Configuration Guide for Kerberos Client Products on HP-UX

HP-UX 11.0, HP-UX 11i v1, HP-UX 11i v2, and HP-UX 11i v3

Manufacturing Part Number: 5991-7718

February 2007

Legal Notices

Contents

1. Overview

Kerberos Overview 23
Authentication Process 24
Kerberos Products and GSS-API on HP-UX 28

2. Introduction to the Kerberos Products and GSS-API

PAM Kerberos 33
The PAM Framework 34
The Authentication Module 36
The Password Module 40
Credential Cache 41
The Account Management Module 46
The Session Management Module 46
Example 47
The pam_user.conf File on HP-UX 11.0 and 11i v1 47
The pam_user.conf File on HP-UX 11i v2 and HP-UX 11i v3 47
The pam.conf File on HP-UX 11.0 and HP-UX 11i v1 47
The pam.conf File on HP-UX 11i v2 and HP-UX 11i v3 48
The pam_krb5 File on HP-UX 11.0 and HP-UX 11i v1 48
The pam_krb5 File on HP-UX 11i v2 and HP-UX 11i v3 48
The pamkrbval Tool 48
Secure Internet Services 52
KRB5 Client Software 54
Libraries and Header Files 54
Kerberos Utilities 56
The kinit Utility 56
The klist Utility 59
The kdestroy Utility 60
The kpasswd Utility 61
The ktutil Utility 61
The kvno Utility 62
HP Kerberos Server 64
Kerberos Server Version 3.12 Features 64
Graphical User Interface (GUI) Based Administration tool 65
Multithreaded Server 65
High Availability 65
Dynamic Propagation 66
Scalability 66

Contents

Windows 2000(R) Interoperability 66
Choice of C-Tree or LDAP Database 67
Auto-Configuration Tool 67
Generic Security Service Application Programming Interface (GSS-API) 68
Credential Management Services 71
Context Level Services 71
Authentication Services 72
Confidentiality Service 72
Support Services 72

3. Configuring the Kerberos Environment

Configuration Files for Kerberos Clients 77
The services File 80
Configuration Files for GSS-API 82
The mech File 82
The /etc/gss/qop File 83
The gsscred.conf File 84
Configuring the Kerberos Server 85
Configuring Your Microsoft Windows 2000 KDC 85
Configuring the Kerberos Client 87
Configuring for PAM Kerberos 88

4. Troubleshooting Kerberos Related Products

Troubleshooting PAM Kerberos 91
Troubleshooting the Kerberos Client Utilities 94
Troubleshooting GSS-API 96
Error Codes 96
Major and Minor Status Values 96
Common GSS-API Errors 96
Calling Error Values 97
Other Common Causes of Errors 99
Troubleshooting Using the pamkrbval Tool 100

A. Sample pam.conf File

B. Sample krb5.conf File

C. Sample krb.conf File

D. Sample krb.realms File

E. Kerberos Error Messages

Kerberos V5 Library Error Codes 119
Kerberos V5 Magic Numbers Error Codes 129
ANSI.1 Error Codes 132
GSSAPI Error Codes 133

F. Kerberos Client Environment Variables

Kerberos Client Environment Variables. 138

Contents

Figures

Figure 1-1. Authentication Process 25
Figure 2-1. HP-UX authentication modules under PAM 34
Figure 2-2. PAM Kerberos calls libkrb5.sl through PAM 35
Figure 2-3. SIS uses Kerberos Client Library Directly 52
Figure 2-4. GSS-API Library 68
Figure 2-5. GSS-API Operation 69

Figures

Tables

Table 1. Publishing History Details 14
Table 2-1. PAM Kerberos Library libpam_krb5 35
Table 2-2. On HP-UX 11.0 and HP-UX 11i v1 37
Table 2-3. On HP-UX 11i v2 and HP-UX 11i v3. 37
Table 2-4. On HP-UX 11.0 and 11iv1 38
Table 2-5. On HP-UX 11i v2 and HP-UX 11i v3. 38
Table 2-6. Kerberos Client Libraries on HP-UX 11i v3 55
Table 2-7. Versions of Kerberos Server on HP-UX Operating Systems 64
Table 2-8. GSS-API Libraries 70
Table 2-9. Additional files in the GSS-API product 70
Table 3-1. Kerberos Configuration Files 77
Table 3-2. Entries in the mech file 82
Table 3-3. Format of the /etc/gss/qop file. 84
Table 4-1. Error Codes and Corrective Actions 91
Table 4-2. Kerberos Client Error Codes 94
Table 4-3. Common GSS-API Errors. 96
Table 4-4. Calling Errors 98
Table 4-5. Supplementary Bits 98
Table 4-6. Error Messages that Appear During keytab Validation. 100

Tables

About This Document

This document describes how to configure a Kerberos environment on HP-UX servers and workstations running on HP-UX 11.0, HP-UX 11i v1, HP-UX 11i v2, and HP-UX servers running on HP-UX 11i v3.

This document is intended for system managers or administrators who configure Kerberos related products on HP-UX. However, this document is not a replacement for the documents provided for HP's Kerberos Server version 3.12.

Publishing History

Table 1 describes the publishing details of this document for various HP-UX releases.

Table 1 Publishing History Details

Document Manufacturing Part Number	Operating Systems Supported	Publication Date
J5849-90003	HP-UX 11.X	December 2000
J5849-90007	HP-UX 11.X	September 2001
T1417-90005	HP-UX 11.X	June 2002
T1417-90006	HP-UX 11.X	July 2003
5991-7718	HP-UX 11.X	February 2007

The latest version of this document is available at: http://www.docs.hp.com.

The document printing date and part number indicate the document's correct edition. The printing date will change when a new edition is printed. Minor changes may be made at reprint without changing the printing date. The document part number will change when extensive changes are made.

Document updates can be issued between editions to correct errors or document product changes. To ensure that you receive the updated or new edition, subscribe to the appropriate support service.

Contact your HP sales representative for details.

