLE.COM
        Forest:        example.com
        Site:          Default-First-Site-Name
        Online check interval:  300 seconds
        [Trusted Domains: 1]

        [Domain: EXAMPLE]

                DNS Domain:       example.com
                Netbios name:     EXAMPLE
                Forest name:      example.com
                Trustee DNS name:
                Client site name: Default-First-Site-Name
                Domain SID:       S-1-5-21-3190566242-
1409930201-3490955248
                Domain GUID:      71c19eb5-1835-f345-
```

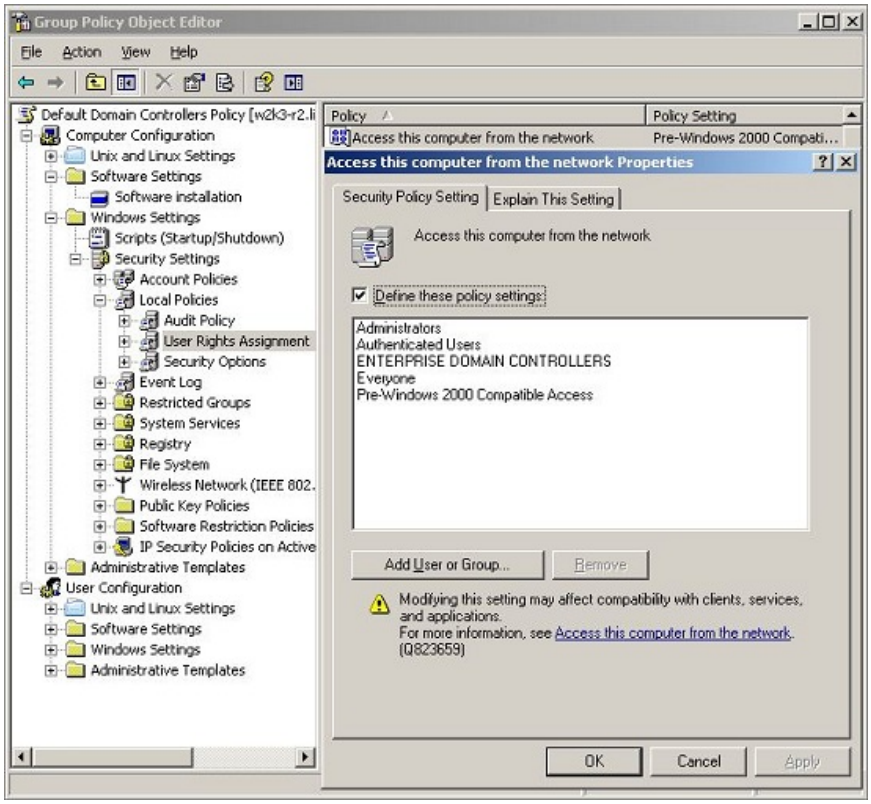
```
ba15-0595fb5b62e3
                Trust Flags:      [0x000d]
                                  [0x0001 - In forest]
                                  [0x0004 - Tree root]
                                  [0x0008 - Primary]
                Trust type:       Up Level
                Trust Attributes: [0x0000]
                Trust Direction:  Primary Domain
                Trust Mode:       In my forest Trust
(MFT)
                Domain flags:     [0x0001]
                                  [0x0001 - Primary]

                [Domain Controller (DC) Information]

                        DC Name:              w2k3-
r2.example.com
                        DC Address:
192.168.92.20
                        DC Site:              Default-
First-Site-Name
                        DC Flags:
[0x000003fd]
                        DC Is PDC:            yes
                        DC is time server:    yes
                        DC has writeable DS:  yes
                        DC is Global Catalog: yes
                        DC is running KDC:    yes

[Authentication provider: lsa-local-provider]

        Status:        Online
        Mode:          Local system
        Domain:        RHEL5D
```

Troubleshoot User Rights with Ldp.exe and Group Policy Modeling

The following Microsoft default domain policy and default domain controller policy can cause a PBIS client to fail to join a domain or to fail to enumerate trusts:

- Access this computer from the network. Users and computers that interact with remote domain controllers require the **Access this computer from the network** user right. Users, computers, and service accounts can lose the user right by being removed from a security group that has been granted the right. Removing the administrators group or the authenticated users group from the policy setting can cause domain join to fail. Microsoft says, "There is no valid reason for removing Enterprise Domain Controllers group from this user right."

- Deny access to this computer from the network. Including the domain computers group in the policy setting, for instance, causes domain-join to fail.



The symptoms of a user-right problem can include the following:

- An attempt to join the domain is unsuccessful.
- The PBIS authentication service, lsass, does not start.
- The `/opt/pbis/bin/get-status` command shows the domain or the AD provider as offline.

You can pin down the issue by using the ldp.exe tool to check whether you can access AD by using the machine account and machine password. Ldp.exe is typically included in the support tools (suptools.msi) for Windows and located on the Windows installation CD (Support folder, Tools subfolder). You might also be able to download the support tools that contain ldp.exe from the Microsoft website.

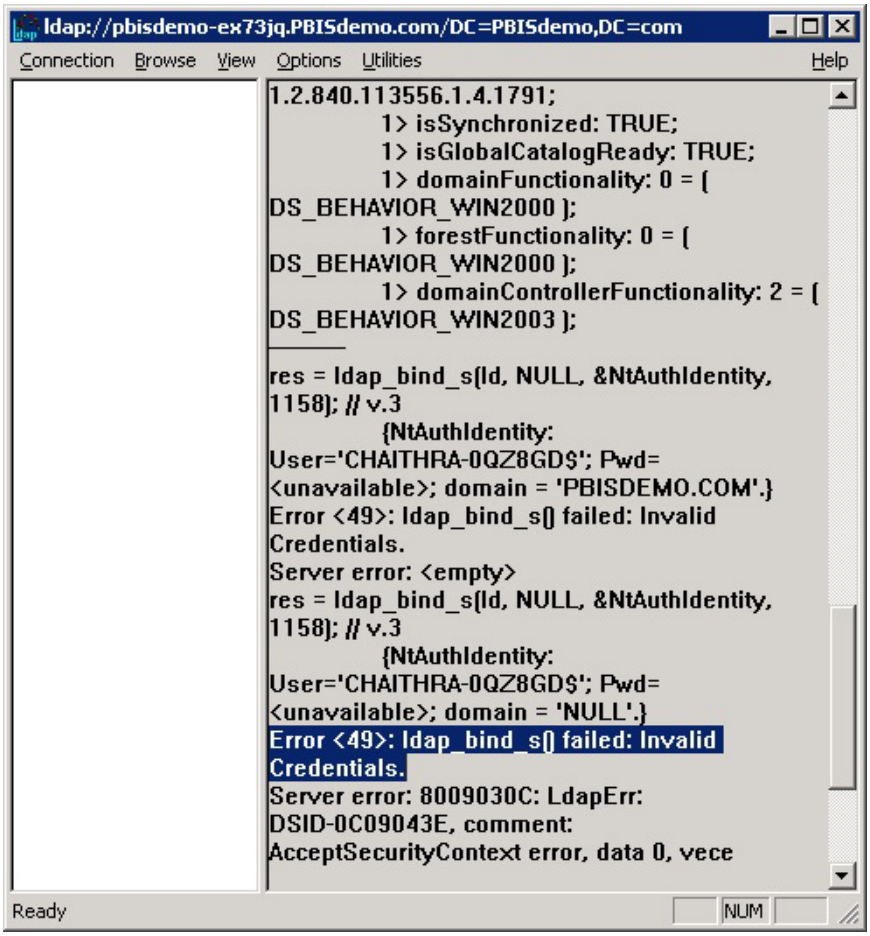
To resolve a user-right issue, you can use Group Policy Modeling in the Group Policy Management Console (GPMC) to find the offending policy setting and then modify it with the Group Policy Management Editor (or the Group Policy Object Editor).

1. On the PBIS client, run the `/opt/pbis/bin/lsa ad-get-machine password` command as root to get the machine password stored in Active Directory:

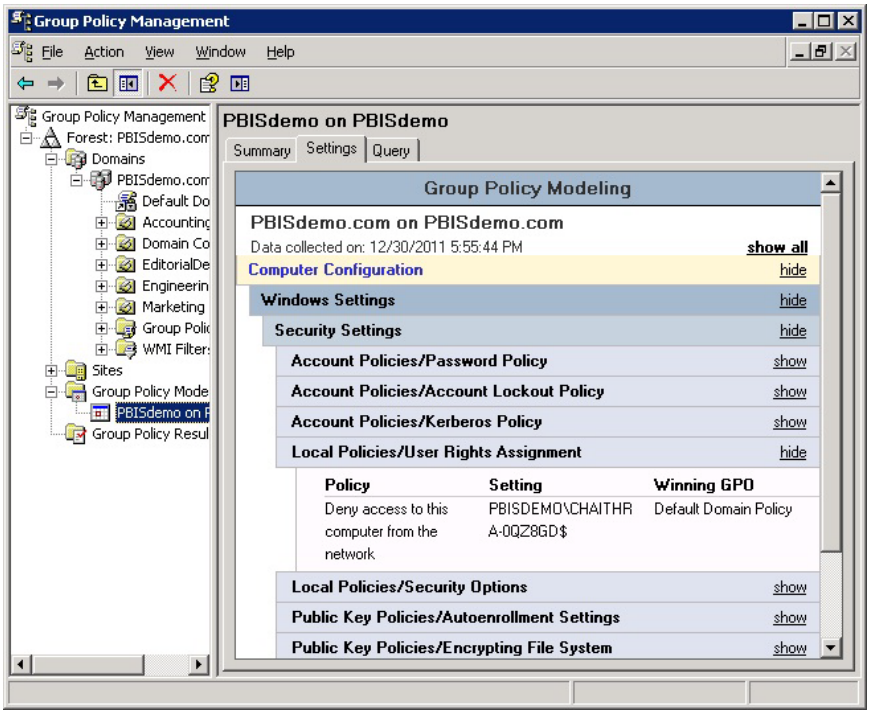
```
/opt/pbis/bin/lsa ad-get-machine password
Machine Password Info:
  DNS Domain Name: EXAMPLE.COM
  NetBIOS Domain Name: EXAMPLE
  Domain SID: S-1-5-21-3190566242-1409930201-3490955248
  SAM Account Name: RHEL5D$
  FQDN: rhel5d.example.com
  Join Type: 1
  Key Version: 0
  Last Change Time: 129401233790000000
  Password: i(2H2e41F7tHN275
```

2. On a Windows administrative workstation that can connect to AD, start ldp.exe and connect to the domain. (See the LDP UI article for more information.)

3. In LDP, on the **Connection** menu, click **Bind**, and then use the PBIS client's SAM account name and machine password from the output of the `lsa ad-get-machine password` command to bind to the directory.
   If the attempt to bind with the machine account and the machine password fails because of invalid credentials, as shown in the LDP output below, go to the GPMC and use Group Policy Modeling to try to identify the policy setting causing the problem.



4. In the GPMC, run Group Policy Modeling to pinpoint the offending policy setting and then modify the policy setting to grant the correct level of user right to the computer or user. For more information, see Group Policy Modeling.
   In the following screen shot, for example, the cause of the problem is that the **Deny access to this computer from the network** policy setting in the Default Domain Policy GPO contains the domain computers group.



Fix Selective Authentication in a Trusted Domain

When you turn on selective authentication for a trusted domain, PowerBroker Identity Services can fail to look up users in the trusted domain because the machine account is not allowed to authenticate with the domain controllers in the trusted domain. Here is how to grant the machine account access to the trusted domain:

1. In the domain the computer is joined to, create a global group and add the computer's machine account to the group.
2. In the trusted domain, in Active Directory Users and Computers, select the **Domain Controllers** container and open **Properties**.
3. On the **Security** tab, click **Advanced**, click **Add**, enter the global group, and then click **OK**.

4. In the **Permission Entry** box, under **Apply onto**, select **Computer objects**. Under **Permissions**, find **Allowed to Authenticate** and enable it. Click **OK** and then click **Apply** in the **Advanced Security Settings** box.
5. If you have already joined the PBIS client computer to the domain, restart the PBIS authentication service:
   `/opt/pbis/bin/lwsm restart lsass`

Cache

If a cache becomes corrupted or if certain conditions occur, you may need to clear caches.

Clear the Authentication Cache

There are certain conditions under which you might need to clear the cache so that a user's ID is recognized on a target computer.

By default, the user's ID is cached for 4 hours. If you change a user's UID for a PowerBroker cell with PBIS Enterprise, during the 4 hours after you change the UID you must clear the cache on a target computer in the cell before the user can log on. If you do not clear the cache after changing the UID, the computer will find the old UID until the cache expires.

There are three PBIS Group Policy settings that can affect the cache time:

- Cache Expiration Time, which stores UID-SID mappings, user/group enumeration lists, `getgrnam()` and `getpwnam()`, and so forth. Its default expiration time is 4 hours.
- ID Mapping Cache Expiration Time, which caches the mapping tables for SIDs, UIDs, and GIDs. Its default is 1 hour. This policy setting applies only to PBIS Enterprise 4.1 or earlier.
- ID Mapping Negative Cache Expiration Time, which stores failed SID-UID-GID lookups to prevent an overload of resolution requests. Its default is 5 minutes. This policy setting applies only to PBIS Enterprise 4.1 or earlier.

For more information about these policy settings, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Tip: While you are deploying and testing PBIS, set the cache expiration time of the PBIS agent's cache to a short period of time, such as 1 minute.

Clear the Cache on a Unix or Linux Computer

To delete all the users and groups from the PBIS AD provider cache on a Linux or Unix computer, execute the following command with superuser privileges:

```
/opt/pbis/bin/ad-cache --delete-all
```

You can also use the command to enumerate users in the cache, which may be helpful in troubleshooting. Here is an example:

```
[root@rhel5d bin]# ./ad-cache --enum-users
TotalNumUsersFound:      0
[root@rhel5d bin]# ssh example.com\\hab@localhost
Password:
Last login: Tue Aug 11 15:30:05 2009 from
rhel5d.example.com
[EXAMPLE\hab@rhel5d ~]$ exit
logout
Connection to localhost closed.
[root@rhel5d bin]# ./ad-cache --enum-users
User info (Level-0):
====================
Name:     EXAMPLE\hab
Uid:      593495196
Gid:      593494529
Gecos:    <null>Shell:    /bin/bash
Home dir: /home/EXAMPLE/hab
TotalNumUsersFound:      1
[root@rhel5d bin]#
```

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/ad-cache --help
```

Clear the Cache on a Mac OS X Computer

On a Mac OS X computer, clear the DirectoryService cache (not the PBIS cache) by running the following command with superuser privileges in Terminal:

dscacheutil -flushcache

Clear a Corrupted SQLite Cache

To clear the cache when PowerBroker Identity Services is caching credentials in its SQLite database and the entries in the cache are corrupted, use the following procedure for your type of operating system.

Clear the Cache on a Linux Computer

1. Stop the PBIS authentication service by executing the following command as root:
   `/opt/pbis/bin/lwsm lsass stop`

2. Clear the AD-provider cache and the local-provider cache by removing the following two files:
   `rm -f /var/lib/pbis/db/lsass-adcache.db`

   `rm -f /var/lib/pbis/db/lsass-local.db`

   Important: Do not delete the other `.db` files in the `/var/lib/pbis/db` directory.

3. Start the PBIS authentication service:
   `/opt/pbis/bin/lwsm lsass start`

Clear the Cache on a Mac

1. In Terminal, stop the PBIS authentication service by executing the following command as sudo:
   `/opt/pbis/bin/lwsm lsass stop`

2. Clear the AD-provider cache and the local-provider cache by removing the following two files:
   `sudo rm -f /var/lib/pbis/db/lsass-adcache.db`

   `sudo rm -f /var/lib/pbis/db/lsass-local.db`

   Important: Do not delete the other `.db` files in the `/var/lib/pbis/db` directory.

3. Restart the PBIS authentication service:
   `/opt/pbis/bin/lwsm lsass start`

Clear the Cache on a Unix Computer

1. Stop the PBIS authentication service by executing the following command as root:
   `/opt/pbis/bin/lwsm stop lsass`

2. Clear the AD-provider cache and the local-provider cache by removing the following two files:
   `rm -f /var/lib/pbis/db/lsass-adcache.db`

   `rm -f /var/lib/pbis/db/lsass-local.db`

   Important: Do not delete the other `.db` files in the `/var/lib/pbis/db` directory.

3. Start the PBIS authentication service:
   `/opt/pbis/bin/lwsm start lsass`

Kerberos

The following resources can help you troubleshoot time synchronization and other Kerberos issues:

- Kerberos Authentication Tools and Settings:
  http://technet.microsoft.com/en-us/library/cc738673(WS.10).aspx
- Authentication Errors Caused by Unsynchronized Clocks:
  http://technet.microsoft.com/en-us/library/cc780011(WS.10).aspx
- Kerberos Technical Supplement for Windows:
  http://msdn2.microsoft.com/en-us/library/aa480609.aspx
- The Kerberos Network Authentication Service (V5) RFC:
  http://www.ietf.org/rfc/rfc4120.txt
- Troubleshooting Windows Server Issues (including Kerberos errors):
  http://technet.microsoft.com/en-us/windowsserver/default.aspx
- Kerberos and LDAP Troubleshooting Tips:
  http://ww-
  w.m-
  icrosoft.com/technet/solutionaccelerators/cits/interopmigration/unix/usecdirw/17wsc

The following topics can help you address common issues related to Kerberos and PowerBroker Identity Services.

Fix a Key Table Entry-Ticket Mismatch

When an AD computer account password changes two or more times during the lifetime of a domain user's credentials, the computer's entry that matches the Kerberos service ticket is dropped from the Kerberos key table. Even though the service ticket has not expired, an action that depends on the entry, such as reading the event log or using single sign-on, will fail.

To avoid issues with Kerberos key tables, keytabs, and single sign-on, the computer password expiration time must be at least twice the maximum lifetime for user tickets, plus a little more time to account for the permitted clock skew.

The expiration time for a user ticket is set by using an Active Directory Group Policy setting called Maximum lifetime for user ticket. The default user ticket lifetime is 10 hours; the default PBIS computer password lifetime is 30 days.

Causes

The computer account password can change more frequently than the user's AD credentials under the following conditions:

- Joining a domain two or more times.
- Setting the expiration time of the computer account password Group Policy setting to be less than twice the maximum lifetime of user tickets. For more information, see the *PowerBroker Identity Services Group Policy Administration Guide*.
- Setting the local `machine-password-lifespan` for the lsass service in the PBIS registry to be less than twice the maximum lifetime for user tickets.

Solution

If a computer's entry is dropped from the Kerberos key table, you must remove the unexpired service tickets from the user's credentials cache by reinitializing the cache. Here is how:

On Linux and Unix, reinitialize the credentials cache by executing the following command with the account of the user who is having the problem:

`/opt/pbis/bin/kinit`

On Mac, you must run both the native kinit command and the PBIS kinit command with the account of the user who is having the problem. You must run both commands because the native ssh client uses the native credentials cache while the PBIS processes, such as those that access the event log, use the MIT credentials cache:

```
/opt/pbis/bin/kinit
kinit
```

Fix a KRB Error During SSO

When you are working in a network with a disjoint namespace in which the Active Directory domain name is different from the DNS domain suffix for computers, you may need to modify the domain_realm section of `/etc/krb5.conf` on your target computer even though your DNS A and PTR records are correct for both DNS domains and can be found both ways.

The following error, in particular, indicates that you might have to modify your `krb5.conf` file before single sign-on (with SSH, for example) will work:

```
KRB ERROR BAD OPTION
```

Assume your computer's Active Directory domain is `bluesky.example.com`and your computer's FQDN is `somehostname.green.example.com` and you have already created the following entries in DNS:

```
_kerberos._tcp.green.example.com 0 100 389
ad2.bluesky.example.com
_kerberos._udp.green.example.com 0 100 389
ad2.bluesky.example.com
```

Meantime, on the target computer, the `[domain_realm]` entry of your `/etc/krb5.conf` file looks like this:

```
[domain_realm]
.bluesky.example.com = BLUESKY.EXAMPLE.COM
bluesky.example.com = BLUESKY.EXAMPLE.COM
```

To resolve the error, add the following two lines to the `[domain_realm]` entry of your `/etc/krb5.conf` file:

```
.green.example.com = BLUESKY.EXAMPLE.COM
green.example.com = BLUESKY.EXAMPLE.COM
```

After adding the two lines above, the complete `[domain_realm]` entry now looks like this:

```
[domain_realm]
.bluesky.example.com = BLUESKY.EXAMPLE.COM
bluesky.example.com = BLUESKY.EXAMPLE.COM
.green.example.com = BLUESKY.EXAMPLE.COM
green.example.com = BLUESKY.EXAMPLE.COM
```

Finally, make sure that you have a correct `.k5login` file and then try to log on again.

Eliminate Logon Delays When DNS Connectivity Is Poor

If connectivity to your DNS servers is tenuous or becomes unavailable, name resolution can time out, delaying the logon process. Because Active Directory is heavily dependent on a well-functioning DNS system, you should work to resolve your DNS issues.

If you cannot fix your DNS system, however, you can as a last resort set up a caching-forwarding name server on the PBIS client to eliminate the logon delay. For instance, you can set up a BIND server on each Linux or Unix computer on which you are running PBIS. Then you can configure BIND as a local caching resolver and add your nameserver addresses to the forwarder list, leaving `/etc/resolv.conf` with only the local loopback address:

```
search example.com
nameserver 127.0.0.1
```

For instructions on how to set up BIND, see the BIND documentation.

Eliminate Kerberos Ticket Renewal Dialog

There is an applet called `krb5-auth-dialog` that by default is active on many Linux distributions. It is intended to assist you with renewing your Kerberos tickets before they expire. Because PowerBroker Identity Services renews your tickets for you, the dialog is superfluous and can be a nuisance.

To disable the dialog:

1. In the menu, click **System, Preferences, More Preferences, Session**.
2. Click the **Startup Programs** tab and disable the **krb5-auth-dialog** program. This change prevents it from restarting next time you log on.
3. Close the **Sessions** window and then run this command from the shell: `pkill krb5-auth-dialog`

PAM

For instructions on how to generate a PAM debug log, see Generate a PAM Debug Log.

Dismiss the Network Credentials Required Message

After leaving the screen saver on a Gnome desktop that is running the Gnome Display Manager, or GDM, you might see a pop-up notification saying that network authentication is required or that network credentials are required. You can ignore the notification. The GDM process that tracks the expiration time of a Kerberos TGT might not recognize the updated expiration time of a Kerberos TGT after it is refreshed by PowerBroker Identity Services.