Document Organization

The *Configuration Guide for Kerberos Related Products on HP-UX* is organized as follows:

Chapter 1	Chapter 1, Overview, – Provides an insight to the Kerberos protocol.
Chapter 2	Chapter 2, Introduction to the Kerberos Products and GSS-API, – Provides information about the different Kerberos products available on HP-UX.
Chapter 3	Chapter 3, Configuring the Kerberos Environment, – Provides instructions for configuring a Kerberos environment.
Chapter 4	Chapter 4, Troubleshooting Kerberos Related Products, – Provides information to help you identify and troubleshoot some common problems that might occur.
Appendix A	Appendix A, Sample pam.conf File, – Provides a sample pam.conf file.
Appendix B	Appendix B, Sample krb5.conf File, – Provides a sample krb5.conf file.
Appendix C	Appendix C, Sample krb.conf File, – Provides a sample krb.conf file.
Appendix D	Appendix D, Sample krb.realms File, – Provides a sample krb.realms file.
Appendix E	Appendix E, Kerberos Error Messages, – Provides some common Kerberos error messages with their respective error codes.
Appendix F	Appendix F, Kerberos Client Environment Variables,-- Provides a list of common Kerberos Client environment variables.

Typographic Conventions

This document uses the following typographic conventions:

audit (5)	An HP-UX manpage. In this example, *audit* is the name and 5 is the section in the *HP-UX Reference*. On the Web and on the Instant Information CD, it may be

	a link to the manpage itself. From the HP-UX command line, you can enter "`man audit`" or "`man 5 audit`" to view the manpage. See *man* (1).
Book Title	The title of a book. On the Web and on the Instant Information CD, it may be a link to the book itself.
KeyCap	The name of a keyboard key. Note that **Return** and **Enter** both refer to the same key.
Emphasis	Text that is emphasized.
Bold	The defined use of an important word or phrase.
`ComputerOut`	Text displayed by the computer.
`UserInput`	Commands and other text that you type.
`Command`	A command name or qualified command phrase.
`Variable`	The name of a variable that you may replace in a command or function or information in a display that represents several possible values.
`|`	Separates items in a list of choices.
`[ ]`	The contents are optional in formats and command descriptions. If the contents are a list separated by `|`, you can choose one of the items.
`{ }`	The contents are required in formats and command descriptions. If the contents are a list separated by `|`, you can choose one of the items.
...	The preceding element may be repeated an arbitrary number of times.

Related Documentation

Given below is a list of related documentation:

- Kerberos Server Version 3.12 Release Notes (5991-7686)
- PAM Kerberos v1.24 Release Notes (5991-7687)
- Installing and Administering Internet Services (B2355-90759)
- Using Internet Services (B2355-90148)

Accessing the World Wide Web

Given below is list of related documents that is available on the HP web sites:

- HP Technical Documentation and White Papers
  - `http://docs.hp.com`
  - `http://www.unixsolutions.hp.com/products/hpux/hpux11/whitepapers/netsecur.pdf`
  - `http://www.hp.com/products1/unix/operating/security/kerberos_wp.pdf`
- HP-UX IT Resource Center:
  - `http://us-support.external.hp.com` (US and Asia Pacific)
  - `http://europe-support.external.hp.com` (Europe)
- The Internet Engineering Task Force RFC Pages
  - `http://www.ietf.org/rfc.html`

Related Request for Comments (RFCs)

Given below is list of related Request for Comments:

- RFC 1510 - The Kerberos Network Authentication Service (V5)
- RFC 1964 - The Kerberos Version 5 GSS-API Mechanism
- RFC 2743 - Generic Security Service Application Program Interface
- RFC 2744 - Generic Security Service API
- Open Group RFC 86.0 - PAM Authentication Module

1 Overview

This chapter provides an overview of Kerberos and the available Kerberos products on HP-UX.

It contains the following sections:

- “Kerberos Overview” on page 23
- “Authentication Process” on page 24
- “Kerberos Products and GSS-API on HP-UX” on page 28

Kerberos Overview

Kerberos is a mature network authentication protocol based on the RFC 1510 specification of the IETF. It is designed to provide strong authentication for client or server applications by using the shared secret-key cryptography.

The basic currency of Kerberos is the ticket, which the user presents in order to use a specific service. Each service, be it a login service or an FTP service, requires a different kind of ticket. Fortunately, the Kerberized applications keep track of all the various kinds of tickets, so you don't have to.

You must authenticate yourself to the server by providing your user name and password. In return, the Kerberos server gives you an initial ticket, which you use to request for additional tickets from the Kerberos server for all the other services. For this reason, the initial ticket is also often called the Ticket Granting Ticket (TGT).

Use the Kerberos protocol to secure the communication between the client and server. Thus, client programs make authentication requests to an authentication server, and server programs in turn service those client requests. Based on your user credentials, the server program grants or denies your request to access network applications and services. The Kerberos server allows entities to authenticate themselves, without having to transmit their passwords in clear text form over the network.

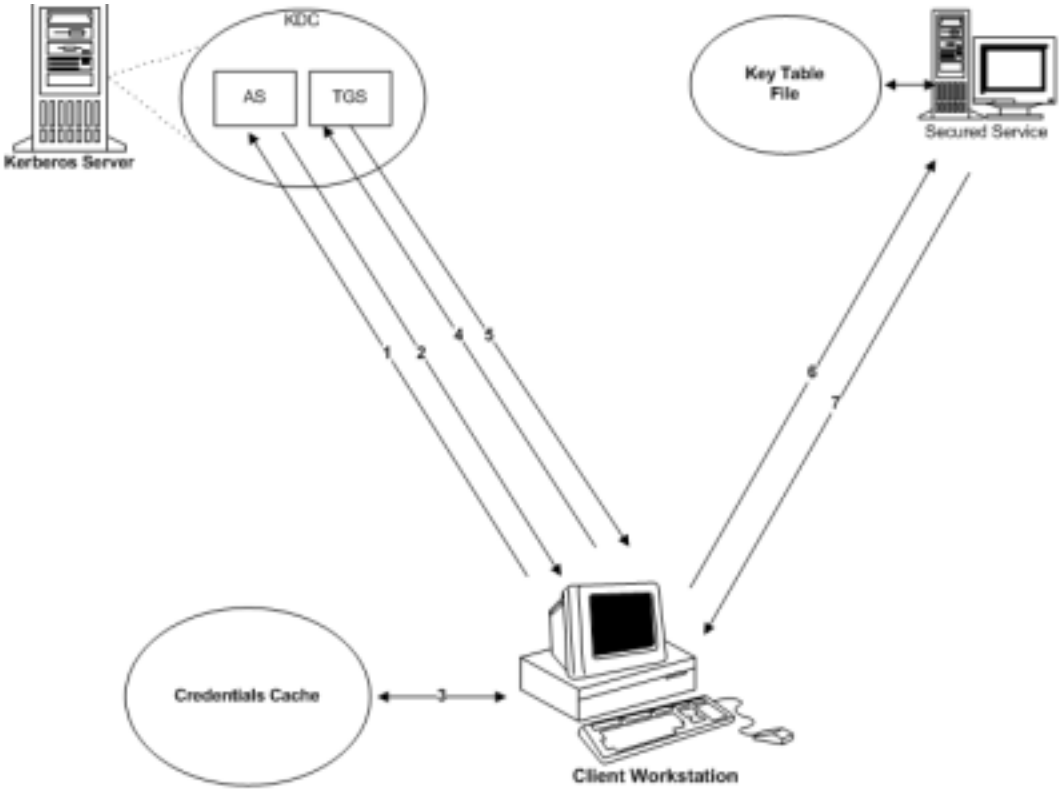
Authentication Process

The Kerberos server grants tickets to your user principal to access secured network services. You must authenticate yourself to the server by providing your user name and password. When the server authenticates you, it returns a set of initial credentials for you, including a TGT and a session key.

The Kerberos server grants a service ticket for a specific service principal that can be associated with one or more Kerberos-secured services. A client application uses your service ticket to authenticate you to a Kerberos-secured network service. The secured client application automatically handles the transactions with the Kerberos Server and the secured application server. Service tickets and associated session keys are generally cached in your user credentials cache along with the TGT of the user.

Figure 1-1 illustrates the actions of the components and the Kerberos protocol in a secured environment.

Figure 1-1 Authentication Process



The following is a description of how a client and server authenticate each other using Kerberos:

Step 1. Send a request to the AS for a TGT. You can choose to request specific ticket flags and specify the key type to be used to construct the secret key. You can also accept the default values configured for the client.

Send the following information to the Authentication Service (AS) to obtain credentials:

- Client-indicates the user name, also referred to as the principal name
- Server-indicates the TGS

- Time stamp
- Nonce

Step 2. If the AS decrypts the message successfully, it authenticates the requesting user and issues a TGT. The TGT contains the user name, a session key for your use, and name of the server to be used for any subsequent communication. The reply message is encrypted using your secret key.

NOTE The AS decrypts the request only when the pre-authentication option is set in the AS request. If the pre-authentication option is not set, the AS issues the TGT if the principal is available in the Kerberos database.

Step 3. The client decrypts the message using your secret key. The TGT and the session key from the message are stored in the client's credential cache. These credentials are used to obtain tickets for each network service the principal wants to access.

The Kerberos protocol exchange has the following important features:

- The authentication scheme does not require that the password be sent across the network, either in encrypted form or in clear text.
- The client (or any other user) cannot view or modify the contents of the TGT.

Step 4. To obtain access to a secured network service such as `rlogin`, `rsh`, `rcp`, `ftp`, or `telnet`, the requesting client application uses the previously obtained TGT in a dialogue with the TGS to obtain a service ticket. The protocol is the same as used while obtaining the TGT, except that the messages contain the name of the server and a copy of the previously obtained TGT.

Step 5. The TGS returns a new service ticket that the application client can use to authenticate to the service. The service ticket is encrypted with the service key shared between the KDC and the application server.

Step 6. The application server authenticates the client using the service key present in the keytab file. It decrypts the service ticket using the service key and extracts the session key. Using the session key, the server decrypts the authenticator and verifies the identity of the user. It also

verifies that the user's service ticket has not expired. If the user does not have a valid service ticket, then the server will return an appropriate error code to the client.

Step 7. (Optional) At the client's request, the application server can also return the timestamp sent by the client, encrypted in the session key. This ensures a mutual authentication between the client and the server.

Kerberos Products and GSS-API on HP-UX

HP-UX supports Kerberos products with a set of three software packages and Generic Security Service Application Programming Interface (GSS-API) for HP-UX 11.0 onwards. These products are:

- PAM Kerberos (PAM-Kerberos)
- Kerberos Client Software
- Kerberos Server
- GSS-API

Application programmers can create “Kerberized” applications using either the GSS-APIs or the Kerberos APIs. However, HP recommends that GSS-APIs be used for application development. HP provides the following Kerberized applications through Secure Internet Services (SIS): `ftp`, `rcp`, `remsh`, `rlogin`, and `telnet`.

NOTE SIS is available on HP-UX 11.0 and HP-UX 11i v1 only. From HP-UX 11i v2 onwards, all these applications directly link to `libkrb5`.

The HP-UX Kerberos-related products and GSS-API are:

- **PAM Kerberos (PAM-Kerberos)**: is the Kerberos implementation of the PAM Framework based on the RFC 86.0 of Open Group. PAM allows multiple authentication technologies to co-exist on HP-UX.
- **Kerberos Client Software**: includes libraries, header files and utilities for implementing Kerberized client/server applications in either 32-bit or 64-bit development environment.

  The client libraries are based on MIT Kerberos V5 1.1.1 release. The KRB5-Client libraries support DES encryption as specified in RFC 1510 of the IETF.