OS-Specific Troubleshooting

The following topics provide PBIS agent troubleshooting guidance that is unique to individual operating systems.

Red Hat and CentOS

The following procedures may help resolve issues with the PBIS agent on computers running the Red Hat or CentOS operating systems.

Modify PAM to Handle UIDs Less Than 500

By default, the configuration file for PAM system authentication—`/etc/pam.d/system-auth`—on Red Hat Enterprise Linux 5 and CentOS 5 contains the following line, which blocks a user with a UID value less than or equal to 500 from logging on to a computer running the PBIS agent. The symptom is a login failure with a never-ending password prompt.

```
auth  requisite  pam_succeed_if.so uid >= 500 quiet
```

Solution: Either delete the line from `/etc/pam.d/system-auth` or modify it to allow users with UIDs lower than 500:

```
auth  requisite  pam_succeed_if.so uid >= 50 quiet
```

Ensure That the Correct Version of the coreutils RPM Is Installed

Some patch levels of the coreutils RPM on Red Hat Enterprise Linux 3 have a version of the `id` command that does not return complete group membership information when the command is run with the id username syntax. The command returns only the UID and primary GID for a user. Secondary groups may be omitted.

On Linux, there are four commands to get group memberships:

- Call `getgroups`. It returns the primary and secondary GIDs of the current process. The `id` command calls this when a username is not passed.
- Call `initgroups` followed by `getgroups`. The `initgroups` call will query nsswitch for the users primary and secondary groups. The result is stored in the process, which is then returned by `getgroups`. You need root access to call `initgroups`, so id sometimes does this when you run the command as root.
- Call `getgrouplist`. This function calls nsswitch to return the group list of a user, and it does not require root access. Unfortunately this function was added in glibc 2.2.4, and the `id` command started using it after that.
- Call `getgrent` to enumerate all the groups on the system, and search for the user in the `gr_mem` field.

On RHEL 3, `id` from coreutils version 4.5.3-28.4 can use the `getgrouplist` function, but that code was removed in 4.5.3-28.7. To check your coreutils version for glibc:

```
rpm -q coreutils glibc coreutils-4.5.3-28.7 glibc-2.3.2-
   95.50
```

Without the `getgrouplist` function, the `id` command falls back on using `getgrent` to get the groups. By default, PBIS returns abbreviated results when enumerating all groups, so `id` does not find the user in the member's field. You could turn on full group enumeration, but then the `id` command would download the group membership of everyone in Active Directory just to obtain the results for one user.

Here is an example.

1. Check the user.

```
[root@example-03293b root]# su - corpqa\\user0001
[CORPQA\user0001@example-03293b user0001]$ id
   CORPQA\\user0002
uid=105559(CORPQA\user0002)
   gid=1661993473(CORPQA\domain^users)
   groups=1661993473(CORPQA\domain^users)
[CORPQA\user0001@example-03293b user0001]$ logout
```

2. Turn on full group enumeration locally by using `config`.

```
[root@example-03293b root]# /opt/pbis/bin/config
   NssGroupMembersQueryCacheOnly false
[root@example-03293b root]# /opt/pbis/bin/config
   NssEnumerationEnabled true
```

3. Check the user again:

```
[root@example-03293b root]# su - corpqa\\user0001
[CORPQA\user0001@example-03293b user0001]$ id
   CORPQA\\user0002
uid=105559(CORPQA\user0002)
   gid=1661993473(CORPQA\domain^users)
groups=1661993-
   473(CORPQA\domain^users),1662020290(CORPQA\enabled),
16620202-
   91(CORPQA\enabledparent),100395(CORPQA\innergroup1),
100401(CORPQA\loopgroup),100394(CORPQA\outergroup),
100381(COR-
   PQA\usergroup0001),100382(CORPQA\usergroup0002),
1662002383-
   (CORPQA\use-
   rgroup0009),1662002420(CORPQA\usergroup0047),
1662003573(CORPQA\usergroup0200)
```

Even with NSS settings enabled, the `id` command does a case-sensitive search even though PBIS does not treat the usernames as case sensitive. Therefore, if you use the non-canonical case, the groups are not displayed.

To fix the output of the `id` command on RHEL 3 computers where this problem occurs, ensure that the correct version of the coreutils RPM is installed.

Ubuntu

Try the following to resolve issues with the PBIS agent on computers running Ubuntu.

su segfaults

On 32-bit versions of Ubuntu 10.10 running PBIS, `su` might segfault. Upgrading to Ubuntu 10.2 or later resolves the issue.

SUSE Linux Enterprise Desktop (SLED)

SUSE Linux Enterprise Desktop 11 (SLED 11) includes PBIS Enterprise.

Home Directory on SLED 11

When a user gains access to SLED 11 through Nomad—a remote desktop using RDP protocol with session management—the default home directory specified in `/lib/security/pam_lsass.so` is ignored. To correct the issue, change `/etc/pam.d/xrdp-sesman` to include the following line:

```
session sufficient /lib/security/pam_lsass.so
```

Updating PAM on SLED 11

Novell has issued a PAM update (pam-config-0.68-1.22) for SLED 11 that modifies the common-session-pc file to include the following entry:

```
session optional pam_gnome_keyring.so auto_start_if=gdm
```

Because the PAM update makes a backup of the file and replaces it with the modified version, the changes that PBIS had made to the file are no longer present, which blocks new AD users from logging on. The following error messages may appear:

```
Could not update ICEauthority file
/home/john/.ICEauthority
There is a problem with the configuration server.
(/user/lib/gconf/2/gconf-sanity-check-2 exited with
status 256)
```

Solution: After you update PAM, run the following command as root:

```
/opt/pbis/bin/domainjoin-cli configure --enable pam
```

Or, you can make the changes manually: Open the backed up version of the common-session-pc file, add the following line to it, and then use it to overwrite the new version of the common-session-pc file:

```
session optional        pam_gnome_keyring.so    auto_
start_if=gdm
```

AIX

Try the following to resolve issues with the PBIS agent on computers running AIX.

Increase Max Username Length on AIX

By default, AIX is not configured to support long user and group names, which might present a conflict when you try to log on with a long Active Directory username. On AIX 5.3 and AIX 6.1, the symptom is that group names, when enumerated through the `groups` command, are truncated.

To increase the max username length on AIX 5.3, use the following syntax:

```
# chdev -l sys0 -a max_logname=MaxUserNameLength+1
```

Example:

```
# chdev -l sys0 -a max_logname=255
```

This command allocates 254 characters for the user and 1 for the terminating null.

The safest value to which you can set `max_logname` is `255`.

You must reboot for the changes to take effect:

```
# shutdown -Fr
```

Note: AIX 5.2 does not support increasing the maximum user name length.

Updating AIX

When you update AIX, the authentication of users, groups, and computers might fail because the AIX upgrade process overwrites changes that PowerBroker Identity Services makes to system files. Specifically, upgrading AIX to version 6.1tl3 overwrites `/lib/security/methods.cfg`, so you must manually add the following code to the last lines of the file after you finish upgrading:

```
LSASS:
   program = /usr/lib/security/LSASS
```

FreeBSD

Try the following to resolve issues with the PBIS agent on computers running FreeBSD.

Keep Usernames to 16 Characters or Less

On FreeBSD, user names that are longer than 16 characters, including the domain name, exceed the FreeBSD username length limit. Attempts to connect by ssh, for example, to a FreeBSD computer with a user name that exceeds the limit can result in the following notification:

```
bvt-fbs72-64# ssh testuser1@localhost
Password:
Connection to localhost closed by remote host.
Connection to localhost closed.
```

The log for sshd, meanwhile, might show an error that looks something like this:

```
Oct  7 18:22:57 vermont02 sshd[66387]:
setlogin(EXAMPLE\adm.kathy):
Invalid argument
Oct  7 18:25:02 vermont02 sshd[66521]:
setlogin(EXAMPLE\adm.kathy):
Invalid argument
```

Although `testuser1` is less than 16 characters, when you use the `id` command to check the account, something longer than 16 characters is returned:

```
[root@bvt-fbs72-64 /home/testuser]# id testuser1
uid=1100(BVT-FBS72-64\testuser1) gid=1801(BVT-FBS72-
64\testgrp)
groups=1801(BVT-FBS72-64\testgrp)
```

The result of the `id` command exceeds the FreeBSD username length limit.

There are several solutions: set the default domain, change the user name to 16 characters or less, or with PBIS Enterprise use aliases. Keep in mind, though, that aliases will not solve the problem in relation to the PBIS local provider.

Solaris

Try the following to resolve issues with the PBIS agent on computers running Solaris.

Turn On Core Dumps on Solaris 10

If you are investigating a process that is crashing on Solaris 10 or Solaris Sparc 10, but a core dump is not being generated, it's probably because per-process core dumps are turned off. You can use the `coreadm` command to manage the core dumps. The settings are saved in the `/etc/coreadm.conf` file.

A configuration for core dumps with the per-process option turned off looks like this:

```
# coreadm
     global core file pattern:
     global core file content: default
       init core file pattern: core
       init core file content: default
            global core dumps: disabled
       per-process core dumps: disabled
      global setid core dumps: disabled
 per-process setid core dumps: disabled
     global core dump logging: disabled
```

You'll need per-process core dumps, though, to troubleshoot a process that is terminating unexpectedly. To turn on core dumps for a process, execute the following command as root:

coreadm -e process

For more information, see Core Dump Management on the Solaris OS and the man page for `coreadm`.

Mac OS X

Try the following to resolve issues with the PBIS agent on computers running Mac OS X.

Find the PBIS Service Manager Daemon on a Mac

To locate the PBIS service manager process on a Mac OS X computer, execute the following command in Terminal:

sudo launchctl list | grep pbis

On a Mac computer, the name of the daemon for the service manager is as follows:

```
com.likewisesoftware.lwsmd
```

Remove Dock Items by Using Workgroup Manager

If you have integrated PBIS Enterprise with Apple's Workgroup Manager by following the instructions in this guide, you can remove dock items by using an MCX policy setting. For instructions, see Apple's support page on Managed Client: Items removed in Workgroup Manager remain in a user's Dock.

Command-Line Reference

This chapter provides an overview of the commands in `/opt/pbis/bin`. Most of the commands are intended to be run as root.

Additional troubleshooting information, some of which involves command-line utilities, is provided in Troubleshooting the PBIS Agent. Commands for managing the event log are covered in Monitoring Events with the Event Log.

For information about troubleshooting the Group Policy commands for PBIS Enterprise, see the *PowerBroker Identity Services Group Policy Administration Guide*.

For an overview of commands such as `rpm` and `dpkg` that can help you manage PBIS on Linux and Unix platforms, see *Package Management Commands*.

Manage PBIS Services (lwsm)

The PBIS Service Manager lets you track and troubleshoot all the PBIS services with a single command-line utility. You can, for instance, check the status of the services and start or stop them. The service manager is the preferred method for restarting a service because it automatically identifies a service's dependencies and restarts them in the right order. In addition, you can use the service manager to set the logging destination and the log level.

To list the status of the services, run the following command with superuser privileges at the command line:

/opt/pbis/bin/lwsm list

Example:

```
[root@rhel5d bin]# /opt/pbis/bin/lwsm list
lwreg       running (standalone: 1920)
dcerpc      running (standalone: 2544)
eventlog    running (standalone: 2589)
lsass       running (standalone: 2202)
lwio        running (standalone: 2191)
netlogon    running (standalone: 2181)
rdr         running (io: 2191)
```

To restart the `lsass` service, run the following command with superuser privileges:

/opt/pbis/bin/lwsm restart lsass

After you change a setting in the registry, you must use the service manager to force the service to begin using the new configuration by executing the following command with super-user privileges. This example refreshes the `lsass` service:

/opt/pbis/bin/lwsm refresh lsass

To view information about the `lsass` service, including its dependencies, run the following command:

/opt/pbis/bin/lwsm info lsass

Example:

```
[root@rhel5d bin] # /opt/pbis/bin/lwsm info lsass
   Service: lsass
   Description: Security and Authentication Subsystem
   Type: module
   Autostart: yes
   Path: /opt/pbis/lib/lw-svcm/lsass.so
   Arguments:
   Environment:
   Dependencies: netlogon lwio lwreg rdr
   Service Group: lsass
   File descriptor limit: 1024
   Core dump size limit: inherit
```

To view all the service manager's commands and arguments, run the following command:

/opt/pbis/bin/lwsm --help

Modify Settings (config)

To quickly change an end-user setting in the registry for the PBIS agent, you can run the `config` command-line tool as root:

/opt/pbis/bin/config

For more information, see Modify Settings with the config Tool.

Start the Registry Shell (regshell)

You can access and modify the PBIS registry by using the registry shell—`regshell`. The shell works in a way that is similar to BASH. You can view a list of the commands that you can execute in the shell by entering `help`:

```
/opt/pbis/bin/regshell
\> help
```

You can also manage the registry by executing the registry's commands from the command line. For more information, see Configuring PBIS with Registry Settings.

Export the Registry to an Editor (edit-reg)

Executing the following command exports the contents of the PBIS registry to the editor specified by your `EDITOR` environment variable. You can use the `edit-reg` command to quickly view the contents of the registry and make changes to the settings. Then, you can launch the registry shell and import the modified file so that your changes take effect.

/opt/pbis/bin/edit-reg

If you have not set a default editor, the script searches for an available editor in the following order: gedit, vi, friends, emacs. On platforms without gedit, an error may occur. You can correct the error by setting the `EDITOR` environment variable to an available editor, such as vi:

```
export EDITOR=vi
```

Set the Log Level (set-log-level)

You can set the PBIS log level for the PBIS authentication service by executing the following command and replacing `level` with one of the available logging levels: error, warning, info, verbose, debug, trace.

/opt/pbis/bin/set-log-level level

Example: `/opt/pbis/bin/set-log-level debug`

The log level is changed only until the authentication service (lsass) or the computer restarts. Syslog messages are logged through the daemon facility. The default setting is `error`.

Change the Hostname in the Local Provider (set-machine-name)

After you change the hostname of a computer, you must also change the name in the PBIS local provider database so that the local PBIS accounts use the correct prefix. To do so, execute the following command as root, replacing `hostName` with the name that you want:

/opt/pbis/bin/set-machine-name hostName

Find a User or a Group

On a Unix or Linux computer that is joined to an Active Directory domain, you can check a domain user's or group's information by either name or ID. These commands can verify that the client can locate the user or group in Active Directory.

Find a User by Name

Execute the following command, replacing `domain\\username` with the full domain user name or the single domain user name of the user that you want to check:

/opt/pbis/bin/find-user-by-name domain\\username

Example: `/opt/pbis/bin/find-user-by-name mydomain\\trejo`

You can optionally specify the level of detail of information that is returned. Example:

```
/opt/pbis/bin/find-user-by-name --level 2 mydomain\\trejo
User info (Level-2):
====================
Name:                           trejo
SID:                            S-1-5-21-3447809367-
3151979076-456401374-1135
UPN:                            trejo@MYDOMAIN.EXAMPLE.COM
Generated UPN:                  NO
DN:
CN=trejo,CN=Users,DC=MYDOMAIN,DC=EXAMPLE,DC=COM
Uid:                            239600751
Gid:                            239600770
Gecos:                          Markus Trejo
Shell:                          /bin/sh
Home dir:                       /home/MYDOMAIN/trejo-
macbook/trejo-bvt
LMHash length:                  0
NTHash length:                  0
Local User:                     NO
Account disabled (or locked): FALSE
Account expired:                FALSE
Password never expires:         TRUE
Password Expired:               FALSE
Prompt for password change:     YES
User can change password:       YES
Days till password expires:     0
Logon restriction:              NO
trejo-macbook:~ root#
```

For more information, execute the following command:

`/opt/pbis/bin/find-user-by-name --help`

Find a User by UID

To find a user by UID, execute the following command, replacing UID with the user's ID:

/opt/pbis/bin/find-user-by-idUID

Example:

```
/opt/pbis/bin/find-user-by-id 593495196
```

Find a User by SID

On a Linux, Unix, or Mac OS X computer that is joined to a domain, you can find a user in Active Directory by his or her security identifier (SID). To find a user by SID, execute the following command as root, replacing SID with the user's security identifier:

/opt/pbis/bin/find-by-sid SID

Example:

```
[root@rhel4d bin]# /opt/pbis/bin/find-by-sid S-1-5-21-
382349973-3885793314-468868962-1180
User info (Level-0):
====================
Name:      EXAMPLE\hab
SID:      S-1-5-21-382349973-3885793314-468868962-1180
Uid:      593495196
Gid:      593494529
Gecos:     Jurgen Habermas
Shell:    /bin/ sh
Home dir: /home/ EXAMPLE/ hab
```

Tip: To view the command's options, type the following command:

```
/opt/pbis/bin/find-by-sid --help
```

Find a Group by Name

/opt/pbis/bin/find-group-by-name domain\\username

Example:

```
/opt/pbis/bin/find-group-by-name example.com\\dnsadmins
```

Find a Group by ID

/opt/pbis/bin/find-group-by-id GID

Example:

```
[root@rhel4d bin]# /opt/pbis/bin/find-group-by-id
593494534
Group info (Level-0):
====================
Name:     EXAMPLE\schema^admins
Gid:      593494534
SID:      S-1-5-21-382349973-3885793314-468868962-518
```

Tip: To view this command's options, type the following command:

`/opt/pbis/bin/find-group-by-id --help`

List Groups for a User (list-groups-for-user)

To find the groups that a user is a member of, execute the following command followed by either the user's name or UID:

`/opt/pbis/bin/list-groups-for-user`

Example: `/opt/pbis/bin/list-groups-for-user 593495196`

Here is the command and its result for the user `example\\hab`:

```
[root@rhel5d bin]# ./list-groups-for-user example\\hab
Number of groups found for user 'example\hab' : 2
Group[1 of 2] name = EXAMPLE\enterprise^admins (gid =
593494535)
Group[2 of 2] name = EXAMPLE\domain^users (gid =
593494529)
```

Tip: To view this command's options, type the following command:

`/opt/pbis/bin/list-groups-for-user --help`

List Groups (enum-groups)

On a Linux, Unix, or Mac OS X computer that is joined to a domain, you can enumerate the groups in Active Directory and view their members, GIDs, and SIDs:

/opt/pbis/bin/enum-groups --level 1

The PBIS agent enumerates groups in the primary domain. Groups in trusted domains and linked cells are not enumerated. NSS membership settings in the registry do not affect the result of the command.

Tip: To view the command's options, type the following command:

`/opt/pbis/bin/enum-groups --help`

List Users (enum-users)

On a Linux, Unix, or Mac OS X computer that is joined to a domain, you can enumerate the users in Active Directory and view their members, GIDs, and SIDs:

/opt/pbis/bin/enum-users

The PBIS agent enumerates users in the primary domain. Users in trusted domains and linked cells are not enumerated. NSS membership settings in the registry do not affect the result of the command.

Tip: To view the command's options, type the following command:

```
/opt/pbis/bin/enum-users --help
```

To view full information about the users, include the `level` option when you execute the command:

```
/opt/pbis/bin/enum-users --level 2
```

Example result for a one-user batch:

```
User info (Level-2):
====================
Name:                        EXAMPLE\sduval
UPN:                         SDUVAL@EXAMPLE.COM
Generated UPN:               NO
Uid:                         593495151
Gid:                         593494529
Gecos:                       Shelley Duval
Shell:                       /bin/sh
Home dir:                    /home/EXAMPLE/sduval
LMHash length:               0
NTHash length:               0
Local User:                  NO
Account disabled:            FALSE
Account Expired:             FALSE
Account Locked:              FALSE
Password never expires:      FALSE
Password Expired:            FALSE
Prompt for password change:  NO
```

List the Status of Authentication Providers (get-status)

PowerBroker Identity Services includes two authentication providers:

1. A local provider
2. An Active Directory provider

If the AD provider is offline, you will be unable to log on with your AD credentials. To check the status of the authentication providers, execute the following command as root:

/opt/pbis/bin/get-status

A healthy result should look like this:

```
LSA Server Status:
Agent version: 5.4.0
Uptime:        22 days 21 hours 16 minutes 29 seconds
[Authentication provider: lsa-local-provider]
        Status:   Online
        Mode:     Local system
[Authentication provider: lsa-activedirectory-provider]
        Status:   Online
        Mode:     Un-provisioned
        Domain:   example.com
        Forest:   example.com
        Site:     Default-First-Site-Name
```

An unhealthy result will not include the AD authentication provider or will indicate that it is offline. If the AD authentication provider is not listed in the results, restart the authentication service.

If the result looks like the line below, check the status of the PBIS services to make sure they are running.

```
Failed to query status from LSA service.  The LSASS server
is not responding.
```

To check the status of the services, run the following command as root:

```
/opt/pbis/bin/lwsm list
```

List the Domain

This command retrieves the Active Directory domain to which the computer is connected. The command's location is as follows:

/opt/pbis/bin/lsa ad-get-machine account

List Domain Controllers (get-dc-list)

This command lists the domain controllers for a target domain. You can delimit the list in several ways, including by site. The command's location is as follows:

/opt/pbis/bin/get-dc-list

Example usage:

```
[root@rhel5d bin]# ./get-dc-list example.com
Got 1 DCs:
===========
DC 1: Name = 'steveh-dc.example.com', Address =
'192.168.100.132'
```

To view the command's syntax and arguments, execute the following command:

`/opt/pbis/bin/get-dc-list --help`

List Domain Controller Information (get-dc-name)

This command displays the name of the current domain controller for the domain you specify. The command can help you select a domain controller. The command's location is as follows:

/opt/pbis/bin/get-dc-name DomainName

To select a domain controller, run the following command as root until the domain controller you want is displayed. Replace `DomainName` with the name of your domain:

`/opt/pbis/bin/get-dc-name DomainName --force`

List Domain Controller Time (get-dc-time)

This command displays the time of the current domain controller for the domain that you specify. The command can help you determine whether there is a Kerberos time-skew error between a PBIS client and a domain controller. The command's location is as follows:

/opt/pbis/bin/get-dc-time

Example:

```
[root@rhel5d bin]# ./get-dc-time example.com
DC TIME: 2009-09-08 14:54:18 PDT
```

List Computer Account Information (lsa ad-get-machine)

You can print out the computer account name, computer account password, SID, and other information by running the following command as root.

/opt/pbis/bin/lsa ad-get-machine account domainDNSName

Example: `/opt/pbis/bin/lsa ad-get-machine account example.com`

Dynamically Update DNS (update-dns)

This command registers an IP address for the computer in DNS. The command is useful when you want to register A and PTR records for your computer and the DHCP server is not registering them.