NOTE

On HP-UX 11i v3, the KRB5-Client libraries are based on MIT Kerberos V5 1.3.5 release. These KRB5-Client libraries support the DES, AES, 3DES and RC4 encryption types.

The Kerberos Client utilities are as follows:

— `kinit`, `klist`, and `kdestroy` to manage credentials

— `kpasswd` to change Kerberos passwords

— `ktutil` to maintain `keytab` file

— `kvno` to display the Kerberos key version number of the principals.

- **Kerberos Server Version 3.12**: The current version of the Kerberos server supersedes the earlier MIT-based Kerberos server (version 1.0), on HP-UX 11i.

  The Kerberos Server is based on a distributed client-server architecture. It ensures secure communication in a networked environment by leveraging individual trust relationships. It then brokers that trust across enterprise-wide, distributed client-server networks.

- **GSS-API**: is an interface that provides a secure client-server application programming. The GSS-API also provides authentication, integrity, and confidentiality services to the calling applications.

- **SIS:** is the built-in support for secure Internet services such as `ftp`, `rcp`, `rlogin`, `telnet` and `remsh` utilities. When secure Internet services are enabled, these commands use Kerberos for authentication without sending passwords in clear text over the network.

2 Introduction to the Kerberos Products and GSS-API

This chapter describes the Kerberos-based products and GSS-API on HP-UX.

It contains the following sections:

- “PAM Kerberos” on page 33
- “Secure Internet Services” on page 52
- “KRB5 Client Software” on page 54
- “HP Kerberos Server” on page 64
- “Generic Security Service Application Programming Interface (GSS-API)” on page 68

PAM Kerberos

HP-UX provides Kerberos authentication as part of the Pluggable Authentication Module (PAM) architecture as specified in RFC 86.0, of the Open Group. PAM allows multiple authentication technologies to co-exist on HP-UX. The `/etc/pam.conf` configuration file determines the authentication module to be used in a manner transparent to the applications that use the PAM library.

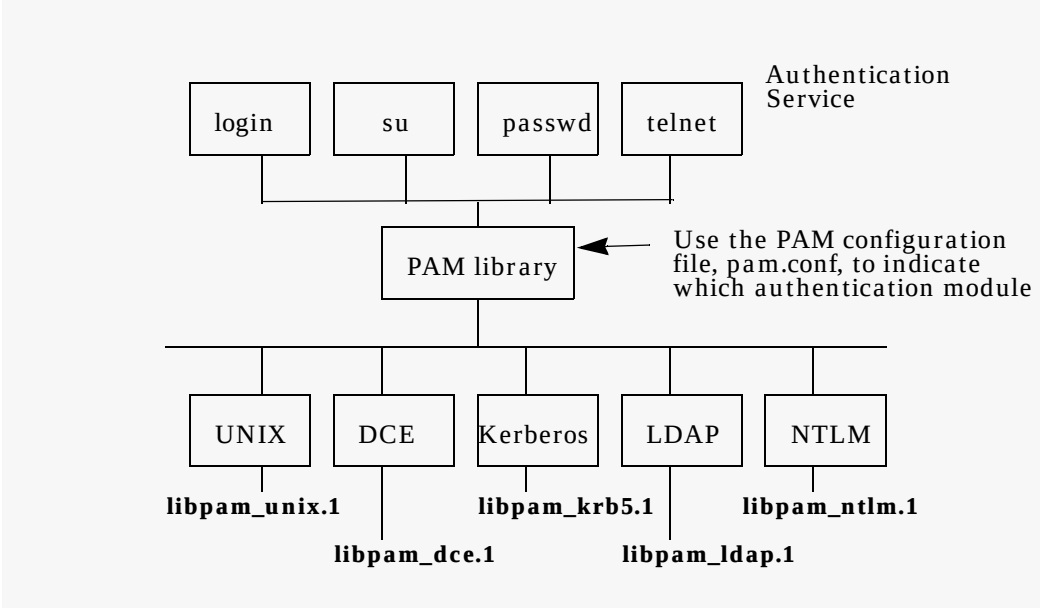
This product consists of the following:

- PAM Kerberos library - `libpam_krb5`
- PAM Kerberos Configuration validation tool - `pamkrbval`. Refer to “The pamkrbval Tool” on page 48, for more information.

The PAM Framework

Figure 2-1 shows the relationship between the PAM Kerberos Library and various authentication modules that HP-UX provides. Note that the PAM Kerberos Library is one of the many authentication modules that PAM can invoke based on what is defined under the PAM configuration file: `/etc/pam.conf`.

Figure 2-1 HP-UX authentication modules under PAM



PAM Kerberos is invoked for user authentication, when PAM's authentication-management module is pointed to the shared dynamically loadable PAM Kerberos library, `libpam_krb5`. Table 2-1 indicates the location of the library on both Itanium® and PA-RISC based platforms.

Table 2-1 PAM Kerberos Library libpam_krb5

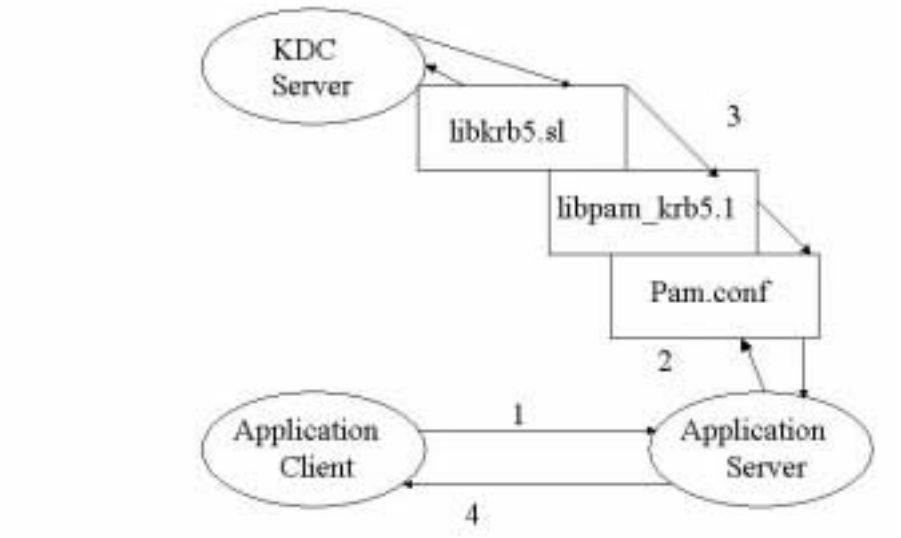
Platform	Location
Itanium® - based platform	`/usr/lib/security/$ISA/libpam_krb5.so.1`
PA-RISC platform	`/usr/lib/security/libpam_krb5.1`