/opt/pbis/bin/update-dns

Here is an example of how to use it to register an IP address:

```
/opt/pbis/bin/update-dns --ipaddress 192.168.100.4 --fqdn
corp.example.com
```

If your system has multiple NICs and you are trying to register all their IP addresses in DNS, run the command once with multiple instances of the `ipaddress` option:

```
 /opt/pbis/bin/update-dns --fqdn corp.example.com --
ipaddress 192.168.100.4 --ipaddress 192.168.100.7 --
ipaddress 192.168.100.9
```

To troubleshoot, you can add the `loglevel` option with the `debug` parameter to the command:

```
 /opt/pbis/bin/update-dns --loglevel debug --fqdn
corp.example.com --ipaddress 192.168.100.4 --ipaddress
192.168.100.7
```

For more information on the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/update-dns --help
```

Manage the AD Cache (ad-cache)

This command manages the PBIS cache for Active Directory users and groups on Linux and Unix computers. The command's location is as follows:

/opt/pbis/bin/ad-cache

You can use the command to clear the cache. The command's arguments can delete from the cache a user, a group, or all users and groups. The following example demonstrates how to delete all the users and groups from the cache:

```
/opt/pbis/bin/ad-cache --delete-all
```

Tip: To reclaim disk space from SQLite after you clear the cache when you are using the non-default SQLite caching option, execute the following command as root, replacing `fqdn` with your fully qualified domain name:

```
/opt/pbis/bin/sqlite3 /var/lib/pbis/db/lsass-adcache.db.fqdn
vacuum
```

You can also use the `ad-cache` command to enumerate users in the cache, which may be helpful in troubleshooting. Example:

```
[root@rhel5d bin]# ./ad-cache --enum-users
TotalNumUsersFound:      0
[root@rhel5d bin]# ssh example.com\\hab@localhost
Password:
Last login: Tue Aug 11 15:30:05 2009 from
rhel5d.example.com
[EXAMPLE\hab@rhel5d ~]$ exit
logout
Connection to localhost closed.
[root@rhel5d bin]# ./ad-cache --enum-users
User info (Level-0):
====================
Name:     EXAMPLE\hab
Uid:      593495196
Gid:      593494529
Gecos:    <null>Shell:    /bin/bash
Home dir: /home/EXAMPLE/hab
TotalNumUsersFound:      1
[root@rhel5d bin]#
```

To view all the command's syntax and arguments, execute the following command:

`/opt/pbis/bin/ad-cache --help`

On Mac OS X

On a Mac OS X computer, clear the clear the DirectoryService cache (not the PBIS cache) by running the following command with superuser privileges in Terminal:

dscacheutil -flushcache

Join or Leave a Domain (domainjoin-cli)

`domainjoin-cli` is the command-line utility for joining or leaving a domain. For instructions on how to use it, see Join Active Directory from the Command Line.

Display NIS Map (ypcat)

This command is the PBIS Network Information Services (NIS) ypcat function for group passwd and netgroup maps.

/opt/pbis/bin/ypcat

Example usage:

`/opt/pbis/bin/ypcat -d example.com -k map-name`

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/ypcat --help
```

Display the Value of a Key in an NIS Map (ypmatch)

This command is the PBIS Network Information Services (NIS) ypmatch function for group passwd and netgroup maps.

/opt/pbis/bin/ypmatch

Example usage:

```
/opt/pbis/bin/ypmatch -d example.com -k key-name map-name
```

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/ypmatch --help
```

Modify Objects in AD (adtool)

PBIS Enterprise includes a tool to modify objects in Active Directory from the command line of a Linux, Unix, or Mac OS X computer. Located at `/opt/pbis/bin/adtool`, the tool has two interrelated functions:

- Query and modify objects in Active Directory.
- Find and manage objects in PowerBroker cells.

You can view a list of these two categories by executing the following command:

/opt/pbis/bin/adtool --help -a

Here is what the output of the command looks like:

```
[root@rhel5d bin]# ./adtool --help -a

  List of Actions

  Generic Active Directory actions:
  --------------------------------

  add-to-group - add a domain user/group to a security
group.
  delete-object - delete an object.
  disable-user - disable a user account in Active
Directory.
  enable-user - enable a user account in Active
Directory.
  lookup-object - retrieve object attributes.
```

```
  move-object - move/rename an object.
  new-computer - create a new computer object.
  new-group - create a new global security group.
  new-ou - create a new organizational unit.
  new-user - create a new user account.
  remove-from-group - remove a user/group from a security
group.
  reset-user-password - reset user's password.
  search-computer - search for computer objects, print
DNs.
  search-group - search for group objects, print DNs.
  search-object - search for any type of objects using
LDAP filter.
  search-ou - search for organizational units, print DNs
  search-user - search for users, print DNs.

  PowerBroker cell management actions:
  --------------------------------

  add-to-cell - add user/group to a PowerBroker cell.
  delete-cell - delete a PowerBroker cell.
  edit-cell - modify PowerBroker cell properties.
  edit-cell-group - modify properties of a cell's group.
  edit-cell-user - modify properties of a cell's user.
  link-cell - link PowerBroker cells.
  lookup-cell - retrieve PowerBroker cell properties.
  lookup-cell-group - retrieve properties of cell's
group.
  lookup-cell-user - retrieve properties of cell's user.
  new-cell - create a new PowerBroker cell.
  remove-from-cell - remove user/group from a PowerBroker
cell.
  search-cells - search for PowerBroker cells.
  unlink-cell - unlink PowerBroker cells.
```

To get information about the options for each action, use the following syntax:

/opt/pbis/bin/adtool --help -a <ACTION>

Here is an example with the information that is returned:

```
/opt/pbis/bin/adtool --help -a new-user

Usage: adtool [OPTIONS] (-a |--action) new-user
<ARGUMENTS>new-user - create a new user account.

Acceptable arguments ([X] - required):

      --dn=STRING                    DN/RDN of the parent
container/OU containing the
```

```
                                        user. (use '-' for
stdin input)
      --cn=STRING                       Common name (CN) of
the new user. (use '-' for
                                        stdin input)
      --logon-name=STRING               Logon name of the
new user.  (use '-' for stdin
                                        input) [X]
      --pre-win-2000-name=STRING        Pre Windows-2000
logon name.
      --first-name=STRING               First name of the
new user.
      --last-name=STRING                Last name of the new
user.
      --description=STRING              Description of the
user.
      --password=STRING                 User's password.
(use '-' for stdin input)
      --no-password-expires             The password never
expires. If omitted - user
                                        must change password
on next logon.
      --account-enabled                 User account will be
enabled. By default it is
                                        disabled on creation
```

Using the Tool

Privileges: When you run the tool, you must use an Active Directory account with privileges that allow you to perform the command's action. The level of privileges that you need is set by Microsoft Active Directory and is typically the same as performing the corresponding action in Microsoft Active Directory Users and Computers. For example, to add a user to a security group, you must be a member of a security group, such as the enterprise administrators security group, that has privileges to perform the action.

For more information on Active Directory privileges, permissions, and security groups, see the following references on the Microsoft TechNet website:

- Active Directory Privileges
- Active Directory object permissions
- Active Directory Users, Computers, and Groups
- Securing Active Directory Administrative Groups and Accounts

Options There are short and long options. You separate arguments from options with either space or equal sign. If you are not sure about the results of an action you want to execute, run it in read-only mode first (-r). Also it can be useful to set log level to TRACE (-l 5) to see all the execution steps the tool is taking. Authentication SSO by default if the computer is domain-joined. Otherwise, KRB5 via a cached ticket, keytab file, or name/password (unless secure authentication is turned-off (--no-sec)) Name resolution In most cases you can reference objects by FQDN, RDN, UPN, or just names that make sense for a specific action. Use "-" if you want the tool to read values from stdin. This allows you to combine commands via pipes, e.g. search and lookup actions. Multi-forest support You can reference object from a name context (forest) different from the one you are currently connected to, provided that there is a proper trust relation between them. In this way, for instance, you can add a user that lives in one forest to a cell defined in another forest.

Creating a New Cell: When you create a new cell, the tool adds the default primary group (domain users) to the cell. If you are adding a user to the cell and the user has a primary group different from the default group, which is an atypical case, you must add the primary group to the cell, too. The tool does not do it automatically.

Adding Users or Groups Across Domains: If you are adding a user or group to a cell, and the user or group is in a domain different from the one hosting the cell, you must use an account that has write permissions in the cell domain and at least read permissions in the domain hosting the user or group. If, for example, you want to add a user such as CORP\kathy, whose primary group is, say, domain users, to a cell in a domain named CORPQA, two conditions must be met: First, you must be authenticated to the CORPQA domain as a user with administrative rights in the CORPQA domain; second, your user account must exist in the CORP domain with at least read permissions for the CORP domain. Further: Since in this example the primary group of CORP\kathy is CORP\domain users, you must add CORP\domain users to the cell in the CORPQA domain, too.

Automating Commands with a Service Account: To run the tool under a service account, such as a cron job, avoid using krb5 tickets for authentication, especially those cached by the PBIS authentication service in the `/tmp` directory. The tickets may expire and the tool will not renew them. Instead, it is recommended that you create an entry for the service account in a keytab file and use the keytab file for authentication.

Working with a Default Cell: The tool uses the default cell only when the value of the `dn` parameter is the root naming context, such as when you use an expression like `--dn DC=corp,DC=example,DC=com` to represent `corp.example.com`.

Options

To view the tool's options and to see examples of how to use them, execute the following command:

```
/opt/pbis/bin/adtool --help
```

```
[root@rhel5d bin]# ./adtool --help
Usage: adtool [OPTIONS] <ACTION> [ACTION_ARGUMENTS]

HELP OPTIONS
  -u, --usage                   Display brief usage
message
  -?, --help                    Show this message, help
on all actions (-a), or help
                                on a specific action (-a
<ACTION>).
  -v, --version                 Print program version and
exit.

COMMON OPTIONS
  -l, --log-level=LOG_LEVEL     Acceptable values: 1
(error), 2(warning), 3(info),
                                4(verbose) 5 (trace)
(Default: warning).
  -q, --quiet                   Suppress printing to
stdout. Just set the return code.
                                print-dn option makes an
exception.
  -t, --print-dn                Print DNs of the objects
to be looked up, modified or
                                searched for.
  -r, --read-only               Do not actually modify
directory objects when
                                executing actions.

CONNECTION OPTIONS
  -s, --server=STRING           Active Directory server
to connect to.
  -d, --domain=STRING           Domain to connect to.
  -p, --port=INT                TCP port number
  -m, --non-schema              Turn off schema mode

AUTHENTICATION OPTIONS
  -n, --logon-as=STRING         User name or UPN.
  -x, --passwd=STRING           Password for
authentication. (use '-' for stdin input)
  -k, --keytab=STRING           Full path of keytab file,
e.g. /etc/krb5.keytab
  -c, --krb5cc=STRING           Full path of krb5 ticket
cache file, e.g.
                                /tmp/krb5cc_
```

```
foo@example.com
  -z, --no-sec                  Turns off secure
authentication. Simple bind will be
                                used. Use with caution!

ACTION
  -a, --action[=<ACTION>]       Action to execute. Type
'--help -a' for a list of
                                actions, or '--help -a
<ACTION>' for information on a
                                specific action.

Try '--help -a' for a list of actions.
```

Examples

Here is an example that shows how to use two authentication options—`logon-as` and `passwd`—to search Active Directory even though the computer on which the command was executed was not connected to the domain. The account specified in the `logon-as` option is an Active Directory administrative account.

```
root@ubuntu:/opt/pbis/bin# ./adtool -a search-cells --
search-base dc=connecticut,dc=com --logon-as=Administrator -
-passwd=-
```

In this case, the successful result looked like this:

```
Enter password:
CN=$LikewiseIdentityCell,DC=connecticut,DC=com

CN=$LikewiseIdentityCell,OU=mySecureOU,DC=connecticut,DC=com
Total cells: 2
```

Here are a variety of examples. In some of them, the command is broken into two lines and the line break is marked by a back slash (\). In such cases, the back slash is not part of the command.

```
Create OU in a root naming context:
adtool -a new-ou --dn OU=TestOu

Create OU in DC=department,DC=company,DC=com:
adtool -a new-ou --dn
OU=TestOu,DC=department,DC=company,DC=com

Create PowerBroker cell in OU TestOU setting the default
login shell property to /bin/ksh:
```

```
adtool -a new-ou --dn OU=TestOu --default-login-
shell=/bin/ksh

Create a new account for user TestUser in
OU=Users,OU=TestOu:
adtool -a new-user --dn OU=Users,OU=TestOu --
cn=TestUserCN --logon-name=TestUser --password=$PASSWD

Enable the user account:
adtool -a enable-user --name=TestUser

Reset user's password reading the password from
TestUser.pwd file:
cat TestUser.pwd | adtool -a reset-user-password --
name=TestUser --password=- --no-password-expires

Create a new group in OU=Groups,OU=TestOu:
adtool -a new-group --dn OU=Groups,OU=TestOu --pre-win-
2000-name=TestGrooup --name=TestGroup

Look up "description" attribute of an OU specified by
name with a wildcard:
adtool -a search-ou --name='*RootOu' -t | adtool -a
lookup-object --dn=- --attr=description

Look up "unixHomeDirectory" attribute of a user with
samAccountName TestUser:
adtool -a search-user --name TestUser -t | adtool -a
lookup-object --dn=- --attr=unixHomeDirectory

Look up "userAccountControl" attribute of a user with CN
TestUserCN:
adtool -a search-user --name CN=TestUserCN -t | adtool -a
lookup-object --dn=- --attr=userAccountControl

Look up all attributes of an AD object using filter-based
search:
adtool -a search-object --filter
'(&(objectClass=person)(displayName=TestUser))' -t |
adtool -a lookup-object

Add user TestUser to group TestGroup:
adtool -a add-to-group --user TestUser --to-
group=TestGroup

Add group TestGroup2 to group TestGroup:
adtool -a add-to-group --group TestGroup2 --to-
group=TestGroup

Remove user TestUser from group TestGroup:
adtool -a remove-from-group --user TestUser --from-
group=TestGroup
```

```
Rename AD object OU=OldName and move it to a new
location:
adtool -a move-object --from
OU=OldName,DC=department,DC=company,DC=com \
--to OU=NewName,OU=TestOU,DC=department,DC=company,DC=com

Add group TestGroup to PowerBroker cell in TestOU:
adtool -a add-to-cell --dn
OU=TestOU,DC=department,DC=company,DC=com --
group=TestGroup

Remove user TestUser from PowerBroker cell in TestOU:
adtool -a remove-from-cell --dn
OU=TestOU,DC=department,DC=company,DC=com --user=TestUser

Search for cells in a specific location:
adtool -a search-cells --search-base
OU=department,DC=country,DC=company,DC=com

Link cell in OU=TestOU1 to the default cell in
DC=country:
adtool -a link-cell --source-dn
OU=TestOU1,DC=department,DC=company,DC=com \
--target-dn DC=country,DC=company,DC=com

Unlink cell in OU=TestOU1 from the default cell in
DC=country:
adtool -a unlink-cell --source-dn
OU=TestOU1,DC=department,DC=company,DC=com \
--target-dn DC=country,DC=company,DC=com

Change the default login shell property of PowerBroker
cell in TestOU:
adtool -a edit-cell --dn OU=TestOU --default-login-
shell=/bin/csh

Find cells linked to PowerBroker cell in
OU=TestOU,DC=department,DC=company,DC=com:
adtool -a lookup-cell --dn OU=TestOU --linked-cells

Look up login shell property of user TestUser in cell
created in TestOU:
adtool -a lookup-cell-user --dn OU=TestOU --user TestUser
--login-shell

Change login shell property of user TestUser in cell
created in TestOU:
adtool -a edit-cell-user --dn OU=TestOU --user TestUser -
-login-shell=/usr/bin/ksh

Delete a cell object and all its children if any (--
force):
adtool -a delete-object --dn OU=TestOU --force
```

```
Search for PowerBroker cells in root naming context
containing user TestUser:
adtool -a search-cells --user TestUser
```

Copy Files Across Disparate Operating Systems (lwio-copy)

The lwio-copy command-line utility lets you copy files across computers running different operating systems. You can, for example, copy files from a Linux computer to a Windows computer.

There two prerequisites to use lwio-copy: The `lwio` service must be running, and the rdr driver—`/opt/pbis/lib/librdr.sys.so`—must be available as specified by the registry. By default, the rdr driver is available.

The location of the tool is as follows:

/opt/pbis/bin/lwio-copy

To view the tool's arguments, execute the following command on your Unix, Linux, or Mac computer:

```
/opt/pbis/bin/lwio-copy --help
```

Modify Local Accounts

The PBIS local authentication provider for local users and groups includes a full local authentication database. With functionality similar to the local SAM authentication database on every Windows computer, the local authentication provider lets you create, modify, and delete local users and groups on Linux, Unix, and Mac OS X computers by using the following commands.

To execute the commands that modify local accounts, you must use either the root account or an account that has membership in the local administrators group. The account can be an Active Directory account if you manually add it to the local administrators group. For example, you could add the Domain Administrators security group from Active Directory to the local administrators group, and then use an account with membership in the Domain Administrators security group to execute the commands.

Add a Local User (add-user)

This command adds a user to the local authentication database. The command's location is as follows:

/opt/pbis/bin/add-user

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/add-user --help
```

Add a Local Group Member (add-group)

This command adds a group member to the local authentication database. The command's location is as follows:

/opt/pbis/bin/add-group

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/add-group --help
```

Remove a Local User (del-user)

This command deletes a user from the local authentication database. The command's location is as follows:

/opt/pbis/bin/del-user

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/del-user --help
```

Remove a Local Group (del-group)

This command deletes a group from the local authentication database. The command's location is as follows:

/opt/pbis/bin/del-group

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/del-group --help
```

Modify a Local User (mod-user)

This command modifies a user's account settings in the local authentication database, including an account's expiration date and password. You can also enable a user, disable a user, unlock an account, or remove a user from a group. The command's location is as follows:

/opt/pbis/bin/mod-user

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/mod-user --help
```

Modify the Membership of a Local Group (mod-group)

This command adds members to or removes members from a group in the local authentication database. The command's location is as follows:

/opt/pbis/bin/mod-group

Here is an example that demonstrates how to add domain accounts to a local group:

```
/opt/pbis/bin/mod-group --add-members DOMAIN\\Administrator
BUILTIN\\Administrators
```

To view the command's syntax and arguments, execute the following command:

```
/opt/pbis/bin/mod-group --help
```

Kerberos Commands

PowerBroker Identity Services includes several command-line utilities for working with Kerberos. It is recommended that you use these Kerberos utilities, located in `/opt/pbis/bin`, to manage those aspects of Kerberos authentication that are associated with PBIS. For complete instructions on how to use the Kerberos commands, see the man page for the command.

Destroy the Kerberos Ticket Cache (kdestroy)

The `kdestroy` utility destroys the user's active Kerberos authorization tickets obtained through PowerBroker Identity Services. Destroying the user's tickets can help solve logon problems.

Note: This command destroys only the tickets in the PBIS Kerberos cache of the user account that is used to execute the `kdestroy` command; tickets in other Kerberos caches, including root, are not destroyed. To destroy another user's cache, use the command with its `- c` option.

To destroy a user's PBIS Kerberos tickets, execute the following command with the user's account:

`/opt/pbis/bin/kdestroy`

Tip: To view this command's options, type the following command:

```
/opt/pbis/bin/kdestroy -
```

View Kerberos Tickets (klist)

On a target Linux or Unix computer, you can see a list of Kerberos tickets by executing the following command:

`/opt/pbis/bin/klist`

The command lists the location of the credentials cache, the expiration time of each ticket, and the flags that apply to the tickets. For more information, see the man page for `klist`.

Because PowerBroker Identity Services includes its own Kerberos 5 libraries (in `/opt/pbis/lib`), you must use the PBIS `klist` command by either changing directories to `/opt/pbis/bin` or including the path in the command.

Example:

```
-sh-3.00$ /opt/pbis/bin/klist
Ticket cache: FILE:/tmp/krb5cc_593495191
Default principal: hoenstiv@EXAMPLE.COM
Valid starting     Expires            Service principal
07/22/08 16:07:23  07/23/08 02:06:39
krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 07/23/08 04:07:23
07/22/08 16:06:39  07/23/08 02:06:39
host/rhel4d.EXAMPLE.COM@
        renew until 07/23/08 04:07:23
07/22/08 16:06:39  07/23/08 02:06:39
host/rhel4d.EXAMPLE.COM@EXAMPLE.COM
        renew until 07/23/08 04:07:23
07/22/08 16:06:40  07/23/08 02:06:39  RHEL4D$@EXAMPLE.COM
        renew until 07/23/08 04:07:23
```

Note: To address Kerberos issues, see Troubleshooting Kerberos Errors at http://technet.microsoft.com/en-us/library/cc728430(WS.10).aspx.

Obtain and Cache a TGT (kinit)

This command obtains and caches an initial ticket-granting ticket for a principal. The command's location is as follows:

/opt/pbis/bin/kinit

To view the command's options and arguments, execute the following command:

```
man kinit
```

Change a Password (kpasswd)

The kpasswd command changes a Kerberos principal's password on a Linux or Unix computer. (On a Mac computer, use the Mac OS X graphical user interface to change a Kerberos principal's password.) The command's location is as follows:

/opt/pbis/bin/kpasswd

To view the command's options and arguments, execute the following command:

```
man kpasswd
```

The Keytab File Maintenance Utility (ktutil)

This command invokes a shell from which you can read, write, or edit entries in a Kerberos keytab. The command's location is as follows:

/opt/pbis/bin/ktutil

To view the command's options and arguments, execute the following command:

```
man ktutil
```

You can use `ktutil` to add a keytab file to a non-default location. When you join a domain, PowerBroker Identity Services initializes a Kerberos keytab by adding the `default_keytab_name` setting to `krb5.conf` and setting it to `/etc/krb5.keytab`. If the keytab file referenced in krb5.conf does not exist, the PBIS domain-join utility changes the setting to `/etc/krb5.conf`.

You can set the keytab file to be in a location that is different from the default. To do so, you must pre-create the keytab file in the location you want and set a symlink to it in `/etc/krb5.keytab`. Then, you must set the `default_keytab_name` in `/etc/krb5.conf` to point to either the symlink or the real file. The result is that the keytab file will already exist and the PBIS domain-join utility will not modify its location setting.