Figure 2-2 shows a secure environment consisting of the following nodes:

- KDC Server
- The application server (`rlogind` process)
- The application client (`rlogin` process)

The application client is not a KDC client under PAM Kerberos.

Figure 2-2 PAM Kerberos calls libkrb5.sl through PAM



When using PAM Kerberos, users only configure the application server as a KDC client. Users are prompted for a password when they first log into the server from the application client. The user has no credential and their password is sent in clear text to the application server. Following are the authentication steps as shown in Figure 2-2:

1. The user sends a password to a remote system
2. The application server invokes `libkrb5.sl` through PAM to request for authentication from the KDC
3. KDC replies with an authenticator
4. If the password provided is valid, then the user is authenticated. If the password is incorrect, the user is denied access.

The Kerberos service module for PAM consists of the following four modules:

- Authentication module
- Account management module
- Session management module
- Password management module

All modules are supported through the same dynamically loadable library, `libpam_krb5`. The `KRB5 PAM` modules are compatible with MIT Kerberos 5 and Microsoft Windows 2000.

The Authentication Module

The Authentication module verifies the identity of a user and sets user-specific credentials. It authenticates the user to KDC with a password. If the password matches, the user is authenticated and a Ticket Granting Ticket (TGT) is granted.

The Authentication Module supports the following options:

- `use_first_pass`
- `krb_prompt`
- `try_first_pass`
- `renewable=<time>`
- `forwardable`

- proxiable
- debug
- ignore

The following paragraphs list and describe each of these options.

Option	Definition
use_first_pass	Uses the same password given to the first module configured for authentication in the pam.conf file (see Figure 2-1). The module does not prompt for the password if the user cannot be authenticated by the first password. This option is used when the system administrator wants to enforce the same password across multiple modules. In the following code fragment from a pam.conf file, both libpam_krb5.1 and libpam_unix.1 are defined in the PAM stack as authentication modules. If a user is not authenticated under libpam_unix.1, PAM tries to authenticate the user through libpam_krb5.1 using the same password used with libpam_unix.1. If the authentication fails, PAM does not prompt for another password.

Table 2-2 On HP-UX 11.0 and HP-UX 11i v1

```
login  auth sufficient  /usr/lib/security/libpam_unix.1
login  auth required  /usr/lib/security/libpam_krb5.1  use_first_pass
```

Table 2-3 On HP-UX 11i v2 and HP-UX 11i v3

```
login  auth sufficient  libpam_unix.so.1
login  auth required       libpam_krb5.so.1  use_first_pass
```

krb_prompt	This option allows the administrator to change the password prompt. When set, the password prompt displayed is, Kerberos Password.
try_first_pass	This option is similar to the use_first_pass option, except that if the primary password is not valid, PAM prompts for a password.

In the following code fragment from a `pam.conf` file, both `libpam_krb5.1` and `libpam_unix.1` are defined in the PAM stack as authentication modules. If a user is not authenticated under `libpam_unix.1`, PAM tries to authenticate the user through `libpam_krb5.1` using the same password that is used with `libpam_unix.1`. If the authentication fails, PAM prompts for another password and tries again.

Table 2-4 On HP-UX 11.0 and 11iv1

```
login  auth sufficient  /usr/lib/security/libpam_unix.1
login  auth required  /usr/lib/security/libpam_krb5.1  try_first_pass
```

Table 2-5 On HP-UX 11i v2 and HP-UX 11i v3

```
login  auth sufficient  libpam_unix.so.1
login  auth required       libpam_krb5.so.1  try_first_pass
```

`renewable=<time>` This option allows the user to implement ticket renewal. Renewable tickets have two “expiration times”: the first is when the current instance of the ticket expires, and the second is the latest permissible value for an individual expiration time. When the latest permissible expiration time arrives, the ticket expires permanently.

For renewable tickets to be granted, you must specify that the user can be granted renewable tickets in the user's account in the Kerberos KDC.

`forwardable` When a user obtains service tickets, they are for a remote system. However, the user can use a secure service to access a remote system and run a secure service from that remote system to a second remote system. This requires a valid TGT for the first remote system. Kerberos provides the option to create TGTs with special attributes, which allow service tickets to be forwarded to remote systems within the realm.

The forwardable flag in a ticket allows the service complete use of the client's identity. It is used when a user logs in to a remote system and wants authentication to work from that system as if the login were local.

For forwardable tickets to be granted, you must specify that the user can be granted forwardable tickets in the user's account in the Kerberos KDC.

`proxiable` At times, it may be necessary for a principal to allow a service to perform an operation on its behalf. The service must be able to take on the identity of the client, but only for a particular purpose by granting it a proxy.

This option allows a client to pass a proxy ticket to a server to perform a remote request on its behalf. For example, a print service client can give the print server a proxy to access the client's files on a particular file server.

For proxy tickets to be granted, you must specify that the user can be granted proxy tickets in the user's account in the Kerberos KDC.

`ignore` The `ignore` option in the `pam_user.conf` file enables you to configure PAM such that certain users or services need not be authenticated. This option returns `PAM_IGNORE`. HP recommends not to use this option for Kerberos authentication in the `pam.conf` file.