The keytab's format does not let you create a keytab file without a keytab, but you can use `ktutil` to manually create one with a place-holder entry. When PBIS adds your computer to the domain, a correct entry will be added to the file.

```
/opt/pbis/bin/ktutil
ktutil:  addent -password -p nonexistent@nonexistent -k 1
-e RC4-HMAC
Password for nonexistent@nonexistent:
ktutil:  wkt /var/OtherPlace/etc/krb5.keytab
ktutil:  quit
```

Acquire a Service Ticket and Print Key Version Number (kvno)

This command acquires a service ticket for the specified Kerberos principals and prints out the key version numbers of each. The command's location is as follows:

/opt/pbis/bin/kvno

To view the command's options and arguments, execute the following command:

```
man kvno
```

Manage PBIS Enterprise from the Windows Command Line (lwopt.exe)

Lwopt.exe is a command-line tool installed on Windows computers running PBIS Enterprise. It is installed in the `C:\Program Files\BeyondTrust\PBIS\Enterprise` folder. Command-line tools for the PBIS Enterprise database are discussed in setting up the database.

Lwopt.exe lets you manage options for PBIS Enterprise from the command-line of a Windows administrative workstation connected to Active Directory. You can, for example, set an option to use sequential IDs instead of hashed IDs. In addition, after you set the option to use sequential IDs, you can set the initial UID number for a cell. Setting UIDs below 1,000 is ill-advised, as they can result in a security vulnerability.

```
C:\Program Files\BeyondTrust\PBIS\Enterprise>lwopt
lwopt - configures local Windows options for PowerBroker
Identity Services
Usage:  lwopt OPTIONS
OPTIONS:
   --status            Show current configuration status
   --narrowsearch      Only search the default cell on the
local domain
   --widesearch        Search the default cell across all
domains and
                       two-way forest trusts
   --sequential        Use sequential IDs instead of
hashed IDs
   --hashed            Use hashed IDs
   --foreignaliases    Allow the use of aliases for users
and groups
                       from other domains.
   --noforeignaliases Disallow the use of aliases for
users and groups
                       from other domains.
   --usegc             Use the Global Catalog to speed up
searches (default)
   --ignoregc          Do not use the Global Catalog to
speed up searches
   --startUID=#        Sets the initial UID number for a
cell (if --sequential)
   --startGID=#        Sets the initial GID number for a
cell (if --sequential)
   --minID=#           Sets minimum UID and GID number
configurable through
                       the UI
```

```
   --cell=LDAPPATH    Identifies the cell whose initial
IDs (if --sequential)
                      Example:
LDAP://somedc/ou=anou,dc=somecom,dc=com
   --enableloginnames  Sets the default login names to
all the users enabled
                       in all the cells.
   --disableloginnames Disable the enable default login
names option to all
                       users enabled in all the cells.
   --help              Displays this usage information
If the --startUID or --startGID options are set, the --
cell option must also
be set.
```

Leaving a Domain and Uninstalling the PBIS Agent

You can remove a computer from a domain without necessarily disabling or deleting the computer's account in Active Directory. If needed, you can uninstall the PBIS agent from a client computer.

Leave a Domain

When you remove a computer from a domain, PBIS reverses most PBIS-specific settings that were made to a computer's configuration when it was joined to the domain. PBIS also reverses any changes that you manually made to `/etc/pbis/lsassd.conf` or to the PBIS registry. Changes to the `nsswitch` module, however, are preserved until you uninstall PBIS, when they are reversed. Before you leave a domain, you can execute the following command to view the changes that will take place:

`domainjoin-cli leave --advanced --preview domainName`

Example:

```
[root@rhel4d example]# domainjoin-cli leave --advanced --
preview example.com
Leaving AD Domain:    EXAMPLE.COM
[X] [S] ssh            - configure ssh and sshd
[X] [N] pam            - configure pam.d/pam.conf
[X] [N] nsswitch       - enable/disable PowerBroker
Identity Services nsswitch module
[X] [N] stop           - stop daemons
[X] [N] leave          - disable machine account
[X] [N] krb5           - configure krb5.conf
    [F] keytab         - initialize kerberos keytab

Key to flags
[F]ully configured        - the system is already
configured for this step
[S]ufficiently configured - the system meets the minimum
configuration
                            requirements for this step
[N]ecessary               - this step must be run or
manually performed.
[X]                       - this step is enabled and will
make changes
[ ]                       - this step is disabled and
will not make changes
```

For information on advanced commands for leaving a domain, see Join Active Directory from the Command Line.

Remove the Computer Account in Active Directory

By default, when you remove a computer from a domain, the computer's account in Active Directory is neither disabled nor deleted.

If you want to disable but not delete the computer's account, include the user name as part of the `leave` command. You can include the user name as part of the `leave` command as follows; you will be prompted for the password of the user account:

```
domainjoin-cli leave userName
```

Example: `domainjoin-cli leave brsmith`

Remove a Linux or Unix Computer from a Domain

On the Linux or Unix computer that you want to remove from the Active Directory domain, use a root account to run the following command:

```
/opt/pbis/bin/domainjoin-cli leave
```

Remove a Mac from a Domain

To leave a domain on a Mac OS X computer, you must have administrative privileges on the Mac.

1. In Finder, click **Applications**.
2. In the list of applications, double-click **Utilities**, and then double-click **Directory Access**.
3. On the **Services** tab, click the lock and enter an administrator name and password to unlock it.
4. In the list, click **Likewise**, and then click **Configure**.
5. Enter a name and password of a local machine account with administrative privileges.
6. On the menu bar at the top of the screen, click the **Domain Join Tool** menu, and then click **Join or Leave Domain**.
7. Click **Leave**.

Remove a Mac from a Domain from the Command Line

Execute the following command with an account that allows you to use sudo:

```
sudo /opt/pbis/bin/domainjoin-cli leave
```

Uninstall the Domain Join GUI

On a Linux computer, you can uninstall the domain join GUI from the command line by running the following command as root. The command applies only to Linux computers on which you installed the domain-join GUI as a separate component. In PBIS 6.0 or later, the domain-join GUI is included in the main installation for Linux platforms and cannot be uninstalled separately.

```
/opt/pbis/setup/djgtk/uninstall
```

Uninstall the Agent on a Linux or Unix Computer

Important: Before uninstalling the agent, you must leave the domain and uninstall the domain-join GUI if you installed it as a separate component. Then execute the `uninstall` command from a directory other than `pbis` so that the uninstall program can delete the `pbis` directory and all its subdirectories—for example, execute the command from the root directory.

If you installed the agent on a Linux or Unix computer by using the shell script, you can uninstall the PBIS agent from the command line by using the same shell script with the `uninstall` option. (To uninstall the agent, you must use the shell script with the same version and build number that you used to install it.) For example, on a Linux computer running `glibc`, change directories to the location of PBIS and then run the following command as root, replacing the name of the script with the version you installed:

```
./PBISOpen-6.5.0.94-linux-oldlibc-i386-rpm.sh uninstall
```

For information about the script's options and commands, execute the following command:

```
./PBISOpen-6.5.0.8011-linux-i386-rpm.sh help
```

Uninstall the Agent on a Mac

On a Mac OS X computer, you must uninstall the PBIS agent by using Terminal. Before uninstalling the agent, you should leave the domain.

1. Log on the Mac by using a local account with privileges that allow you to use `sudo`.
2. Open a Terminal window: In Finder, on the **Go** menu, click **Utilities**, and then double-click **Terminal**.
3. At the Terminal shell prompt, execute the following command:

   ```
   sudo /opt/pbis/bin/macuninstall.sh
   ```

Monitoring Events with the Event Log

The PBIS Event Log records and categorizes information about authentication transactions, authorization requests, network events, and other security events on Linux, Unix, and Mac OS X computers. Monitoring events such as failed logon attempts and failed sudo attempts can help prevent unauthorized access to commands, applications, and sensitive resources.

The events are stored in a SQLite database, which is included when you install the PBIS agent. The database is at `/var/lib/pbis/db/lwi_events.db` and its libraries are at `/opt/pbis/lib/`. For viewing and modifying the database, PBIS includes a command-line utility at `/opt/pbis/bin/sqlite3`. For information about SQLite and instructions on how to use the command-line utility, see http://www.sqlite.org/.

The event log records the following events: service initializations, successful logins, failed logins, denied sudo attempts, the application of new Group Policy Objects (GPOs), offline-online transitions and other network connectivity events, and a periodic heartbeat that identifies whether the computer is active.

PBIS includes methods by which you can specify which user and group accounts have read or write access permissions to the event log. The typical methods for setting permissions are the local PBIS configuration registry and PBIS Group Policy settings administered from Active Directory. You can filter events in the event log and you can decide which event categories to log.

Event logging is turned off by default. You can turn on event logging by editing the registry or by using a Group Policy setting. Then, you can configure the options for the log in the registry or manage them with the corresponding Group Policy settings. Keep in mind that Group Policy settings are available only with PBIS Enterprise; PBIS Open does not apply Group Policy settings.

After you modify the settings in the registry, you must restart the event log service with the root account for the changes to take effect:

```
/opt/pbis/bin/lwsm refresh eventlog
```

For information about managing the event log with the registry, see Configuring PBIS with the Registry. For information about managing the event log with Group Policy settings, see the *PowerBroker Identity Services Group Policy Administration Guide*.

View the Local Event Log

On a Linux, Unix, or Mac OS X computer, you view the local PBIS Event Log by using the eventlog command-line utility with the root account:

/opt/pbis/bin/eventlog-cli

To view the command's arguments, execute the following command:

```
/opt/pbis/bin/eventlog-cli -h
```

You can gain access to the event log by using either `localhost` or the virtual loopback interface of the computer, which is typically assigned to the address `127.0.0.1`.

To view a summary of events, execute the following command with the root account:

```
/opt/pbis/bin/eventlog-cli -s - localhost
```

Example output:

```
========================================
Event Record: (392/396) (392 total)
========================================
Event Record ID......... 392
Event Table Category.... System
Event Type.............. Information
Event Date.............. 2010-02-16
Event Time.............. 07:37:58 AM
Event Source............ Likewise LSASS
Event Category.......... Service
Event Source ID......... 1004
Event User.............. SYSTEM
Event Computer.......... example03
Event Description....... Likewise authentication service
provider configuration settings have been reloaded.

     Authentication provider:           lsa-
activedirectory-provider
     Current settings are...
     Cache reaper timeout (secs):       2592000
     Cache entry expiry (secs):         14400
     Space replacement character:       '^'
     Domain separator character:        '\'
     Enable event log:                  true
     Logon membership requirements:
        CORP\EXAMPLE03_Users
        CORP\EnterpriseTeam
     Log network connection events:     false
     Create K5Login file:               true
     Create home directory:             true
     Sign and seal LDAP traffic:        false
```

```
     Assume default domain:             false
     Sync system time:                  true
     Refresh user credentials:          true
     Machine password sync lifetime:    2592000
     Default Shell:                     /bin/sh
     Default home directory prefix:     /Users
     Home directory template:           %H/local/%D/%U
     Umask:                             18
     Skeleton directory:
System/Library/User Template/Non_localized,
/System/Library/User Template/English.lproj
     Cell support:                      Invalid
     Trim user membership:              true
     NSS group members from cache only: false
     NSS user members from cache only:  false
     NSS enumeration enabled:           true
     Domain Manager check domain online (secs):
300
     Domain Manager unknown domain cache timeout (secs):
3600

=====================================
```

Or, with the following command, you can view the event log in table format:

`/opt/pbis/bin/eventlog-cli -t - localhost`

Example:

```
[root@rhel5d bin]# su example\\user2
[EXAMPLE\user2@rhel5d bin]$ sudo blah
Password:
Sorry, try again.
Password:
Sorry, try again.
Password:
sudo: 2 incorrect password attempts
[EXAMPLE\user2@rhel5d bin]$ exit
[root@rhel5d bin]# /opt/pbis/bin/eventlog-cli -t -
localhost
Id:| Type          | Time        | Source         |
Category     | Event | User
83 | Information   | 02:11:29 PM | Likewise LSASS |
Service      | 1004  | SYSTEM
84 | Success Audit | 02:13:07 PM | Likewise LSASS |
Login/Logoff | 1201  | EXAMPLE\user2
85 | Failure Audit | 02:13:30 PM | Likewise LSASS |
Login/Logoff | 1205  | EXAMPLE\user2
86 | Failure Audit | 02:13:33 PM | Likewise LSASS |
Login/Logoff | 1205  | EXAMPLE\user2
87 | Failure Audit | 02:13:39 PM | Likewise LSASS |
```

```
Login/Logoff | 1205  | EXAMPLE\user2
88 | Success Audit | 02:14:57 PM | Likewise LSASS |
Login/Logoff | 1220  | EXAMPLE\user2
[root@rhel5d bin]#
```

You can also use SQL filters to query the event log by event type, source ID, and a variety of other field names. Example:

```
[root@rhel5d bin]# /opt/pbis/bin/eventlog-cli -s
"(EventType = 'Failure Audit') AND (EventSourceId =
1205)" localhost
Event Record: (1/3) (1 total)
========================================
Event Record ID......... 85
Event Table Category.... Security
Event Type............... Failure Audit
Event Date............... 2009-07-29
Event Time............... 02:13:30 PM
Event Source............ Likewise LSASS
Event Category.......... Login/Logoff
Event Source ID......... 1205
Event User............... EXAMPLE\user2
Event Computer.......... rhel5d
Event Description....... Logon Failure:

     Authentication provider: lsa-activedirectory-
provider

     Reason:                  Unknown username or bad
password
     User Name:               EXAMPLE\user2
     Login phase:             User authenticate
Event Data............... Error: The password is incorrect
for the given username [error code: 32789]
========================================
```

Event Types

The `Event Type` is typically one of the following:

```
SUCCESS_AUDIT_EVENT_TYPE    "Success Audit"
FAILURE_AUDIT_EVENT_TYPE    "Failure Audit"
INFORMATION_EVENT_TYPE      "Information"
WARNING_EVENT_TYPE          "Warning"
ERROR_EVENT_TYPE            "Error"
```

Event Sources

The `Event Source` is typically one of the following values:

- Likewise LSASS
- Likewise GPAGENT
- Likewise DomainJoin
- Likewise NETLOGON
- System Log

Event Source IDs

Each event source defines its own list of `Event Source Id` values. Here is a list of events categorized by source.

```
========================================
EventSource = "Likewise LSASS"

LSASS_EVENT_INFO_SERVICE_STARTED
   1000
LSASS_EVENT_ERROR_SERVICE_START_FAILURE
   1001
LSASS_EVENT_INFO_SERVICE_STOPPED
   1002
LSASS_EVENT_ERROR_SERVICE_STOPPED
   1003
LSASS_EVENT_INFO_SERVICE_CONFIGURATION_CHANGED
   1004

// Logon events
LSASS_EVENT_SUCCESSFUL_LOGON_AUTHENTICATE
   1200
LSASS_EVENT_SUCCESSFUL_LOGON_CREATE_SESSION
   1201
LSASS_EVENT_SUCCESSFUL_LOGON_CHECK_USER
   1203
LSASS_EVENT_FAILED_LOGON_UNKNOWN_USERNAME_OR_BAD_PASSWORD
   1205
LSASS_EVENT_FAILED_LOGON_TIME_RESTRICTION_VIOLATION
   1206
LSASS_EVENT_FAILED_LOGON_ACCOUNT_DISABLED
   1207
LSASS_EVENT_FAILED_LOGON_ACCOUNT_EXPIRED
   1208
LSASS_EVENT_FAILED_LOGON_MACHINE_RESTRICTION_VIOLATION
   1209
LSASS_EVENT_FAILED_LOGON_TYPE_OF_LOGON_NOT_GRANTED
   1210
LSASS_EVENT_FAILED_LOGON_PASSWORD_EXPIRED
   1211
LSASS_EVENT_FAILED_LOGON_NETLOGON_FAILED
   1212
LSASS_EVENT_FAILED_LOGON_UNEXPECTED_ERROR
   1213
```

```
LSASS_EVENT_FAILED_LOGON_ACCOUNT_LOCKED
   1214
LSASS_EVENT_FAILED_LOGON_CHECK_USER
   1215

LSASS_EVENT_LOGON_PHASE_AUTHENTICATE
   1
LSASS_EVENT_LOGON_PHASE_CREATE_SESSION
   2
LSASS_EVENT_LOGON_PHASE_CHECK_USER
   3

// Logoff events
LSASS_EVENT_SUCCESSFUL_LOGOFF
   1220

// User password change events
LSASS_EVENT_SUCCESSFUL_PASSWORD_CHANGE
   1300
LSASS_EVENT_FAILED_PASSWORD_CHANGE
   1301
LSASS_EVENT_SUCCESSFUL_USER_ACCOUNT_KERB_REFRESH
   1302
LSASS_EVENT_FAILED_USER_ACCOUNT_KERB_REFRESH
   1303

// Machine password change events
LSASS_EVENT_SUCCESSFUL_MACHINE_ACCOUNT_PASSWORD_UPDATE
   1320
LSASS_EVENT_FAILED_MACHINE_ACCOUNT_PASSWORD_UPDATE
   1321
LSASS_EVENT_SUCCESSFUL_MACHINE_ACCOUNT_TGT_REFRESH
   1322
LSASS_EVENT_FAILED_MACHINE_ACCOUNT_TGT_REFRESH
   1323

// Account management events
LSASS_EVENT_ADD_USER_ACCOUNT
   1400
LSASS_EVENT_DELETE_USER_ACCOUNT
   1401
LSASS_EVENT_ADD_GROUP
   1402
LSASS_EVENT_DELETE_GROUP
   1403

// Lsass provider events
LSASS_EVENT_SUCCESSFUL_PROVIDER_INITIALIZATION
   1500
LSASS_EVENT_FAILED_PROVIDER_INITIALIZATION
   1501
LSASS_EVENT_INFO_REQUIRE_MEMBERSHIP_OF_UPDATED
   1502
```

```
LSASS_EVENT_INFO_AUDITING_CONFIGURATION_ENABLED
   1503
LSASS_EVENT_INFO_AUDITING_CONFIGURATION_DISABLED
   1504

// Runtime warnings
LSASS_EVENT_WARNING_CONFIGURATION_ID_CONFLICT
   1601
LSASS_EVENT_WARNING_CONFIGURATION_ALIAS_CONFLICT
   1602

// Network events
LSASS_EVENT_INFO_NETWORK_DOMAIN_ONLINE_TRANSITION
   1700
LSASS_EVENT_WARNING_NETWORK_DOMAIN_OFFLINE_TRANSITION
   1701


========================================
EventSource = "Likewise DomainJoin"

DOMAINJOIN_EVENT_INFO_JOINED_DOMAIN            1000
DOMAINJOIN_EVENT_ERROR_DOMAIN_JOIN_FAILURE     1001
DOMAINJOIN_EVENT_INFO_LEFT_DOMAIN              1002
DOMAINJOIN_EVENT_ERROR_DOMAIN_LEAVE_FAILURE    1003


========================================
EventSource = "Likewise GPAGENT"

GPAGENT_EVENT_INFO_SERVICE_STARTED
 1000
GPAGENT_EVENT_ERROR_SERVICE_START_FAILURE
 1001
GPAGENT_EVENT_INFO_SERVICE_STOPPED
 1002
GPAGENT_EVENT_ERROR_SERVICE_STOPPED
 1003
GPAGENT_EVENT_INFO_SERVICE_CONFIGURATION_CHANGED
 1004

// GPAgent policy update events
GPAGENT_EVENT_POLICY_UPDATED
 1100
GPAGENT_EVENT_POLICY_UPDATE_FAILURE
 1101

// GPAgent policy processing issue events
GPAGENT_EVENT_INFO_POLICY_PROCESSING_ISSUE_RESOLVED
 1200
GPAGENT_EVENT_ERROR_POLICY_PROCESSING_ISSUE_ENCOUNTERED
 1201
```

```
========================================
EventSource = "Likewise NETLOGON"

// Netlogon service events
LWNET_EVENT_INFO_SERVICE_STARTED
   1000
LWNET_EVENT_ERROR_SERVICE_START_FAILURE
   1001
LWNET_EVENT_INFO_SERVICE_STOPPED
   1002
LWNET_EVENT_ERROR_SERVICE_STOPPED
   1003
LWNET_EVENT_INFO_SERVICE_CONFIGURATION_CHANGED
   1004


========================================
EventSource = "System Log"

Syslog entries are parsed by the reapsysl service
to create PBIS eventlog entries for the following:

Text console logon failure
   1
Text console logon success
   2
SSH logon failure
   3
SSH logon success
   4
SUDO bad password
   5
SUDO access denied
   6
SUDO success
   7
SSH with AD account failure
   8
SSH with AD account success
   9
Text console login with AD account failure
   10
Text console login with AD account success
   11
```

Single Sign-On Using PBIS

When you log on a Linux, Unix, or Mac OS X computer by using your Active Directory domain credentials, PowerBroker Identity Services initializes and maintains a Kerberos ticket granting ticket (TGT). The TGT lets you log on other computers joined to Active Directory or applications provisioned with a service principal name and be automatically authenticated with Kerberos and authorized for access through Active Directory. In a transparent process, the underlying Generic Security Services (GSS) system requests a Kerberos service ticket for the Kerberos-enabled application or server. The result: single sign-on.

To gain access to another computer, you can use various protocols and applications:

- SSH (See how to configure single sign-on for SSH, including platform-specific issues.)
- rlogin
- rsh
- Telnet
- FTP
- Firefox (for browsing of intranet sites)
- LDAP queries against Active Directory
- HTTP with an Apache HTTP Server

How PBIS Makes SSO Happen

Since Microsoft Windows 2000 was released, Active Directory's primary authentication protocol has been Kerberos. When a user logs on to a Windows computer that is joined to a domain, the operating system uses the Kerberos protocol to establish a key and to request a ticket for the user. Active Directory serves as the Kerberos key distribution center, or KDC.

PBIS configures Linux and Unix computers to interact with Active Directory in a similar way. When a user logs on a Linux and Unix computer joined to a domain, PBIS requests a ticket for the user. The ticket can then be used to implement SSO with other applications.

PBIS fosters the use of the highly secure Kerberos 5 protocol by automating its configuration on Linux and Unix computers. To ensure that the Kerberos authentication service is properly configured, PBIS does the following:

- Ensures that DNS is properly configured to resolve names associated with Active Directory (AD).

- Performs secure, dynamic DNS updates to ensure that Linux and Unix computer names can be resolved with AD-integrated DNS servers.
- Configures Kerberos. In an environment with multiple KDCs, PBIS makes sure that Kerberos selects the right server.
- Configures SSHD to support SSO through Kerberos by using GSSAPI.
- Creates a keytab for the computer in the following way: When you join a Linux or Unix computer to AD, PBIS creates a computer account for the computer. PBIS then automatically creates a keytab for the SPN and places it in the standard system location (typically `/etc/krb5.keytab`).
- Creates a keytab for the user during logon. On most systems, the user keytab is placed in the `/tmp` directory and named `krb5cc_UID`, where `UID` is the numeric user ID assigned by the system.