For example, with the following configuration, no Kerberos authentication is conducted for the root user.

On HP-UX 11.0 and HP-UX 11i v1

```
pam_user.conf:
#
# configuration for user root. KRB5 PAM module uses the ignore

# option and returns PAM_IGNORE without any processing.
#
root auth      /usr/lib/security/libpam_krb5.1 ignore
root password  /usr/lib/security/libpam_krb5.1 ignore
root account   /usr/lib/security/libpam_krb5.1 ignore
root session   /usr/lib/security/libpam_krb5.1 ignore
```

On HP-UX 11i v2 and HP-UX 11i v3

```
pam_user.conf:
#
# configuration for user root. KRB5 PAM module uses the ignore
```

```
# option and returns PAM_IGNORE without any processing.
#
root auth      /usr/lib/security/$ISA/libpam_krb5.so.1 ignore
root password  /usr/lib/security/$ISA/libpam_krb5.so.1 ignore
root account   /usr/lib/security/$ISA/libpam_krb5.so.1 ignore
root session   /usr/lib/security/$ISA/libpam_krb5.so.1 ignore
```

To enable the configuration defined in the pam_user.conf file, the libpam_updbe module must be the first module in the stack in the pam.conf file. PAM Kerberos uses libpam_updbe to read user policy definitions from the pam_user.conf file. Refer to the manpage pam_updbe (5) for more information about per user PAM configuration.

`debug` The `debug` option sets `syslog` debugging information at the `LOG_DEBUG` level.

The Password Module

The Password Management module provides a function to change passwords in the Kerberos password database. Unlike when changing a Unix password, a root user is always prompted for the old password.

The following options can be passed to this PAM module through the `/etc/pam.conf (4)` file:

`debug` This option allows `syslog(3C)` debugging information at `LOG_DEBUG` level.

`krb_prompt` This option allows the administrator to change the password prompt. When set, the password prompt displayed is `Old/New Kerberos Password`.

When a user logs onto a system using PAM kerberos they obtain credentials that are stored in a file. This file is deleted when the user logs out of the system if the `/etc/pam.conf` file contains an entry for PAM Kerberos under session management and the application calls `pam_close_session()`.

In the `/etc/pam.conf`, if the flag `krb_prompt` is added to either the login/password entry, the prompt explicitly specifies Kerberos as shown below:

```
$ old password <--- Previous output
```

```
$ old Kerberos password <--- Output if
krb_prompt is specified
```

`user_first_prompt` This option allows the initial password (entered when the user is authenticated to the first authentication module in the stack) to authenticate with Kerberos. If the user cannot be authenticated or if this is the first authentication module in the stack, it quits without prompting for a password. HP recommends using this option only if the authentication module is designated as optional in the `/etc/pam.conf(4)` configuration file.

`try_first_pass` This option allows the initial password (entered when the user is authenticated to the first authentication module in the PAM stack) to authenticate with Kerberos. If the user cannot be authenticated or if this is the first authentication module in the stack, it prompts the user for a password.

`ignore` This option returns `PAM_IGNORE`. HP recommends not using this option. However, if you do not want to authenticate certain users or services with Kerberos, you can use this option in the `/etc/pam_user.conf(4)` file for per user configuration. HP recommends not using this option in the `pam.conf(4)`file.

Refer to `/etc/pam.krb5` in Appendix A, “Sample pam.conf File,” for a sample `pam.conf` file configured for PAM Kerberos.

Credential Cache

The credential management function in Kerberos sets user-specific credentials. It stores the credentials in a cache file and exports the `KRB5CCNAME` environment variable to identify the cache file. Any subsequent kerberos service access can use the same credential file. The name of that file is retrieved from `KRB5CCNAME`.

A credential file is created in `the /tmp` directory when the user accesses the system.

If the user first accesses the system from any system entry service -- such as `login`, `ftp`, `rlogin`, or `telnet` -- a unique credential file is created in the `/tmp/creds` directory. This file is named `krb5cc_<ppid>_<pid>`, where:

`ppid`	is the parent process
`pid`	is the process id of the process that is creating this credential file

An example `PAM` configuration file is as shown below:

On HP-UX 11.0 and 11iv1

```
#
# PAM configuration
#
# Authentication management
#
login      auth sufficient   /usr/lib/security/libpam_krb5.1
login      auth required     /usr/lib/security/libpam_unix.1
try_first_pass
su         auth sufficient   /usr/lib/security/libpam_krb5.1
su         auth required     /usr/lib/security/libpam_unix.1
try_first_pass
dtlogin    auth sufficient   /usr/lib/security/libpam_krb5.1
dtlogin    auth required     /usr/lib/security/libpam_unix.1
try_first_pass
dtaction   auth sufficient   /usr/lib/security/libpam_krb5.1
dtaction   auth required     /usr/lib/security/libpam_unix.1
try_first_pass
ftp        auth sufficient   /usr/lib/security/libpam_krb5.1
ftp        auth required     /usr/lib/security/libpam_unix.1
try_first_pass
OTHER      auth sufficient     /usr/lib/security/libpam_unix.1
#
# Account management
#
login      account required /usr/lib/security/libpam_krb5.1
login      account required /usr/lib/security/libpam_unix.1
su         account required /usr/lib/security/libpam_krb5.1
su         account required /usr/lib/security/libpam_unix.1
dtlogin    account required /usr/lib/security/libpam_krb5.1
dtlogin    account required /usr/lib/security/libpam_unix.1
dtaction   account required /usr/lib/security/libpam_krb5.1
dtaction   account required /usr/lib/security/libpam_unix.1
ftp        account required /usr/lib/security/libpam_krb5.1
ftp        account required /usr/lib/security/libpam_unix.1
OTHER      account sufficient /usr/lib/security/libpam_unix.1
#
# Session management
#
login      session required /usr/lib/security/libpam_krb5.1
login      session required /usr/lib/security/libpam_unix.1
dtlogin    session required /usr/lib/security/libpam_krb5.1
dtlogin    session required /usr/lib/security/libpam_unix.1
dtaction   session required /usr/lib/security/libpam_krb5.1
dtaction   session required /usr/lib/security/libpam_unix.1
```

```
OTHER      session sufficient /usr/lib/security/libpam_unix.1
#
# Password management
#
login      password sufficient /usr/lib/security/libpam_krb5.1
login      password required /usr/lib/security/libpam_unix.1
passwd     password sufficient /usr/lib/security/libpam_krb5.1
passwd     password required /usr/lib/security/libpam_unix.1
dtlogin    password sufficient /usr/lib/security/libpam_krb5.1
dtlogin    password required /usr/lib/security/libpam_unix.1
dtaction   password sufficient /usr/lib/security/libpam_krb5.1
dtaction   password required /usr/lib/security/libpam_unix.1
OTHER      password sufficient /usr/lib/security/libpam_unix.1
```

On HP-UX 11i v2 and HP-UX 11i v3

```
#
# PAM configuration
#
# Notes: This pam.conf file is intended as an example only.
# If the path to a library is not absolute, it is assumed to be
# relative to one of the following directories:
# /usr/lib/security         (PA 32-bit)
# /usr/lib/security/pa20_64 (PA 64-bit)
# /usr/lib/security/hpux32  (IA 32-bit)
# /usr/lib/security/hpux64  (IA 64-bit)
# The IA file name convention is normally used; for example:
# libpam_unix.so.1
# For PA libpam_unix.so.1 is a symbolic link to the PA library:
# ln -s libpam_unix.1 libpam_unix.so.1
# Also note that the use of pam_hpsec(5) is mandatory for some of the
# services. See pam_hpsec(5).
# Authentication management
#
login      auth sufficient   libpam_krb5.so.1
login      auth required     libpam_unix.so.1
try_first_pass
su         auth sufficient   libpam_krb5.so.1
su         auth required     libpam_unix.so.1
try_first_pass
dtlogin    auth sufficient   libpam_krb5.so.1
dtlogin    auth required     libpam_unix.so.1
try_first_pass
dtaction   auth sufficient   libpam_krb5.so.1
dtaction   auth required     libpam_unix.so.1
try_first_pass
ftp        auth sufficient   libpam_krb5.so.1
```

```
ftp        auth required     libpam_unix.so.1
try_first_pass
OTHER      auth sufficient   libpam_unix.so.1
#
# Account management
#
login      account required   libpam_krb5.so.1
login      account required   libpam_unix.so.1
su         account required   libpam_krb5.so.1
su         account required   libpam_unix.so.1
dtlogin    account required   libpam_krb5.so.1
dtlogin    account required   libpam_unix.so.1
dtaction   account required   libpam_krb5.so.1
dtaction   account required   libpam_unix.so.1
ftp        account required   libpam_krb5.so.1
ftp        account required   libpam_unix.so.1
OTHER      account sufficient libpam_unix.so.1
#
# Session management
#
login      session required   libpam_krb5.so.1
login      session required   libpam_unix.so.1
dtlogin    session required   libpam_krb5.so.1
dtlogin    session required   libpam_unix.so.1
dtaction   session required   libpam_krb5.so.1
dtaction   session required   libpam_unix.so.1
OTHER      session sufficient libpam_unix.so.1
#
# Password management
#
login      password sufficient libpam_krb5.so.1
login      password required   libpam_unix.so.1
passwd     password sufficient libpam_krb5.so.1
passwd     password required   libpam_unix.so.1
dtlogin    password sufficient libpam_krb5.so.1
dtlogin    password required   libpam_unix.so.1
dtaction   password sufficient libpam_krb5.so.1
dtaction   password required   libpam_unix.so.1
OTHER       password sufficient    libpam_unix.so.1
```

The Account Management Module

The Account Management module provides a function to perform account management. This function retrieves the user's account and password expiration information from the Kerberos database and verifies that they have not expired. The module does not issue any warning if the account or the password is about to expire.

The following options can be passed to the Account Management module through the `/etc/pam.conf(4)` file:

`debug` This option allows `syslog(3C)` debugging information at `LOG_DEBUG` level.

`ignore` This option returns `PAM_IGNORE`. HP recommends not using this option unless it is not necessary to authenticate certain users or services with Kerberos. In such cases you can use the `ignore` option in the `pam_user.conf` file for per user configuration.

HP does not recommend using this option in the `pam.conf` file.

The Session Management Module

The session management module provides function to terminate sessions. It cleans up the credential cache file created by the Authentication module.

The following options can be passed to the session management module through the `/etc/pam.conf(4)` file:

`debug` This option allows `syslog(3C)` debugging information at `LOG_DEBUG` level.

`ignore` This option returns `PAM_IGNORE`. HP recommends not using this option unless it is not necessary to authenticate certain users or services with Kerberos. In such cases you can use the ignore option in the `pam_user.conf` file for per user configuration. HP does not recommend using this option in the `pam.conf` file.

Example

The following is a sample configuration in which no authentication is done with Kerberos for root. KRB5 PAM module does nothing. It just returns PAM_IGNORE for user root. For every user other than root, it tries to authenticate using Kerberos. If Kerberos succeeds, the user is authenticated. If Kerberos fails to authenticate the user, PAM tries to authenticate with UNIX PAM using the same password.