How to Implement SSO with PBIS

When you install PBIS on a Linux, Unix, or Mac OS X computer and join it to Active Directory, PBIS prepares it for single sign-on by creating a keytab for the computer. However, when you use PBIS to implement SSO with other applications or services, you will likely have to configure the application to use GSSAPI and Kerberos 5 authentication and you will likely have to provision each application user for external Kerberos authentication. At the very least, you will have to provision your application with a service principal name in Active Directory. A service principal name, or SPN, is the name with which a client uniquely identifies an instance of a service. Kerberos then uses the SPN to authenticate a service.

Note: Configuring an external application for SSO with Kerberos is beyond the scope of the PBIS documentation; for more information, see the vendor's manual for your application.

The following process outlines the steps for setting up an application or service to use PBIS for single sign-on. For a detailed example of how to configure an application for SSO, see Configure Apache for SSO. For examples of how to create a service account in AD, register an SPN for the service account, and create a keytab for the SPN, see creating a Kerberos service principal and keytab file for SSO on the IBM website.

1. Create a service account for the application in Active Directory.
2. Associate a service principal name, or SPN, with the service account in Active Directory; see the overview of setspn.exe on Microsoft TechNet.
3. Create a keytab for the SPN with the ktpass utility.
4. Place the keytab in the appropriate location on the Linux or Unix computer.

5. Configure the authentication module to get its Kerberos key from the generated keytab.
6. Configure the authentication module to determine appropriate roles by examining Active Directory group membership.
7. Configure an application to restrict access to Active Directory authenticated users in certain roles.
8. Test SSO by accessing restricted websites from a Windows client running Microsoft Internet Explorer or Mozilla Firefox. Repeat this step on Linux and Unix using Firefox.

Enable PAM for SSH

If your Active Directory account is not working with SSH, make sure that `UsePAM` is enabled in `sshd_config` and make sure that your sshd is linked to the PAM libraries.

1. Determine which sshd is running by executing the following command:

```
bash-3.2# ps -ef | grep sshd
    root  8199     1  0  Feb  6  ?         0:00
/opt/ssh/sbin/sshd
    root  2987  8199  0  Mar  3  ?         0:04 sshd:
root@notty
    root 24864  8199  0 12:16:25 ?         0:00 sshd:
root@pts/0
    root  2998  8199  0  Mar  3  ?         0:05 sshd:
root@notty
    root 24882 24880  0 12:16:54 pts/0     0:00 grep
sshd
```

2. Either use `lsof` to find out which conf file it is reading, or start it up with debugging to figure out the default path. Example:

```
username@computer:~$ /usr/sbin/sshd -dd -t
   debug2: load_server_config: filename /etc/ssh/sshd_
config
   debug2: load_server_config: done config len = 664
   debug2: parse_server_config: config /etc/ssh/sshd_
config len 664
   debug1: sshd version OpenSSH_5.1p1 Debian-3ubuntu1
   Could not load host key: /etc/ssh/ssh_host_rsa_key
   Could not load host key: /etc/ssh/ssh_host_dsa_key
```

3. Verify that `UsePAM` is enabled in the config file. As a best practice, make a backup copy of the configuration file before you change it.

4. Run `ldd` on `sshd` to make sure it links with `libpam`. Example from an IA64 HP system:

```
bash-3.2# ldd /opt/ssh/sbin/sshd
        libpam.so.1 =>  /usr/lib/hpux64/libpam.so.1
        libdl.so.1 =>   /usr/lib/hpux64/libdl.so.1
        libnsl.so.1 =>  /usr/lib/hpux64/libnsl.so.1
        libxnet.so.1 => /usr/lib/hpux64/libxnet.so.1
        libsec.so.1 =>  /usr/lib/hpux64/libsec.so.1
        libgssapi_krb5.so =>
/usr/lib/hpux64/libgssapi_krb5.so
        libkrb5.so =>   /usr/lib/hpux64/libkrb5.so
        libpthread.so.1 =>
/usr/lib/hpux64/libpthread.so.1
        libc.so.1 =>    /usr/lib/hpux64/libc.so.1
        libxti.so.1 =>  /usr/lib/hpux64/libxti.so.1
        libxti.so.1 =>  /usr/lib/hpux64/libxti.so.1
        libm.so.1 =>    /usr/lib/hpux64/libm.so.1
        libk5crypto.so =>
/usr/lib/hpux64/libk5crypto.so
        libcom_err.so =>
/usr/lib/hpux64/libcom_err.so
        libk5crypto.so =>
/usr/lib/hpux64/libk5crypto.so
        libcom_err.so =>
/usr/lib/hpux64/libcom_err.so
        libdl.so.1 =>   /usr/lib/hpux64/libdl.so.1
bash-3.2#
```

Configure PuTTY for Windows-Based SSO

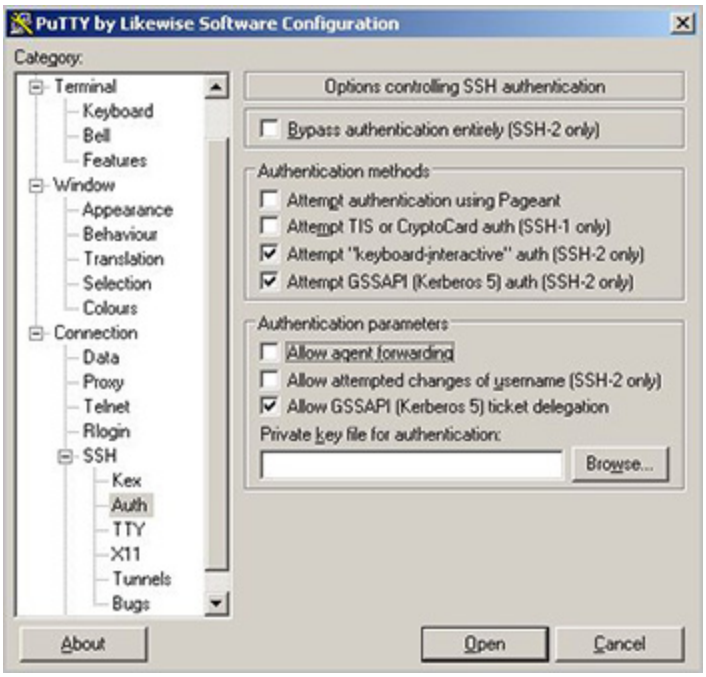
To use PuTTY to connect to a Linux or Unix machine from a Windows machine and then connect to a second Linux or Unix, you must configure PuTTY to allow ticket forwarding and you must set the base Linux or Unix computer in Active Directory to be trusted for delegation.

Important: The following procedure assumes that you are using a GSSAPI-enhanced version of PuTTY, such as PuTTY by BeyondTrust Software, Inc., which you can download at www.beyondtrust.com. The procedure also assumes that there are DNS entries for all three computers and that you use host names to connect to the target computers. If DNS search domains are properly setup on your client systems, you can use short host names.

Configure PuTTY

1. In the PuTTY Configuration dialog, select **Allow GSSAPI (Kerberos 5) ticket delegation**. (With some versions of PuTTY, the option is named **Allow Kerberos 5 ticket forwarding (SSH 1/2)**.)

2. Select **Attempt GSSAPI (Kerberos 5) auth (SSH-2 only)**. With some versions of PuTTY, the option is named **Attempt GSSAPI/Kerberos 5 authentication**.



Configure the Base Linux Computer in Active Directory

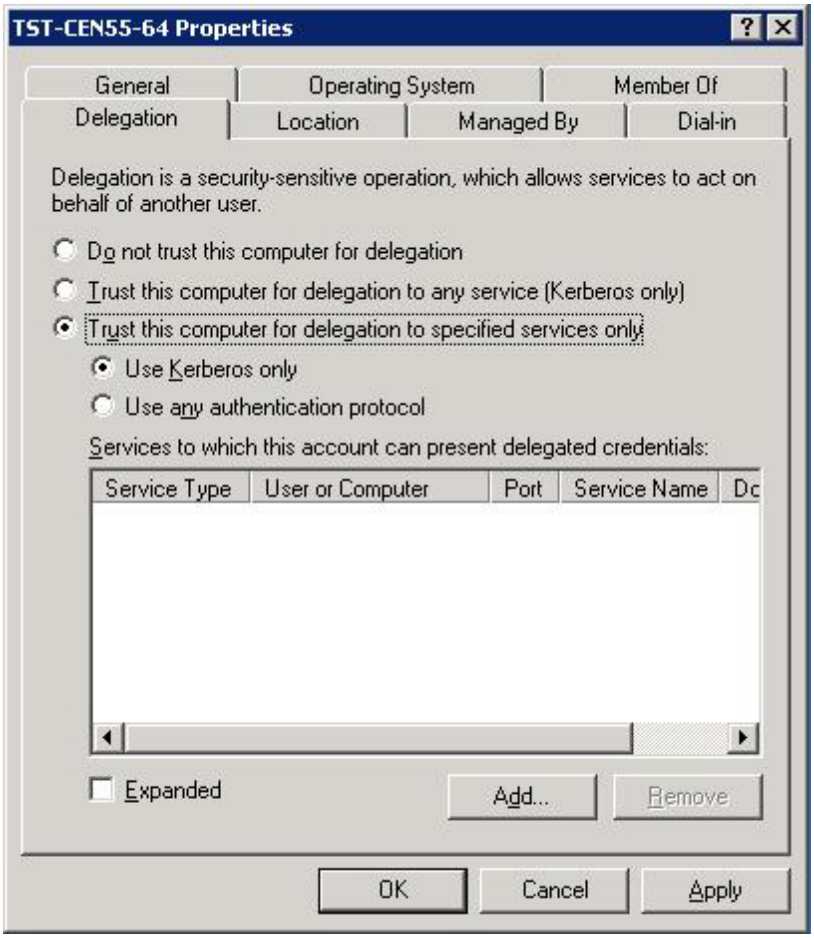
This procedure assumes the base Linux or Unix computer is joined to Active Directory with PBIS. To perform this procedure, you must be a member of the Domain Administrators security group or the Enterprise Administrators security group, or you must have been delegated authority.

Windows Server 2003 R2

1. In Active Directory Users and Computers, in the console tree, click **Computers**.

2. In the details pane, right-click the computer that you want, and then click **Properties**.

3. On the **Delegation** tab, click **Trust this computer for delegation to specified services only**:



4. Confirm that **Use Kerberos only** is selected.

5. Click **Add** and, in **Add Services**, click **Users and Computers**.

6. In **Enter the object names to select**, type the name of the user or computer that the computer will be trusted to delegate for, and then click **OK**.

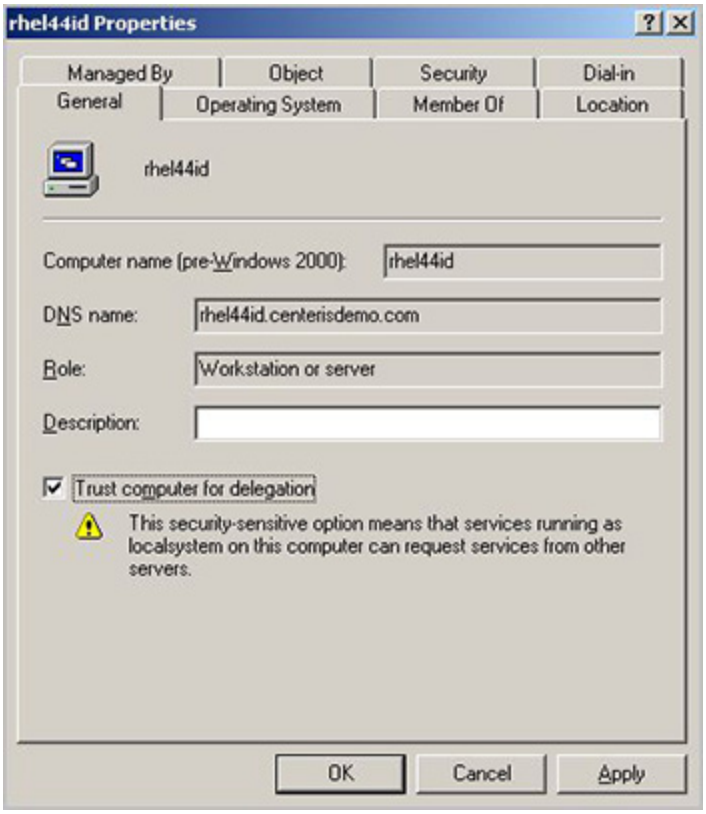
7. In **Add Services**, click the service or services that will be trusted for delegation and then click **OK**.

Windows 2000

1. In Active Directory Users and Computers, in the console tree, click **Computers**.

2. In the details pane, right-click the computer that you want, and then click **Properties**.

3. On the **General** tab, select **Trust computer for delegation**:



Configure Apache for SSO

This topic describes how to configure PowerBroker Identity Services and the Apache HTTP Server to provide single sign-on authentication through Active Directory with Kerberos 5. The instructions assume that you know how to administer Active Directory, the Apache HTTP Server, and computers running Linux.

Single sign-on for the Apache HTTP server uses the Simple and Protected GSS-API Negotiation Mechanism, or SPNEGO, to negotiate authentication with Kerberos. SPNEGO is an Internet standard documented in RFC 2478 and is commonly referred to as the negotiate authentication protocol. The PBIS `mod_auth_kerb` module lets an Apache web server running on a Linux or Unix system authenticate and authorize users based on their Active Directory domain credentials.

For information about configuring web browsers to use SSO after you have configured Apache, see Configure Firefox for SSO or Configure Internet Explorer for SSO.

For information about resolving issues with Kerberos authentication, see Troubleshooting Kerberos Authentication.

Prerequisites

- PBIS Open or PBIS Enterprise installed on the Linux computer running your Apache HTTP Server.
- Application integration package downloaded from the BeyondTrust website and installed. The file name should be similar to `PBISAppIntegration-6.5.0.8656-linux-i386-rpm.sh`. This installs the Apache `mod_auth_kerb` module that is required to configure your Apache HTTP Server for single sign-on.
- The Linux or Unix computer that is hosting the Apache web server is joined to Active Directory.
- An Apache HTTP Server 2.0 or 2.2 that supports dynamically loaded modules. To check whether your Apache web server supports dynamically loaded modules, execute the following command and verify that `mod_so.c` appears in the list of compiled modules:

  ```
  httpd -l
  ```

  ```
  Compiled in modules:
    core.c
    prefork.c
    http_core.c
    mod_so.c
  ```

  For Apache installations that are compiled from the source code, make sure that `--enable-module=so` is specified when `./configure` is executed:

  ```
  ./configure --enable-module=so
  ```

- Your Kerberos libraries must support SPNEGO. For example, MIT Kerberos libraries that are version 1.5 and later support SPNEGO; earlier versions do not. Make sure your Kerberos libraries support SPNEGO by running `ldd`:

```
which httpd
/usr/sbin/httpd
ldd /usr/sbin/httpd
```

In the results, find the line that references `libgssapi`:

```
libgssapi_krb5.so.2 => /usr/lib/libgssapi_krb5.so.2
(0x00231000)
```

Finally, query the version number of the library and make sure it is **1.5 or later**:

```
rpm -qif /usr/lib/libgssapi_krb5.so.2
```

```
Name        : krb5-libs
                   Relocations: (not
relocatable)
Version     : 1.5
                               Vendor: Red Hat,
Inc.
Release     : 17
                           Build Date: Tue 16
Jan 2007 10:01:00 AM PST
Install Date: Fri 14 Dec 2007 09:09:44 AM PST
     Build Host: ls20-bc1-13.build.redhat.com
Group       : System Environment/Libraries
  Source RPM: krb5-1.5-17.src.rpm
Size        : 1333337
                         License: MIT, freely
distributable.
Signature   : DSA/SHA1, Wed 17 Jan 2007
10:57:33 AM PST, Key ID 5326810137017186
Packager    : Red Hat, Inc.
<http://bugzilla.redhat.com/bugzilla>
URL         : http://web.mit.edu/kerberos/www/
Summary     : The shared libraries used by
Kerberos 5.
```

```
Description :
Kerberos is a network authentication system.
The krb5-libs package
contains the shared libraries needed by
Kerberos 5. If you are using
Kerberos, you need to install this package.
[root@rhel5d sbin]#
```

Configure Apache HTTP Server 2.2 for SSO on RHEL 5

The following instructions demonstrate how to configure PBIS and Apache for SSO on a Red Hat Enterprise Linux 5 computer. The steps vary by operating system and by Apache version. Ubuntu, in particular, uses `apache2` instead of `httpd` for commands, the name of the daemon, the configuration directory, the name of the configuration file, and so forth.

Important: Configuring web servers is complex. Before you deploy your configuration to a production web server, implement and test it in a test environment. More: Before you change your web server's configuration, read and understand the Apache HTTP Server documentation at http://httpd.apache.org/docs/ and the mod_auth_kerb documentation at http://modauthkerb.sourceforge.net/configure.html. Before you change a file, make a backup copy of it.

1. Determine whether your Apache server is 2.0 or 2.2 by running the following command:

   `httpd -v`

   ```
   Server version: Apache/2.2.3
   Server built:   Nov 29 2006 06:33:19
   ```

2. Edit your Apache configuration file—`/etc/httpd/conf/httpd.conf`—to add a directive to load the PBIS `auth_kerb_module` for your version of Apache. Since my Red Hat computer is running Apache 2.2.3, I have added the 2.2 version of the module to the list after the other `auth` modules (which were already listed in the file):

```
LoadModule auth_basic_module modules/mod_auth_
basic.so
LoadModule auth_kerb_module
/opt/pbis/apache/2.2/mod_auth_kerb.so
```

3. In `/etc/httpd/conf/httpd.conf`, configure authentication for a directory and then restart the web server; example:

```
<Directory "/var/www/html/secure">Options Indexes
MultiViews FollowSymLinks
AllowOverride None
Order deny,allow
Deny from all
Allow from 127.0.0.0/255.0.0.0 ::1/128
AuthType Kerberos
AuthName "Kerberos Login"
KrbAuthRealms EXAMPLE.COM
Krb5Keytab /etc/apache2/http.ktb
Require valid-user
</Directory>
```

Tip: You can require that a user be a member of a security group to access the Apache web server by replacing `Require valid-user` with `Require group name-of-your-group`, as shown in the example below. To control group access by requiring group membership, however, you must first install and load `mod_auth_pam`; for instructions on how to set up `mod_auth_pam`, see http://pam.sourceforge.net/mod_auth_pam/install.html. (Because `mod_auth_pam` is no longer maintained, you should consider using `mod_authz_unixgroup` instead; see the instructions later in this section.)

```
<Directory "/var/www/html/secure">Options Indexes
MultiViews FollowSymLinks
AllowOverride None
Order deny,allow
Deny from all
Allow from 127.0.0.0/255.0.0.0 ::1/128
AuthType Kerberos
AuthName "Kerberos Login"
KrbAuthRealms EXAMPLE.COM
Krb5Keytab /etc/apache2/http.ktb
Require group linuxfulladmins
</Directory>
```

4. Configure your web server for Secure Socket Layer (SSL). For instructions, see the Apache HTTP Server documentation.

   Important: If SSO fails and you have not turned on SSL, your server will prompt you for an ID and password—which will be sent in clear text. SSL encrypts all data that passes between the client browser and the web server. SSL can also perform Basic Authentication in a secure fashion, providing a fallback mechanism in the event that Kerberos authentication fails. Using SSL is especially important if the protected website also needs to be accessible from outside the corporate network. For more information, see http://modauthkerb.sourceforge.net/configure.html.

5. In Active Directory, create a user account for the Apache web server in the same OU (or, with PBIS Enterprise, cell) to which the Linux computer hosting the web server is joined. Set the password of the user account to never expire. In the examples that follow, the user account for my Apache web server is named `httpUser`.

6. On the domain controller, create an RC4-HMAC keytab for the Apache web server by using Microsoft's ktpass utility. For information on ktpass, see http://technet.microsoft.com/en-us/library/cc776746.aspx. The keytab that you must create can vary by Windows version.

   Example:

```
C:\>ktpass /out keytabfile /princ
HTTP/rhel5d.example.com@EXAMPLE.COM /pass
SkiAlta2008 /mapuser example\httpUser /ptype
KRB5_NT_PRINCIPAL
Targeting domain controller: steveh-
dc.example.com
Using legacy password setting method
Successfully mapped HTTP/rhel5d.example.com to
httpUser.
Key created.
Output keytab to keytabfile:
Keytab version: 0x502
keysize 80 HTTP/rhel5d.example.com@EXAMPLE.COM
ptype 0 (KRB5_NT_UNKNOWN) vno 3 etype 0x17
(RC4-HMAC) keylength 16
(0x2998807dc299940e2c6c81a08315c596)
```

   Note: On Windows 2000, do not specify the domain name as part of the `/mapuser` parameter; just enter the name of the user.

7. Use secure FTP or another method to transfer the keytab file to the Linux computer that hosts your Apache web server and place the file in the location specified in your `<Directory>` configuration in `httpd.conf`. For example, using the configuration shown in Step 3 above, the keytab file would be placed in `/etc/apache2/http.ktb`.

8. Set the permissions of the keytab file to be readable by the ID under which the Apache web server runs and no one else.

   Important: The Kerberos keytab file is necessary to authenticate incoming requests. It contains an encrypted, local copy of the host's key and, if compromised, might allow unrestricted access to the host computer. It is therefore crucial to protect it with file-access permissions.

Control Group Access with mod_authz_unixgroup

Instead of using the `mod_auth_pam`, which is no longer maintained, you can require that a user be a member of a security group to access the Apache web server by using `mod_authz_unixgroup`. First, install `mod_authz_unixgroup`:

```
yum install httpd-devel
wget http://mod-auth-external.googlecode.com/files/mod_
authz_unixgroup-1.0.2.tar.gz
tar -xzvf mod_authz_unixgroup-1.0.2.tar.gz
cd mod_authz_unixgroup-1.0.2
apxs -c mod_authz_unixgroup.c
apxs -i -a mod_authz_unixgroup.la
```

Then, in `/etc/httpd/conf/httpd.conf`, replace `Require valid-user` with `AuthzUnixgroup on` and `Require group name-of-your-group`:

```
<Directory "/var/www/html/secure">...
KrbAuthRealms EXAMPLE.COM
Krb5Keytab /etc/apache2/http.ktb
AuthzUnixgroup on
Require group linuxfulladmins
</Directory>
```

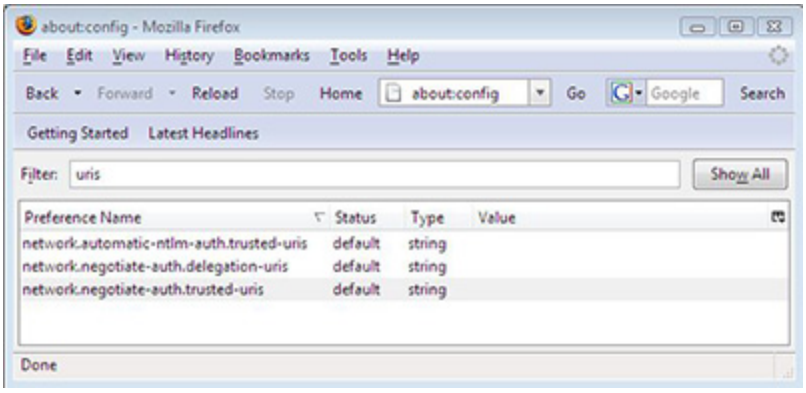
For more information, see the documentation for `mod_authz_unixgroup` .

Configure Firefox for SSO

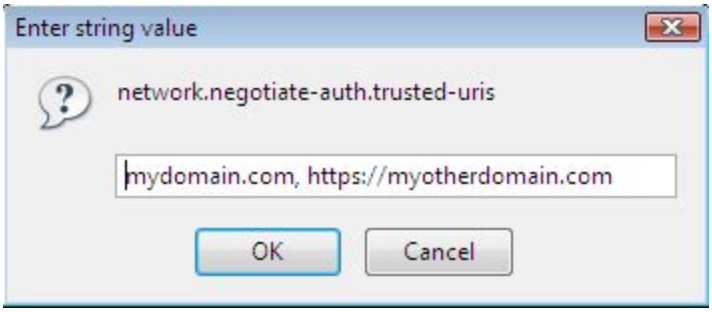
To set up Firefox for single sign-on, you must turn on the Simple and Protected GSS-API Negotiation Mechanism, or SPNEGO, to negotiate authentication with Kerberos.

1. Open Firefox.
2. In the **Go** box, type `about:config`, and then click **Go**.

3. In the **Filter** box, type `uris`.



4. Double-click **network.negotiate-auth.trusted-uris**, enter a comma-separated list of URL prefixes or domains that are permitted to engage in SPNEGO authentication with the browser, and then click **OK**. Example:

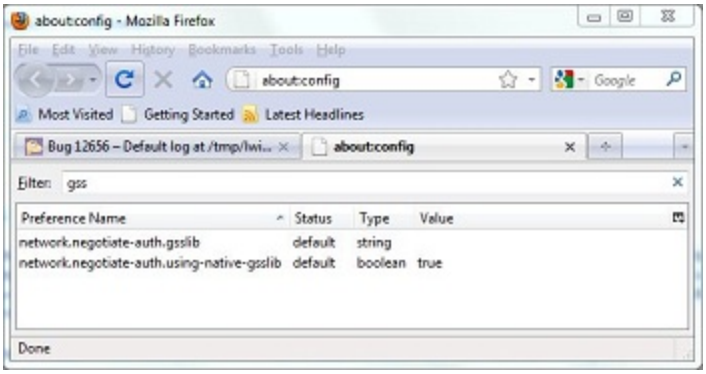


5. Double-click **network.negotiate-auth.delegation-uris**, enter a comma-separated list of the sites for which the browser may delegate user authorization to the server, and then click **OK**.

   For more information on how to configure Firefox, see http://grolmsnet.de/kerbtut/firefox.html.

6. To negotiate with your web server through the GSSAPI by using NTLM as the preferred authentication protocol on a Mac OS X computer, you must also modify the GSS preferences as follows. To find the preferences, type `gss` into Firefox's filter box:

```
network.negotiate-auth.gsslib  user set  string
/opt/pbis/lib/libgssapi_krb5.2.2.dylib
network.negotiate-auth.using-native-gsslib  user set
boolean   false
```

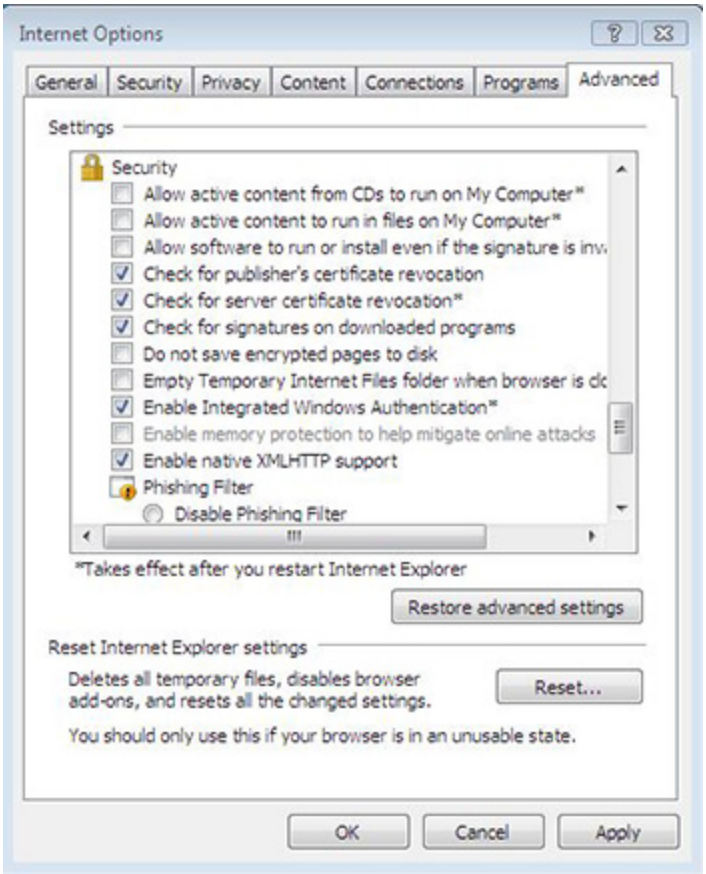


Configure Internet Explorer for SSO

Here is how to configure Internet Explorer 7.0 to use SPNEGO and Kerberos. The settings for other versions of IE might vary; see your browser's documentation for more information.

1. Start Internet Explorer 7.0.
2. On the **Tools** menu, click **Internet Options**.

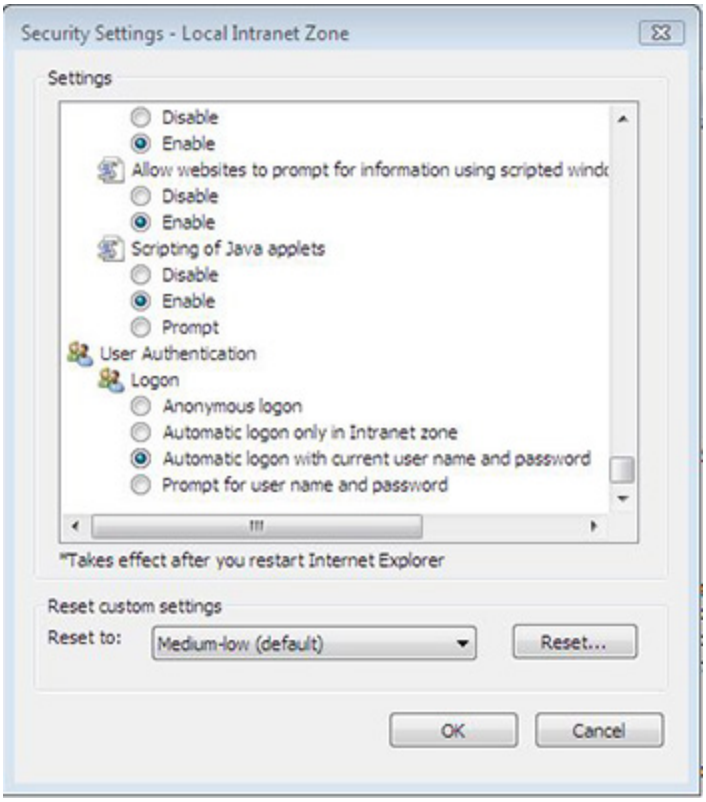
3. Click the **Advanced** tab and make sure that the **EnableIntegrated Windows Authentication** box is selected:



4. Click the **Security** tab.

5. Select a zone—for example, **Local intranet**—and then click **Custom level**.

6. In the **Settings** list, under **User Authentication**, click **Automatic logon with current user name and password** for a trusted site, or **Automatic logon only in Intranet zone** for a site you added to IE's list of Intranet sites. For more information, see your browser's documentation.



7. Return to the **Security** tab for **Internet Options** and set your web server as a trusted site.

8. Restart Internet Explorer.

Troubleshooting Kerberos Authentication

The following tools and procedures can help diagnose and resolve problems with Kerberos authentication when using the Apache HTTP Server for single sign-on (SSO).

Apache Log File

The location of the Apache error logs is specified in the Apache configuration file under the `ErrorLog` directive. Here is an example directive from `/etc/httpd/conf/httpd.conf` on RHEL 5: `ErrorLog logs/error_log`

Microsoft Kerbtray Utility

The Microsoft Kerbtray.exe utility, part of the Windows 2000 Resource Kit, can verify whether Internet Explorer obtained a Kerberos ticket for your web server. You can download the utility at the following URL:

http://www.microsoft.com/downloads/details.aspx?familyid=4E3A58BE-29F6-49F6-85BE-E866AF8E7A88

Klist Utility

You can use the klist utility in `/opt/pbis/bin/klist` to check the Kerberos keytab file on a Linux or Unix computer. The command shows all the service principal tickets contained in the keytab file so you can verify that the correct service principal names appear. Confirm that `HTTP/myserver@EXAMPLE.COM` and `HTTP/myserver.example.com@EXAMPLE.COM` appear in the list. It is normal to see multiple entries for the same name.

Example:

```
klist -k krb5_myserver.keytab
   Keytab name: FILE:krb5_myserver.keytab
   KVNO Principal
   ---- --------------------------------------
      6 HTTP/myserver@EXAMPLE.COM
      6 HTTP/myserver@EXAMPLE.COM
      6 HTTP/myserver@EXAMPLE.COM
      6 HTTP/myserver.example.com@EXAMPLE.COM
      6 HTTP/myserver.example.com@EXAMPLE.COM
      6 HTTP/myserver.example.com@EXAMPLE.COM
```

If your service principal names are incorrect, generate a new Kerberos keytab file.

Tip: Use an Alternate Kerberos Credentials Cache

Because you cannot store credentials for more than one principal in a Kerberos credentials cache at a time, you must maintain two or more credential caches by using the `KRB5CCNAME` environment variable and then switch to the cache that you want to use. To use an alternate Kerberos cache with PBIS, for example, you could execute the following sequence of commands as root:

```
[root@oracle1 ~]# KRB5CCNAME=/var/lib/pbis/krb5cc_
   lsass
[root@oracle1 ~]# export KRB5CCNAME
```

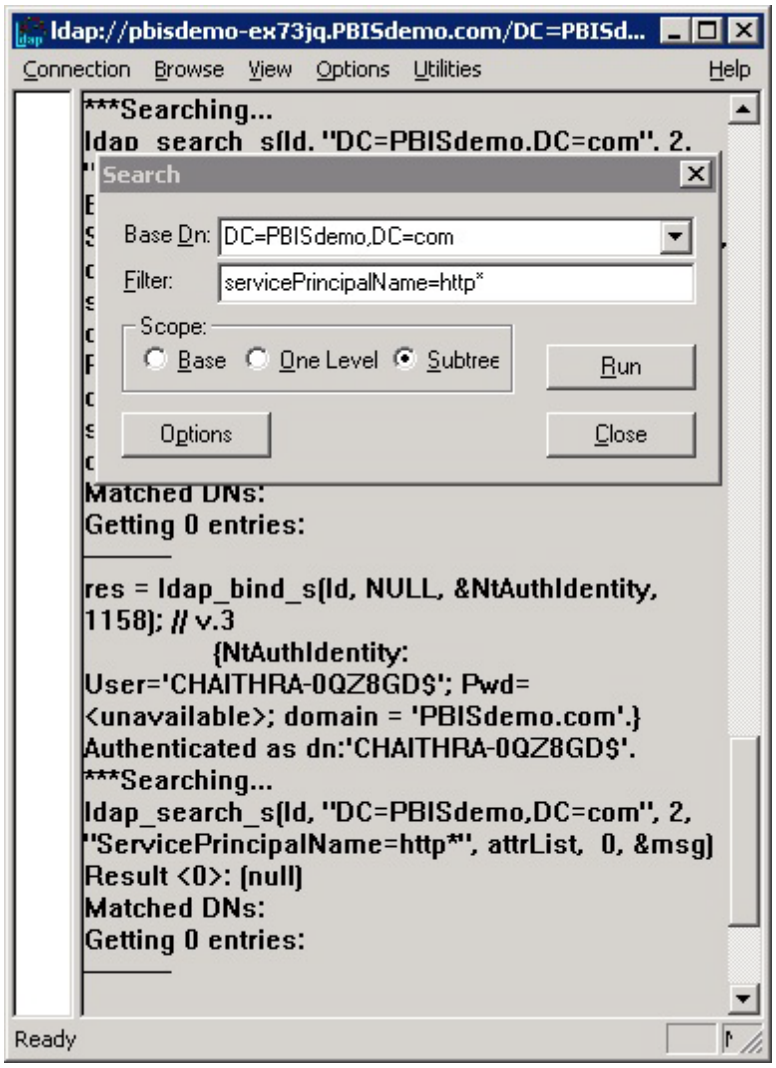
```
[root@oracle1 ~]# klist
Ticket cache: FILE:/var/lib/pbis/krb5cc_lsass
```

Resolving Common Problems

Authentication problems can be difficult to diagnose. First, check all the configuration parameters, including the validity of the keytab file. Second, make sure none of the common problems listed in the following table are sabotaging authentication.

Problem	Solution
The system's clock is out of sync.	The Kerberos standard requires that system clocks be no more than 5 minutes apart. Make sure that the system clocks on the Active Directory domain controller, the Linux or Unix web server, and the client are synchronized.
The user accessing the website is not on the `require` list	If Kerberos ticket was obtained on the client or the user correctly entered his credentials during the Basic Authentication prompt, it might be because authentication worked but the authorization failed. If so, the Apache error_log will contain a line like this: `access to / failed, reason: user EXAMPLE\\user not allowed access` Add the user to the `require user` directive or add the user's group to the `require group` directive.
The user accessing the website is logged on the wrong domain.	If the client user is logged on a domain different from the domain of the web server, one of two things will happen: 1. If the `KrbMethodK5Passwd` directive is set to `on`, or was not specified and thus defaults to `on`, the user will be prompted for credentials. 2. If `KrbMethodK5Passwd` is set to `off`, authentication will fail and the `Authorization Required` page will be displayed.
Internet Explorer does not consider the URL to be part of the Local Intranet zone or the Trusted sites.	This problem commonly occurs when the website is accessed by using a URL that includes the full domain name, such as `https://myserver.example.com`. Internet Explorer tries to obtain Kerberos tickets only for websites that are in the Local Intranet zone.

	Try to access the website by using only the server name, for example `https://myserver`. Or, you can add the URL to a list of Local Intranet sites or the trusted sites by changing your options in Internet Explorer.
The service principal name of the website is mapped to more than one object in the Active Directory.	Although this problem is rare, it is difficult to diagnose because the error messages are vague. The problem can occur after the ktpass utility was used repeatedly to generate a Kerberos keytab file for the web server. To check for this problem, log on your Active Directory domain controller and open the Event Viewer. Look for an event of type=Error, source=KDC, and event ID=11. The text of the event will be similar to the message below: `There are multiple accounts with name HTTP/myserver.example.com of type DS_SERVICE_PRINCIPAL_NAME.` To fix the problem, find the computer or user objects that were used to map the service principal name in Active Directory and then use the ADSI Edit to manually remove the "HTTP/myserver.example.com" string from the servicePrincipalName object property. Below the table is a screen shot that provides an example of how to find an object named `HTTP` by using Ldp:



Resolving Kerberos Library Mismatch

Because some operating systems, such as the 64-bit version of Red Hat Enterprise Linux 5, use an outdated version of `/lib/libcom_err.so`, the PBIS authentication agent cannot the locate the proper system library, leading to an error that looks like this:

```
httpd: Syntax error on line 202 of
/etc/httpd/conf/httpd.conf:
Cannot load /opt/pbis/apache/2.2/mod_auth_kerb.so into
server:
/opt/pbis/lib/libcom_err.so.3: symbol krb5int_strlcpy,
version
krb5support_0_MIT not defined in file libkrb5support.so.0
with link time reference
```

Solution: Force the `httpd` daemon to use the PBIS krb5 libraries by opening the startup script for the Apache HTTP Server—`/etc/init.d/httpd`—and adding the path to the PBIS Kerberos libraries on the line that starts Apache. The line that starts the daemon can vary by operating system. Example on a 64-bit system:

```
LD_LIBRARY_PATH=/opt/pbis/lib64 LANG=$HTTPD_LANG daemon
$httpd $OPTIONS
```

On a 32-bit system, the path would look like this:

```
/opt/pbis/lib
```

Note: This modification changes the version of the Kerberos libraries that are used by the Apache HTTP Server. The change might result in compatibility issues with other modules of Apache that use Kerberos.

Examples

To view sample code that shows you how to use PowerBroker Identity Services for single sign-on with protocols such as FTP and Telnet, see *Single Sign-On Examples* on the BeyondTrust website.

Configuring PBIS with the Registry

The PBIS registry is a hierarchical database that stores configuration information for PBIS services, authentication providers, drivers, and other services. On Linux, Unix, and Mac computers, the PBIS services continually access the registry to obtain settings for their parameters. The PBIS authentication service, for example, queries the registry to determine which log level to use or which home directory template to apply to a user. In version 5.4 or later, the registry replaces the text-based configuration files like `lsassd.conf` that were used in version 5.3 or earlier.

When you install the PBIS agent on a Linux, Unix, or Mac computer but do not install PBIS Enterprise on a Windows administrative workstation connected to Active Directory, you cannot configure local PBIS settings with Group Policy settings. Instead, you must edit the local PBIS registry. You can access the registry and modify its settings by using the PBIS registry shell—`regshell`—in `/opt/pbis/bin/`.

This chapter describes the structure of the registry, demonstrates how to change a value in it, and lists the local PBIS configuration options.

Note: Most of the registry settings can be centrally managed with Group Policy settings when you use PBIS Enterprise; see the *PowerBroker Identity Services Group Policy Administration Guide*. If you modify a setting in the registry that is managed by a Group Policy setting, the change will not persist: It will be overwritten by the setting in the Group Policy Object (GPO) as soon as the GPO is updated, which typically takes place once every 30 minutes. PBIS Open does not apply Group Policy settings.

The Structure of the Registry

The PBIS registry contains one predefined top-level, or root, key: `HKEY_THIS_MACHINE`. Within the root key, the structure of the registry is delineated by service into branches of keys, subkeys, and values. A key is similar to a folder; it can contain additional keys and one or more value entries. A value entry is an ordered pair with a name and a value. A subkey, similar to a subfolder, is simply a child key that appears under another key, the parent. A branch describes a key and all of its contents, including subkeys and value entries.

The upper level of the PBIS registry's hierarchical structure looks like the following:

```
\> ls
[HKEY_THIS_MACHINE]
```

```
\> cd HKEY_THIS_MACHINE\
HKEY_THIS_MACHINE\> ls

[HKEY_THIS_MACHINE\Services]

HKEY_THIS_MACHINE\> cd Services\
HKEY_THIS_MACHINE\Services> ls

[HKEY_THIS_MACHINE\Services\]
[HKEY_THIS_MACHINE\Services\dcerpc]
[HKEY_THIS_MACHINE\Services\eventlog]
[HKEY_THIS_MACHINE\Services\lsass]
[HKEY_THIS_MACHINE\Services\lwio]
[HKEY_THIS_MACHINE\Services\lwreg]
[HKEY_THIS_MACHINE\Services\netlogon]
[HKEY_THIS_MACHINE\Services\rdr]
```

Each of the services corresponds to a PBIS services or driver. The subkeys within each service contain value entries. A value specifies the setting for an entry, often presented under the `parameters` key.

Data Types

The PBIS registry employs four data types to store values. The values of data types are case sensitive. The following table lists the data types that are defined and used by PBIS. The maximum size of a key is 255 characters (absolute path).

Name	Data Type	Description
Binary Value	REG_ BINARY	A sequence of bytes. Displayed in the registry shell in hexadecimal format. The maximum size is 1024 bytes.
DWORD Value	REG_ DWORD	Data represented by a 32-bit integer. Parameters and services are typically set as this data type. The values are displayed in the registry shell in hexadecimal and decimal format. When a parameter is turned off, it is set to `0`; when a parameter is turned on, it is set to `1`.
Multi-String Value	REG_ MULTI_ SZ	A multiple string. Values that include lists or multiple values typically use this data type. Values are strings in quotation marks separated by spaces. In an import of a PBIS registry file, the multi-string values typically contain an `sza:` prefix. In an export of the registry, the multi-string values typically contain an `hex(7):` prefix. The maximum size of a `REG_MULTI_SZ` is 1024 bytes, total, not each string in the multi string. There are, however, null bytes between strings that contribute to the count, so the actual byte count is slightly less.
String Value	REG_SZ	A text string. The maximum size of a `REG_SZ` value is 1023 characters (1024 bytes, including the null terminator).

Modify Settings with the config Tool

To quickly change an end-user setting in the registry that is not managed by a Group Policy setting, you can run the config command-line tool as root:

/opt/pbis/bin/config

The syntax to change the value of a setting is as follows, where `setting` is replaced by the registry entry that you want to change and `value` by the new value that you want to set:

```
/opt/pbis/bin/config setting value
```

Here is an example of how to use config to change the `AssumeDefaultDomain` setting:

```
[root@rhel5d bin]# ./config --detail AssumeDefaultDomain
❶
Name: AssumeDefaultDomain
Description: Apply domain name prefix to account name at
logon
Type: boolean
Current Value: false
Accepted Values: true, false
Current Value is determined by local policy.

[root@rhel5d bin]# ./config AssumeDefaultDomain true ❷

[root@rhel5d bin]# ./config --show AssumeDefaultDomain ❸
boolean
true
local policy
```

❶ Use the `--detail` option to view the setting's current value and to determine the values that it accepts.

❷ Set the value to `true`.

❸ Use the `--show` option to confirm that the value was set to `true`.

To view the registry settings that you can change with config, execute the following command:

```
/opt/pbis/bin/config --list
```

You can also import and apply a number of settings with a single command by using the `--file` option combined with a text file that contains the settings that you want to change followed by the values that you want to set. Each setting-value pair must be on a single line. For example, the contents of my flat file, named `newRegistryValuesFile` and saved to the desktop of my Red Hat computer, looks like this:

```
AssumeDefaultDomain true
RequireMembershipOf "example\\support"
"example\\domain^admins"
HomeDirPrefix /home/ludwig
LoginShellTemplate /bash/sh
```

To import the file and automatically change the settings listed in the file to the new values, I would execute the following command as root:

```
/opt/pbis/bin/config --file
/root/Desktop/newRegistryValuesFile
```

Another Example

Here is another example of how to use config to find a setting and change it. Let's say you want to view the available trust settings because you know there are inaccessible trusts in your Active Directory network and you want to set PBIS to ignore all the trusts before you try to join a domain.

To do so, use `grep` with the `list` option:

/opt/pbis/bin/config --list | grep -i trust

The results will look something like this:

```
DomainManagerIgnoreAllTrusts
DomainManagerIncludeTrustsList
DomainManagerExcludeTrustsList
```

Next, use the `details` option to list the values that the `DomainManagerIgnoreAllTrusts` setting accepts:

```
[root@rhel5d bin]# ./config --details
DomainManagerIgnoreAllTrusts
Name: DomainManagerIgnoreAllTrusts
Description: When true, ignore all trusts during domain
enumeration.
Type: boolean
Current Value: false
Accepted Values: true, false
Current Value is determined by local policy.
```

Now change the setting to `true` so that PBIS will ignore trusts when you try to join a domain.

```
[root@rhel5d bin]# ./config DomainManagerIgnoreAllTrusts
true
```

Finally, check to make sure the change took effect:

```
[root@rhel5d bin]# ./config --show
DomainManagerIgnoreAllTrusts
boolean
true
local policy
```

In the example output that shows the setting's current values, `local policy` is listed—meaning that the policy is managed locally through `config` because a PBIS Group Policy setting is not managing the setting. You cannot locally modify a setting that is managed by a Group Policy setting.

For more information on the arguments of config, run the following command:

/opt/pbis/bin/config --help

Access the Registry

You can access and modify the registry by using the registry shell—`regshell`—in `/opt/pbis/bin`. The shell works in a way that is similar to BASH. You can navigate the registry's hierarchy with the following commands:

```
cd
ls
pwd
```

You can view a list of commands that you can execute in the shell by entering `help`:

```
/opt/pbis/bin/regshell
\> help
usage: regshell [--file | -f] command_file.txt
       add_key [[KeyName]]
       list_keys [[keyName]]
       delete_key [KeyName]
       delete_tree [KeyName]
```

```
        cd [KeyName]
        pwd
        add_value [[KeyName]] "ValueName" Type "Value"
["Value2"] [...]
        set_value [[KeyName]] "ValueName" "Value"
["Value2"] [...]
        list_values [[keyName]]
        delete_value [[KeyName]] "ValueName"
        set_hive HIVE_NAME
        import file.reg
        export [[keyName]] file.reg
        upgrade file.reg
        exit | quit | ^D

          Type: REG_SZ | REG_DWORD | REG_BINARY | REG_
MULTI_SZ
                REG_DWORD and REG_BINARY values are
hexadecimal
          Note: cd and pwd only function in interactive
mode
          Note: HKEY_THIS_MACHINE is the only supported
hive
\>
```

Note: In the unlikely event that you want to restore all the registry's default values, you must leave the domain, stop all the PBIS services, manually delete `/var/lib/pbis/db/registry.db`, and then reinstall PBIS.

Change a Registry Value by Using the Shell

You can change a value in the registry by executing the `set_value` command with the shell. The following procedure demonstrates how to change the value of the PAM key's `LogLevel` entry. The procedure to change other keys is similar. After you modify a registry setting for a PBIS service, you must refresh the corresponding service with the PBIS Service Manager for the changes to take effect.

1. With the root account, start `regshell`:

   `/opt/pbis/bin/regshell`

2. Change directories to the location of the PAM key and list its current settings:

```
[root@rhel5d bin]# ./regshell
\> cd HKEY_THIS_MACHINE\Services\lsass\Parameters\PAM
HKEY_THIS_MACHINE\Services\lsass\Parameters\PAM> ls

[HKEY_THIS_MACHINE\Services\lsass\Parameters\PAM\]
  "DisplayMotd"         REG_DWORD       0x00000001 (1)
  "LogLevel"            REG_SZ          "error"
  "UserNotAllowedError" REG_SZ          "Access
denied"
```

3. Execute the `set_value` command with the name of the value as the first argument and the new value as the second argument:

```
HKEY_THIS_MACHINE\services\lsass\Parameters\PAM> set_
value LogLevel debug
```

4. List the key's value entries to confirm that the value was changed:

```
HKEY_THIS_MACHINE\services\lsass\Parameters\PAM> ls

[HKEY_THIS_MACHINE\services\lsass\Parameters\PAM\]
  "DisplayMotd"         REG_DWORD       0x00000001 (1)
  "LogLevel"            REG_SZ          "debug"
  "UserNotAllowedError" REG_SZ          "Access
denied"
```

5. Exit the shell:

```
HKEY_THIS_MACHINE\Services\lsass\Parameters\PAM> quit
```

6. After you change a setting in the registry, you must use the PBIS Service Manager—`lwsm`—to force the service to begin using the new configuration. Because we changed a configuration of the lsass service, we must refresh it by executing the following command with super-user privileges:

   ```
   /opt/pbis/bin/lwsm refresh lsass
   ```

Set Common Options with the Registry Shell

This section shows you how to modify several common PBIS settings by using the registry shell: the default domain, the home directory, and the shell.

1. As root or with sudo, start the registry shell:

   ```
   /opt/pbis/bin/regshell
   ```

2. Change directories to the following location:

   ```
   cd HKEY_THIS_
   MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory
   ```

3. Change the shell to, for example, bash:

   ```
   set_value LoginShellTemplate /bin/bash
   ```

   For more information, see Set the Home Directory and Shell for Domain Users.

4. Set the option to use the default domain:

   ```
   set_value AssumeDefaultDomain 1
   ```

5. Leave the shell:

   ```
   quit
   ```

6. After you change a setting in the registry, you must use the PBIS Service Manager—`lwsm`—to force the service to begin using the new configuration. Because we changed a configuration of the lsass service, we must refresh it by executing the following command with super-user privileges:

   ```
   /opt/pbis/bin/lwsm refresh lsass
   ```

Here is how the string of commands looks in the registry shell:

```
[root@rhel5d docs]# /opt/pbis/bin/regshell
\> cd HKEY_THIS_
MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory
```

```
HKEY_THIS_
MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory>
 set_value AssumeDefaultDomain 1
HKEY_THIS_
MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory>
 set_value LoginShellTemplate /bin/bash
HKEY_THIS_
MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory>
 quit
[root@rhel5d docs]# /opt/pbis/bin/lwsm refresh lsass
```

Change a Registry Value from the Command Line

You can change a value in the registry by executing the `set_value` command from the command line. The following code block demonstrates how to change the value of the PAM key's `LogLevel` entry without using the shell. After you modify a registry setting for a PBIS service, you must refresh the corresponding service with the PBIS Service Manager for the changes to take effect.

```
/opt/pbis/bin/regshell ls '[HKEY_THIS_
MACHINE\Services\lsass\Parameters\PAM\]'
[HKEY_THIS_MACHINE\\Services\lsass\Parameters\PAM]
  "DisplayMotd"         REG_DWORD       0x00000001 (1)
  "LogLevel"            REG_SZ          "error"
  "UserNotAllowedError" REG_SZ          "Access denied"

/opt/pbis/bin/regshell set_value '[HKEY_THIS_
MACHINE\Services\lsass\Parameters\PAM\]' LogLevel debug

/opt/pbis/bin/regshell ls '[HKEY_THIS_
MACHINE\Services\lsass\Parameters\PAM\]'
[HKEY_THIS_MACHINE\\Services\lsass\Parameters\PAM]
  "DisplayMotd"         REG_DWORD       0x00000001 (1)
  "LogLevel"            REG_SZ          "debug"
  "UserNotAllowedError" REG_SZ          "Access denied"
```

Find a Registry Setting

When you're unsure where to find a setting that you want to change, you can export the registry's structure to a file and then search the file for the value entry's location.

Important: You must export the registry as root.

1. With the root account, start `regshell`:

   `/opt/pbis/bin/regshell`

2. In the shell, execute the `export` command with the root key as the first argument and a target file as the second argument:

   ```
   export HKEY_THIS_MACHINE\ lwregistry.reg
   ```

   The file is exported to your current directory unless you specify a path.

   In a text editor such as vi, open the file to which you exported the registry and search for the entry that you are want to find.

lsass Settings

This section lists values in the lsass branch of the registry.

Log Level Value Entries

There is a `LogLevel` value entry under several keys, including `lsass/Parameters` and `PAM`. Although the default value is typically `error`, you can change it to any of the following values: disabled, error, warning, info, verbose.

Locations

[HKEY_THIS_MACHINE\Services\lsass\Parameters]

[HKEY_THIS_MACHINE\Services\lsass\Parameters\PAM]

Value Entry

LogLevel

Example with default value:

```
"LogLevel"="error"
```

Turn on Event Logging

You can capture information about authentication transactions, authorization requests, and other security events by turning on event logging. For information about managing and viewing events, see Monitoring Events with the Event Log.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters]

Value Entry

EnableEventlog

Example with default value:

```
"EnableEventlog"=dword:00000000
```

Turn off Network Event Logging

After you turn on event logging, network connection events are logged by default. On laptop computers, computers with a wireless connection, or other computers whose network status might be in flux, you can turn off event logging so that the event log is not inundated with connectivity events.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters]

Value Entry

LogNetworkConnectionEvents

Example with default value:

```
"LogNetworkConnectionEvents"=dword:00000001
```

Restrict Logon Rights

You can require that a user be a member of a group to log on a computer, or you can limit logon to only the users that you specify. PBIS checks `require_membership_of` information in both the authentication phase and the account phase.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

RequireMembershipOf

Notes

Add each user or group to the value entry by using an NT4-style name (the short domain name with the group name) or an Active Directory security identifier (SID). Aliases are not supported. The entries must be in the form of a list of quoted entries: Each entry must be enclosed in quotation marks. A slash character must be escaped by being preceded by a slash. Example:

```
"RequireMembershipOf"="example\\support"
"example\\domain^admins" "example\\joe" "S-1-5-21-
3447809367-3151979076-456401374-513"
```

Only the users that you specify and the users who are members of the groups that you specify are allowed to log on the computer.

Display an Error to Users Without Access Rights

You can set PBIS to display an error message when a user attempts to log on a computer without the right to access it.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\PAM]

Value Entry

UserNotAllowedError

Notes

Add the text of the error message that you want to display to the value of the entry. Example with default value:

```
"UserNotAllowedError"="Access denied"
```

Display a Message of the Day

You can set PBIS to display a message of the day (MOTD). It appears after a user logs on but before the logon script executes to give users information about a computer. The message can, for instance, remind users of the next scheduled maintenance window.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location in registry:

```
[HKEY_THIS_MACHINE\Services\lsass\Parameters\PAM]
```

Value Entry

DisplayMotd

Example with the value set to `1`, or true, to display a message:

```
"DisplayMotd"=dword:00000001
```

Change the Domain Separator Character

The default domain separator character is set to \. So, by default, the Active Directory group `DOMAIN\Administrators` appears as `DOMAIN\administrators` on target Linux and Unix computers. The PBIS authentication service renders all names of Active Directory users and groups lowercase.

You can, however, replace the slash that acts as the separator between an Active Directory domain name and the SAM account name with a character that you choose by modifying the `DomainSeparator` value entry in the registry.

The following characters cannot be used as the separator:

- alphanumeric characters (letters and digits)
- @
- #
- And not the character that you used for the `space-replacement` setting; for more information, see Change the Replacement Character for Spaces.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters]

Value Entry

DomainSeparator

Example entry with default value:

```
"DomainSeparator"="\\"
```

Note: In the default value, the slash character is escaped by the slash that precedes it.

Change Replacement Character for Spaces

The default replacement character is set to ^. So, by default, the Active Directory group `DOMAIN\Domain Users` appears as `DOMAIN\domain^users` on target Linux and Unix computers. You can, however, replace the spaces in Active Directory user and group names with a character that you choose by editing the `SpaceReplacement` value entry in the registry.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters]

Value Entry

SpaceReplacement

Example with default value:

```
"SpaceReplacement"="^"
```

Notes

The following characters cannot be used as the separator:

- whitespace - spaces and tabs
- alphanumeric characters - letters and digits
- @
- \
- #

The PBIS authentication service renders all names of Active Directory users and groups lowercase.

Turn Off System Time Synchronization

With PBIS Open and PBIS Enterprise, you can specify whether a joined computer synchronizes its time with that of the domain controller. By default, when a computer is joined to a domain without using the `notimesync` command-line option, the computer's time is synchronized with the domain controller's when there is a difference of more than 60 seconds but less than the maximum clock skew, which is typically 5 minutes.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

SyncSystemTime

Example with default value:

```
"SyncSystemTime"=dword:00000001
```

Set the Default Domain

If your Active Directory environment has only one domain, you can set PBIS to assume the default domain, liberating users from typing the domain name before their user or group name each time they log on a computer or switch users.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

AssumeDefaultDomain

Example with default value:

```
"AssumeDefaultDomain"=dword:00000000
```

Set the Home Directory and Shell for Domain Users

When you install PowerBroker Identity Services on a Linux, Unix, or Mac computer but not on Active Directory, you cannot associate a PowerBroker cell with an organizational unit, and thus you have no way to define a home directory or shell in Active Directory for users who log on the computer with their domain credentials. To set the home directory and shell for a Linux, Unix, or Mac computer that is using PBIS Open or PBIS Enterprise without cell, edit the value entry in registry.

If you use PBIS Enterprise to set the shell and home directory both in Active Directory and in the registry, the settings in Active Directory take precedence.

After you change the home directory or shell in the registry, you must clear the PBIS authentication cache, log off, and then log on before your changes will take effect.

In the lsass branch, there are two keys that contain value entries for the home directory and shell. One is for the local provider, the other is for the Active Directory provider. Locations:

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\Local]

The following value entries for the home directory and shell, shown with their default settings, appear under both the `Active Directory` and `Local` provider keys:

```
"LoginShellTemplate"="/bin/sh"
"HomeDirTemplate"="%H/local/%D/%U"
"HomeDirPrefix"="/home"
"CreateHomeDir"=dword:00000001
```

Set the Shell

Under the key for a provider, modify the value of the following entry to set the shell that you want:

`LoginShellTemplate`

Example with default value:

`"LoginShellTemplate"="/bin/sh"`

Note: `/bin/bash` might not be available on all systems.

Set the Home Directory

You can modify the `HomeDirTemplate` value entry to set the home directory that you want by using these variables:

Variable	Description
`%U`	The default user name. It is required.
`%D`	The default domain name. It is optional.
`%H`	The default home directory. It is optional. If used, it must be set as an absolute path. This value, if used, is typically the first variable in the sequence.
`%L`	The hostname of the computer. It is optional.

Here is an example with all four variables set: `%H/%L/%D/%U`

Example with default value:

`"HomeDirTemplate"="%H/local/%D/%U"`

In the example above, the `HomeDirTemplate` is using the `%H` variable for the `HomeDirPrefix` to set the user's home directory. In the example, the `HomeDirPrefix` is not preceded by a slash because the slash is included in the default `HomeDirPrefix` to ensure that the path is absolute. By default, the `%H` variable automatically changes to be compatible with the operating system to generate a home directory path. On Solaris, for example, the `%H` variable maps to `/export/home`. On Mac OS X it maps to `/Users`; on Linux, it maps to `/home`.

Optionally, you can set the `HomeDirPrefix` by changing the prefix to the path that you want. However, the `HomeDirPrefix` must be an absolute path—so you must precede it with a slash. Example with default value:

```
"HomeDirPrefix"="/home"
```

You must use the default user name variable (`%U`). You may specify the default domain name by using the domain name variable (`%D`), but it is not required.

All the users who log on the computer by using their Active Directory domain credentials will have the shell and home directory that you set under the `Providers\ActiveDirectory` key. All the users who log on the computer by using their local PBIS provider credentials will have the shell and home directory that you set under the `Providers\Local` key.

Important: On Solaris, you cannot create a local home directory in `/home`, because `/home` is used by autofs, Sun's automatic mounting service. The standard on Solaris is to create local home directories in `/export/home`.

On Mac OS X, to mount a remote home directory, you must first create the directory on the remote server as well as the folders for music, movies, and so forth. See Use the createhomedir Command to Create Home Directories and other information on Apple's website.

Turn Off Home Directories

By default, a user's home directory is created upon logon. To turn off the creation of home directories, change value of the following entry to `0`, for false:

```
CreateHomeDir
```

Example with default setting of `1`, which creates a home directory:

```
"CreateHomeDir"=dword:00000001
```

See Also

Fix the Shell and Home Directory Paths

Set the Umask for Home Directories

PBIS presets the umask for the home directory and all the files in it to `022`. With a umask value of `022`, the default file permissions for your AD user account are as follows: Read-write access for files and read-write-search for directories you own. All others have read access only to your files and read-search access to your directories. You can, however, set the umask for home directories by modifying its value entry in the registry.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Locations

[HKEY_THIS_
MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\Local]

Value Entry

HomeDirUmask

Example with default value:

`"HomeDirUmask"="022"`

Set the Skeleton Directory

By default, PBIS adds the contents of `/etc/skel` to the home directory created for a new user account on Linux and Unix computers. Using `/etc/skel` or a directory that you designate ensures that all users begin with the same settings or environment.

On Mac OS X computers, the default skeleton directory is as follows:

```
System/Library/User Template/Non_localized,
/System/Library/User Template/English.lproj
```

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Locations

[HKEY_THIS_
MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\Local]

Value Entry

SkeletonDirs

Example with default value:

`"SkeletonDirs"="/etc/skel"`

Note: Add the skeleton directory that you want to set to the entry. You can add multiple entries, but each entry must be enclosed in quotation marks and separated by a space.

Force PBIS Enterprise to Work Without Cell Information

To use the PBIS Enterprise agent to join a Linux, Unix, or Mac OS X computer to a domain that has not been configured with cell information, you must change the value of `CellSupport` to `unprovisioned`. This setting, which applies only to PBIS Enterprise, forces the authentication service to ignore the following Unix information even though it is set in Active Directory:

- Home directory
- UID
- GID
- Unix shell

Instead of using the information from Active Directory, the `unprovisioned` value sets the authentication service to hash the user's security identifier and use local settings for the Unix shell and the home directory.

Location

[HKEY_THIS_ MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

CellSupport

Notes

The value must be set as one of the following: `no-unprovisioned`, `full` or `unprovisioned`.

The default is `no-unprovisioned`, a setting that requires you to create a cell in Active Directory before you join a PBIS client to it. If you are using PBIS Enterprise with cells and you want to use the Unix settings in AD, it is recommended that you leave `cell-support` set to its default value of `no-unprovisioned`:

"CellSupport"="no-unprovisioned"

Here is an example with the value set to `unprovisioned` to force PBIS Enterprise to ignore Unix settings and other cell information in AD:

"CellSupport"="unprovisioned"

Setting the value to `full` configures the PBIS Enterprise agent to use cell information when it appears in AD and local settings when no cells are in AD:

"CellSupport"="full"

Refresh User Credentials

By default, PBIS automatically refreshes user credentials, but you can turn off automatic refreshes by modifying the configuration of the PBIS authentication service.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

RefreshUserCredentials

Example with default setting:

"RefreshUserCredentials"=dword:00000001

Turn Off K5Logon File Creation

By default, PBIS creates a .k5login file in the home directory of an Active Directory user who is authenticated by Kerberos when logging on a Linux, Unix, or Mac OS X computer. You can, however, stop the creation of a .k5login file.

The .k5login file contains the user's Kerberos principal, which uniquely identifies the user within the Kerberos authentication protocol. Kerberos can use the .k5login file to check whether a principal is allowed to log on as a user. A .k5login file is useful when your computers and your users are in different Kerberos realms or different Active Directory domains, which can occur when you use Active Directory trusts.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

CreateK5Login

Example with default value:

```
"CreateK5Login"=dword:00000001
```

Change the Duration of the Computer Password

You can set the computer account password's expiration time. The expiration time specifies when a computer account password is reset in Active Directory if the account is not used. The default is 30 days.

Active Directory handles computer accounts for Linux, Unix, and Mac in the same way as those for Windows computers; for more information, see the Microsoft Active Directory documentation.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

MachinePasswordLifespan

Example with default value, which is shown as seconds in hexadecimal format:

```
"MachinePasswordLifespan"=dword:000927c0
```

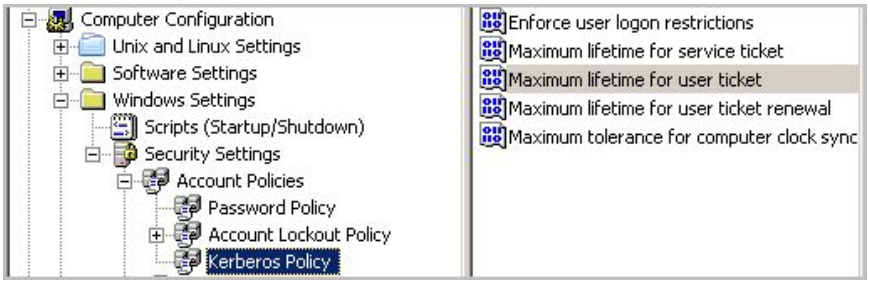
Notes

Setting the value to `0` disables expiration. The minimum value is 1 hour, expressed in seconds, and the maximum is 60 days, expressed in seconds. To avoid issues with Kerberos key tables and single sign-on, the `MachinePasswordLifespan` must be at least twice the maximum lifetime for user tickets, plus a little more time to account for the permitted clock skew. The expiration time for a user ticket is set by using an Active Directory Group Policy setting called **Maximum lifetime for user ticket**. The default user ticket lifetime is 10 hours; the default PBIS computer password lifetime is 30 days.

Check the Maximum Lifetime for a User Ticket

1. Open the default domain policy in the Group Policy Management Editor.

2. In the console tree under **Computer Configuration**, expand **Windows Settings**, expand **Security Settings**, expand **Account Policies**, and then click **Kerberos policy**.



3. In the details pane, double-click **Maximum lifetime for user ticket**.

4. In the **Ticket expires in** box, make sure that the number of hours is no more than half that of the `MachinePasswordLifespan` you set in the registry.

See Also

Fix a Key Table Entry-Ticket Mismatch

Sign and Seal LDAP Traffic

You can sign and seal LDAP traffic to certify it and to encrypt it so that others cannot see your LDAP traffic on your network. This setting can help improve network security.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

LdapSignAndSeal

Example with default value:

```
"LdapSignAndSeal"=dword:00000000
```

NTLM Settings

There are a number of NTLM settings that system administrators can use to manage NTLM sessions.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\Local]

Value Entry with Default Values

```
"AcceptNTLMv1"=dword:00000001
```

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\NTLM]

Value Entries with Default Values

```
"SendNTLMv2"=dword:00000000
"Support128bit"=dword:00000001
"Support56bit"=dword:00000001
"SupportKeyExchange"=dword:00000001
"SupportNTLM2SessionSecurity"=dword:00000001
"SupportUnicode"=dword:00000001
```

Each NTLM value entry is described in the following table. For additional information, see Microsoft's description of the LAN Manager authentication levels.

Value Entry	Description
AcceptNTLMv1	Controls whether the PBIS local provider accepts the older and less secure NTLM protocol for authentication in addition to NTLMv2. This setting does not apply to the Active Directory provider because it passes off NTLM and NTLMv2 authentication to a domain controller through schannel; it is the domain controller's settings that determine which versions of NTLM are allowed.
SendNTLMv2	Forces `lsass` to use NTLMv2 rather than the older and less secure NTLM when `lsass` acts as a client. (`Lsass` typically serves as an NTLM client in relation to domain controllers.)
Support128bit and Support56bit	Control the length of the encryption key. They are intended to serve as a mechanism for debugging NTLM sessions. There are no corresponding settings in Windows.
SupportKeyExchange	Allows the protocol to exchange a session key—Kerberos has a similar feature. During authentication, an alternate key is exchanged for subsequent encryption to reduce the risk of exposing a password. It is recommended that you use the default setting.
SupportNTLM2SessionSecurity	Permits the client to use a more secure variation of the protocol if the client discovers that the server supports it. Corresponds to a similar setting in Windows.
SupportUnicode	Sets NTLM to represent text according to the Unicode industry standard. It is recommended that you use the default setting—which is to support Unicode.

Additional Subkeys

There are additional subkeys in the lsass branch that the lsass service uses to store information for the PBIS application. It is recommended that you do not change these subkeys or their value entries.

- [HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory\DomainJoin\YourD

  Stores information about domain trusts.

- [HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory\DomainJoin\YourD

  Stores data used by the Active Directory authentication provider.

- [HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory\DomainJoin\YourD

  Caches information about the computer and the user's Active Directory account, including the computer password. The computer password is visible only to root users when they view or export the registry.

- [HKEY_THIS_MACHINE\Services\lsass\Parameters\RPCServers]

  Stores information that the system uses to execute remote procedure calls.

Add Domain

This value entry controls whether the domain-join process adds domain groups to the local PBIS groups and whether the domain-leave process removes domain groups from the local PBIS groups. The default setting is 0, for disabled—no domain groups are added to local groups.

When the setting is enabled, the AD group `Domain Admins` is added to `BUILTIN\\Administrators`, and `Domain Users` is added to `BUILTIN\\Users`.

After joining or leaving a domain, you can verify that the domain groups were added to or removed from the local groups by running the `lsa enum-members` command for the `BUILTIN\\Administrators` group and the `BUILTIN\\Users` group.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

AddDomainToLocalGroupsEnabled

Control Trust Enumeration

PBIS includes the following settings for controlling how the domain manager component of the authentication service enumerates trusts. The settings can help improve performance of the authentication service in an extended AD topology.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Important: The setting that specifies an include list is dependent on defining the setting for ignoring all trusts: To use the include list, you must first enable the setting to ignore all trusts. The include-list setting must explicitly contain every domain that you want to enumerate. It is insufficient to include only the forests that contain the domains.

For a domain that is added to the include list, PBIS tries to discover its trust. If some of the domains are not included in the space-separated list, the resulting trust relationships might run counter to your intentions: The PBIS agent might process the trust as a one-way forest child trust when it is not.

Changes to the trust enumeration settings take effect when you restart either the computer or the PBIS authentication service (`lsass`).

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entries

Value Entry	Description
DomainManagerIgnoreAllTrusts	Determines whether the authentication service discovers domain trusts. In the default configuration of disabled, the service enumerates all the parent and child domains as well as forest trusts to other domains. For each domain, the service establishes a preferred domain controller by checking for site affinity and testing server responsiveness, a process that can be slowed by WAN links, subnet firewall blocks, stale AD site topology data, or invalid DNS information.

	When it is unnecessary to enumerate all the trusts—because, for example, the intended users of the target computer are only from the forest that the computer is joined to—turning on this setting can improve startup times of the authentication service.
DomainManagerIncludeTrustsList	When the setting `DomainManagerIgnoreAllTrusts` is turned on, only the domain names in the space-separated include list are enumerated for trusts and checked for server availability. Each item in the list must be separated by a space.
DomainManagerExcludeTrustsList	When the setting `DomainManagerIgnoreAllTrusts` is turned off (its default setting), the domain names in the space-separated exclude list are not enumerated for trusts and not checked for server availability. Each item in the list must be separated by a space.

Modify Smart Card Settings

The following settings are available only with PBIS Enterprise.

Location in registry:

`[HKEY_THIS_MACHINE\Services\lsass\Parameters\PAM]`

Value Entries

SmartCardPromptGecos

SmartCardServices

Set the Interval for Checking the Status of a Domain

This value entry determines how frequently the PBIS domain manager checks whether a domain is online. The default is 5 minutes.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

DomainManagerCheckDomainOnlineInterval

Example with default value:

```
"DomainManagerCheckDomainOnlineInterval"=dword:0000012c
```

Set the Interval for Caching an Unknown Domain

This value entry determines how long the PBIS domain manager caches an unknown domain as unknown. The default is 1 hour.

Location

[HKEY_THIS_ MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

DomainManagerUnknownDomainCacheTimeout

Example with default value:

```
"DomainManagerUnknownDomainCacheTimeout"=dword:00000e10
```

lsass Cache Settings

Many of the following cache settings can be managed by the Group Policy settings of PBIS Enterprise. For more information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Set the Cache Type

By default, the lsass service uses SQLite to cache information about users, groups, and the state of the computer. You can, however, change the cache to store the information in memory, which might improve the performance of your system.

Location

[HKEY_THIS_ MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

CacheType

Example with default value:

```
"CacheType"="sqlite"
```

Notes

To use the memory cache, change the value to `memory`. Example:

```
"CacheType"="memory"
```

Cap the Size of the Memory Cache

By default, the lsass service caches information about users, groups, and the state of the computer in a SQLite database. If, however, you change the cache to store the data in memory, you can limit the size of the cache to prevent it from consuming too much memory. It is suggested that the size of the cache be between 1 MB and 10 MB, but the size limit that you choose will depend on your environment. Groups with many members call for a larger memory cache to enumerate all the users.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

MemoryCacheSizeCap

Example with default value:

```
"MemoryCacheSizeCap"=dword:00000000
```

Notes

To limit the memory cache to a maximum value, change the value to the byte count that you want. When the total cache size exceeds the limit, old data is purged. The default value is `0`: no limit is set.

Change the Duration of Cached Credentials

You can specify how long the PBIS agent caches information about an Active Directory user's home directory, logon shell, and the mapping between the user or group and its security identifier (SID). This setting can improve the performance of your system by increasing the expiration time of the cache.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entry

CacheEntryExpiry

Example with default value:

"CacheEntryExpiry"=dword:00003840

Note: Set the value to an interval, in seconds. The minimum entry is 0 seconds and the maximum is 1 day, expressed in seconds.

Change NSS Membership and NSS Cache Settings

To customize PBIS to meet the performance needs of your network, you can specify how the PBIS agent parses and caches group and user membership information with the following value entries in the registry:

Location

[HKEY_THIS_MACHINE\Services\lsass\Parameters\Providers\ActiveDirectory]

Value Entries

Here are the value entries with their default values:

```
"TrimUserMembership"=dword:00000001
"NssGroupMembersQueryCacheOnly"=dword:00000001
"NssUserMembershipQueryCacheOnly"=dword:00000000
"NssEnumerationEnabled"=dword:00000000
```

Each setting is described in the table that follows.

Setting	Description
`TrimUserMembership`	Specifies whether to discard cached information from a Privilege Attribute Certificate (PAC) entry when it conflicts with new information retrieved through LDAP. Otherwise, PAC information, which does not expire, is updated the next time the user logs on. The default setting is `1`: It is turned on.
`NssGroupMembersQueryCacheOnly`	Specifies whether to return only cached information for the members of a group when queried through nsswitch. More specifically, the setting determines whether nsswitch-based group APIs obtain group membership information exclusively from the cache, or whether they search for additional group membership data through LDAP.

	This setting is made available because, with large amounts of data, the LDAP enumeration can be slow and can affect performance. To improve performance for groups with more than 10,000 users, set this option to `yes`. Without the LDAP enumeration, only when a user logs on can that user's complete group membership be retrieved based on the PAC. The default setting is `1`: It is turned on.
`NssUserMembershipQueryCacheOnly`	When set to `yes`, enumerates the groups to which a user belongs using information based solely on the cache. When set to `no`, it checks the cache and searches for more information over LDAP. The default setting is `0`: It is turned off.
`NssEnumerationEnabled`	Controls whether all users or all groups can be incrementally listed through NSS. On Linux computers and Unix computers other than Mac, the default setting is `0`, or turned off. On Mac OS X computers, the default setting is `1`, or turned on. To allow third-party software show Active Directory users and groups in lists, you can change this setting to `1`, but performance might be affected.

Note: When you run the `id` command for an Active Directory user other than the current user on some Linux systems, such as SLES 10 and SLED 10, the command returns only that user's primary group. The command enumerates all the groups and searches for the user in the groups' membership. To properly find another user's membership with the `id` command on SLES 10 and SLED 10, you must turn on NSS enumeration.

eventlog Settings

This section lists values in the eventlog branch of the registry.

Allow Users and Groups to Delete Events

This entry specifies the Active Directory users and groups who can delete events from the PBIS event log.

Location

[HKEY_THIS_MACHINE\Services\eventlog\Parameters]

Value Entry

AllowDeleteTo

Notes

Add the users and groups, separated by commas, to the value entry by using NT4-style names (the short domain name with the group name), the user's or group's alias, or an Active Directory security identifier (SID). The comma-separated list must be enclosed in quotation marks. Example:

```
AllowDeleteTo="example\support, example\domain^admins,
example\joe, jane, S-1-5-21-3447809367-3151979076-456401374-
513, sales^admins"
```

Allow Users and Groups to Read Events

This value entry specifies the Active Directory users and groups who can read events in the PBIS event log.

Location

[HKEY_THIS_MACHINE\Services\eventlog\Parameters]

Value Entry

AllowReadTo

Notes

Add the users and groups, separated by commas, to the value entry by using NT4-style names (the short domain name with the group name), the user's or group's alias, or an Active Directory security identifier (SID). The comma-separated list must be enclosed in quotation marks. Example:

```
AllowReadTo="example\support, example\domain^admins,
example\joe, jane, S-1-5-21-3447809367-3151979076-456401374-
513, sales^admins"
```

Allow Users and Groups to Write Events

This value entry specifies the Active Directory users and groups who can write events in the PBIS event log.

Location

[HKEY_THIS_MACHINE\Services\eventlog\Parameters]

Value Entry

AllowWriteTo

Notes

Add the users and groups, separated by commas, to the value entry by using NT4-style names (the short domain name with the group name), the user's or group's alias, or an Active Directory security identifier (SID). The comma-separated list must be enclosed in quotation marks. Example:

```
AllowWriteTo="example\support, example\domain^admins,
example\joe, jane, S-1-5-21-3447809367-3151979076-456401374-
513, sales^admins"
```

Set the Maximum Disk Size

This value entry specifies the maximum size of the event log. The default is 512 KB. The minimum size is 64 KB. The maximum is 419424 KB.

Location

[HKEY_THIS_MACHINE\Services\eventlog\Parameters]

Value Entry

MaxDiskUsage

Example with default value:

```
"MaxDiskUsage"=dword:06400000
```

Set the Maximum Number of Events

This value entry defines the maximum number of events that can reside in the event log. The default is 100,000. The minimum number is 100. The maximum is 2,000,000.

Location

[HKEY_THIS_MACHINE\Services\eventlog\Parameters]

Value Entry

MaxNumEvents

Example with default value:

```
"MaxNumEvents"=dword:000186a0
```

Set the Maximum Event Timespan

This value entry defines maximum length of time, in days, that events can remain in the event log. Events older than the specified time span are removed. The default is 90 days. The maximum is 365 days.

Location

[HKEY_THIS_MACHINE\Services\eventlog\Parameters]

Value Entry

MaxEventLifespan

Example with the default value of 90 days:

```
"MaxEventLifespan"=dword:0000005a
```

Change the Purge Interval

This value entry defines the number of days after which to purge the database of events. The default is 1 day.

Location

[HKEY_THIS_MACHINE\Services\eventlog\Parameters]

Value Entry

EventDbPurgeInterval

Example with default value of 1 day:

```
"EventDbPurgeInterval"=dword:00000001
```

netlogon Settings

The `netlogon` branch contains registry values for setting the expiration of the cache that holds information for the site affinity service, including the optimal domain controller and global catalog. The `netlogon` service generates the value entries under the `[HKEY_THIS_MACHINE\Services\netlogon\cachedb]` subkey to cache information about your domain controllers and global catalog. It is recommended that you do not change the registry values under the `cachedb` subkey.

```
[HKEY_THIS_MACHINE\Services\netlogon]
"Arguments"="/opt/pbis/sbin/netlogond --syslog"
"Dependencies"="lwreg"
"Description"="Likewise Site Affinity Service"
"Path"="/opt/pbis/sbin/netlogond"
"Type"=dword:00000001

[HKEY_THIS_MACHINE\Services\netlogon\cachedb]

[HKEY_THIS_MACHINE\Services\netlogon\Parameters]
"NegativeCacheTimeout"=dword:0000003c
"PingAgainTimeout"=dword:00000384
"WritableRediscoveryTimeout"=dword:00000708
"WritableTimestampMinimumChange"=dword:00000000
```

Only the values under the `Parameters` subkey are documented in this section.

Set the Negative Cache Timeout

This setting is reserved for internal use only.

Location

[HKEY_THIS_MACHINE\Services\netlogon\Parameters]

Value Entry

NegativeCacheTimeout

Example with default value:

`"NegativeCacheTimeout"=dword:0000003c`

Set the Ping Again Timeout

The netlogon service periodically tests whether cached domain controllers are available. This setting controls how often it does so.

Location

[HKEY_THIS_MACHINE\Services\netlogon\Parameters]

Value Entry

PingAgainTimeout

Example with default value:

```
"PingAgainTimeout"=dword:00000384
```

Set the Writable Rediscovery Timeout

When a service requests a writable domain controller and one does not exist in the local site, this setting controls how long the service stays affinitized to the writable domain controller before reaffinitizing to a closer read-only domain controller.

Location

[HKEY_THIS_MACHINE\Services\netlogon\Parameters]

Value Entry

WritableRediscoveryTimeout

Example with default value:

```
"WritableRediscoveryTimeout"=dword:00000708
```

Set the Writable Timestamp Minimum Change

Netlogon keeps track of when a writable domain controller was last requested. Related to `WritableDiscoveryTimeout`, this setting controls how often that timestamp is changed.

Location

[HKEY_THIS_MACHINE\Services\netlogon\Parameters]

Value Entry

WritableTimestampMinimumChange

Example with default value:

```
"WritableTimestampMinimumChange"=dword:00000000
```

Set CLdap Options

The netlogon service uses multiple asynchronous CLDAP searches in a single thread to find servers that act as domain controllers and global catalogs. To improve performance in the context of your unique network, you can adjust the following settings for the Connection-less Lightweight Directory Access Protocol.

Location

[HKEY_THIS_MACHINE\Services\netlogon\Parameters]

Value Entries

`CLdapMaximumConnections` is the maximum number of servers that will be pinged simultaneously. The default is 100.

`CLdapSearchTimeout` is the timeout for the entire search (in seconds). The default is 15 seconds.

`CLdapSingleConnectionTimeout` is the timeout for pinging a single server (in seconds). The default is 15 seconds.

lwio Settings

The `lwio` branch contains registry settings for the input-output service, lwio.

The settings under the `shares` subkey define shared folders and the security descriptors that control access to them. It is recommended that you do not directly change the values under the `shares` subkey while the lwio service is running.

Sign Messages If Supported

Although signing messages is turned off by default, you can set the input-output service to sign messages. Doing so, however, can degrade performance. When signing is turned off, the input-output service will reject clients that require signing.

Location

[HKEY_THIS_MACHINE\Services\lwio\Parameters\Drivers\rdr]

Value Entry

SignMessagesIfSupported

Example with default value:

`"SignMessagesIfSupported"=dword:00000000`

Lwedsplugin Settings for Mac Computers

The PBIS registry includes the following settings to manage the directory services plugin on a Mac OS X computer.

Note: With PBIS Enterprise, you can manage this feature by using a PBIS Group Policy setting. For information, see the *PowerBroker Identity Services Group Policy Administration Guide*.

Here is an example configuration in the registry:

```
[HKEY_THIS_MACHINE\Services\lwedsplugin\Parameters\]
  "AllowAdministrationBy"          REG_SZ
```

```
"CORP\\EnterpriseTeam"
  "EnableForceHomedirOnStartupDisk" REG_DWORD
0x00000001 (1)
  "EnableMergeAdmins"               REG_DWORD
0x00000001 (1)
  "UncProtocolForHomeLocation"      REG_SZ          "smb"
  "UseADUncForHomeLocation"         REG_DWORD
0x00000001 (1)
```

Each setting is described in the following table.

DS Plugin Setting in the Registry	Description
Allow administration by	Specifies the administrators included the local admin group (GID: 80) on the computer. The setting can specify Active Directory users or groups. Local entries are overwritten unless you also set the parameter to merge administrators who are defined locally.
Force home directory on startup disk	Sets a computer to use a local home directory path. When a user with a home folder connection defined in Active Directory logs on, the connection is created in the dock under `/Network/Servers/homeFolderName`.
Merge Administrators	Preserves members of the admin group who are defined locally but are not specified in the `allow administration by` policy.
Set the UNC Protocol for the Home Location	Sets the protocol for the home location.
Use UNC path from Active Directory to create home location	Sets the computer to connect to the network share defined in the Active Directory user account. The UNC path is converted to SMB when the target share is running Windows or AFP when the target is running Mac OS X. If the setting for forcing the home directory on the startup disk is enabled, the UNC path is used to create a folder in the user's dock and the home directory is set to the user's local home directory path.

To set the path for the home directory, go to the **Profile** tab of the user's properties in ADUC and under **Home folder** select **Connect**, choose a drive letter (which is ignored by a Mac OS X computer), and then in the **To** box type the UNC path that you want.

Here is the form the path takes: `\\server\share\folder`

Here is an example of a path:
`\\example\homes\fanthony`

Contact Technical Support

BeyondTrust Software, Inc. provides an online knowledge base, as well as telephone and web-based support. Also, this guide includes topics about Troubleshooting Domain-Join Problems and Troubleshooting the PBIS Agent.

Before Contacting Technical Support

To expedite support, collect the following information to provide to Technical Support:

- PBIS Open version (Available in the PBIS Console by clicking **Help, About** on the menu bar.)
- PBIS Agent version and build number (See Check the Version and Build Number.)
- Linux or Unix version
- Windows or Windows Server version

As a best practice, if you are contacting Technical Support about one of the provide the following problems, also provide the diagnostic information specified.

Segmentation Faults

Provide the following additional information when contacting Technical Support:

- Core dump of the PowerBroker Identity Services application:
  `ulimit - c unlimited`
- Exact patch level or exact versions of all installed packages. (See Check the Version and Build Number.)

Program Freezes

Provide the following additional information when contacting Technical Support:

- Debug logs
- tcpdump
- An `strace` of the program

Domain-Join Errors

See Troubleshooting Domain-Join Problems.

Provide the following additional information when contacting Technical Support:

- Debug logs (See Generate a Domain-Join Log or grab the log file from `/var/log/pbis-join.log`.)
- tcpdump

All Active Directory Users Are Missing

See Solve Logon Problems on Linux or Unix or Solve Logon Problems from Windows.

Provide the following additional information when contacting Technical Support:

- Run `/opt/pbis/bin/get-status` (See List the Status of the Authentication Providers.)
- Contents of `nsswitch.conf`

All Active Directory Users Cannot Log On

Provide the following additional information when contacting Technical Support:

- Output of `id <user>`
- Output of `su -c 'su <user>' <user>`
- Lsass debug logs (See Generate an Authentication Agent Debug Log.)
- Contents of pam.d/pam.conf
- The sshd and ssh debug logs and syslog

AD Users or Groups are Missing

Provide the following additional information when contacting Technical Support:

- The debug logs for lsass
- Output for `getent passwd` or `getent group` for the missing object
- Output for `id <user>` if user
- tcpdump
- Copy of lsass cache file. (For more about the file name and location of the cache files, see PBIS Agent.)

Poor Performance When Logging On or Looking Up Users

Provide the following additional information when contacting Technical Support:

- Output of `id <user>`
- The lsass debug log
- Copy of lsass cache file. (For more about the file name and location of the cache files, see PBIS Agent.)
- tcpdump

Contacting Support

If you encounter problems that are not covered in the documentation, contact BeyondTrust Technical Support.

When contacting Technical Support, provide the following information:

- Your company name
- Telephone and email address where you can be contacted
- Description of the problem and the steps you have taken to resolve it
- Diagnostic information requested in Before Contacting Technical Support

You can contact BeyondTrust Technical Support by email or through the BeyondTrust website. If you are located in the United States, you can also contact Technical Support by telephone. Support is staffed 24 hours per day, seven days per week.

Telephone: +1 800-234-9072 or +1 818-575-4040

Email: pbis-support@beyondtrust.com

Web: To submit a support request online:

1. Browse to http://www.beyondtrust.com.
2. Click **Login** and log into the BeyondTrust website using the password provided to you by BeyondTrust.
3. After reading the Welcome message, scroll to the top of the **BeyondTrust Partner Portal** pane and click **Customer Support Center**.
4. Scroll down to the **Add/View Incidents** section and click the **+** icon.
5. In the **View Your Incidents** pane, click **Add Incident**, enter the details requested, and click **Submit Incident** to file your request for technical support.