The pam_user.conf File on HP-UX 11.0 and 11i v1

```
# configuration for user root. KRB5 PAM module uses the
# ignore option and returns PAM_IGNORE
root    auth     /usr/lib/security/libpam_krb5.1 ignore
root    password /usr/lib/security/libpam_krb5.1 ignore
root    account  /usr/lib/security/libpam_krb5.1 ignore
root    session  /usr/lib/security/libpam_krb5.1 ignore
```

The pam_user.conf File on HP-UX 11i v2 and HP-UX 11i v3

```
# configuration for user root. KRB5 PAM module uses the
# ignore option and returns PAM_IGNORE
root    auth     /usr/lib/security/$ISA/libpam_krb5.so.1 ignore
root    password /usr/lib/security/$ISA/libpam_krb5.so.1 ignore
root    account  /usr/lib/security/$ISA/libpam_krb5.so.1 ignore
root    session  /usr/lib/security/$ISA/libpam_krb5.so.1 ignore
```

The pam.conf File on HP-UX 11.0 and HP-UX 11i v1

```
# For per user configuration the libpam_updbe.1 (pam_updbe(5)) module
# must be the first module in the stack. If Kerberos authentication
# is valid the UNIX authentication function will not be invoked.

login   auth      required    /usr/lib/security/libpam_updbe.1
login   auth      sufficient  /usr/lib/security/libpam_krb5.1
login   auth      required    /usr/lib/security/libpam_unix.1 try_first_pass
login   password  required    /usr/lib/security/libpam_updbe.1
login   password  required    /usr/lib/security/libpam_krb5.1
login   password  required    /usr/lib/security/libpam_unix.1 try_first_pass
login   account   required    /usr/lib/security/libpam_updbe.
login    account    required     /usr/lib/security/libpam_krb5.1
```

The pam.conf File on HP-UX 11i v2 and HP-UX 11i v3

```
# For per user configuration the libpam_updbe.1 (pam_updbe(5)) module
# must be the first module in the stack. If Kerberos authentication
# is valid the UNIX authentication function will not be invoked.

login   auth      required    libpam_updbe.so.1
login   auth      sufficient  libpam_krb5.so.1
login   auth      required    libpam_unix.so.1  try_first_pass
login   password  required    libpam_updbe.so.1
login   password  required    libpam_krb5.so.1
login   password  required    libpam_unix.so.1  try_first_pass
login   account   required    libpam_updbe.so.1
login    account    required     libpam_krb5.so.1
```

The pam_krb5 File on HP-UX 11.0 and HP-UX 11i v1

```
login   account   required    /usr/lib/security/libpam_unix.1
login   session   required    /usr/lib/security/libpam_updbe.1
login   session   required    /usr/lib/security/libpam_krb5.1
login   session   required        /usr/lib/security/libpam_unix.1
```

The pam_krb5 File on HP-UX 11i v2 and HP-UX 11i v3

```
login   account   required    /usr/lib/security/$ISA/libpam_unix.so.1
login   session   required    /usr/lib/security/$ISA/libpam_updbe.so.1
login   session   required    /usr/lib/security/$ISA/libpam_krb5.so.1
login    session    required     /usr/lib/security/$ISA/libpam_unix.so.1
```

The pamkrbval Tool

Use the `pamkrbval` tool to validate your PAM Kerberos configuration. This tool verifies PAM Kerberos configuration files and enables the system administrator to diagnose the problem, if any. Following are the files, the `pamkrbval` tool verifies:

- `/etc/pam.conf`
- `/etc/pam_user.conf`
- `/etc/krb5.conf`
- `/etc/krb5.keytab`

This tool also checks if the default realm KDC is up and running.

The `pamkrbval` tool validates the following:

- Checks for the validity of the `control_flags` and the `module_types` specified for the PAM Kerberos specific entries in the `/etc/pam.conf` file.
- Checks if the PAM Kerberos specific `module_path` specified in the `/etc/pam.conf` file exists. If the `module_path` name is not absolute it is assumed to be relative to `/usr/lib/security/$ISA/`. The `$ISA` (Instruction Set Architecture) token is replaced by this tool with `hpux32` for Itanium® 32-bit option (ia32), or with `hpux64` for Itanium® 64 bit option (ia64), or with null for PA-32 bit option (pa32), or with pa20_64 for PA 64-bit option (pa64).
- Checks if the options specified for the `pam_krb5` library are valid PAM Kerberos options.
- Validates the `/etc/pam_user.conf` file only if `libpam_updbe` is configured in the `/etc/pam.conf` file. This validation is similar to the `/etc/pam.conf` validation.
- Validates the syntax of the Kerberos configuration file, `/etc/krb5.conf`.
- Validates if the default realm KDC is issuing tickets. At least one KDC must reply to the ticket requests for the default realm.
- Validates the host service principal, `host/<hostname>@default_realm>` in `/etc/krb5.keytab`, if present. If the keytab entry for this host service principal is not present in the default keytab file, `/etc/krb5.keytab` then that validation is ignored and success is assumed.

NOTE

An entry in `/etc/pam.conf` file is considered to be PAM Kerberos entry if the file name in the `module_path` begins with `libpam_krb5`. An example of a PAM Kerberos entry in `/etc/pam.conf` is as shown:

```
login auth required /usr/lib/security/$ISA/libpam_krb5.so.1
```

The machine is considered to be configured with `libpam_updbe` if the file name in the `module_path` of an entry in `/etc/pam.conf` begins with `libpam_updbe`. Following is an example of a `pam_updbe` entry in the `/etc/pam.conf` file:

```
login auth required /usr/lib/security/$ISA/libpam_updbe.so.1
```

Logging

The pamkrbval tool logs all messages to stdout. Following are the log categories provided:

[LOG]	These messages are logged when the verbose option is set.
[NOTICE]	These messages are logged to notify the user about the erroneous lines in the PAM configuration files or notify about the skipping of /etc/pam_user.conf file validation.
[FAIL]	These messages are logged when validation fails.
[WARNING]	These messages are logged to notify the user about a potentially erroneous configuration on the system that may result in failure.
[PASS]	These messages are logged when any validation succeeds.
[IGNORE]	These messages are logged when validation of the /etc/krb5.keytab is ignored.
ERROR	These messages are logged to inform the user about the exact problem in the PAM configuration files
[HELP]	These messages will give some minimal help to the user to rectify the problem.

If you get any [FAIL] or ERROR messages, you must diagnose the nature of the problem. See “Troubleshooting Using the pamkrbval Tool” on page 100 for more information.

Options

Use the following command with the options listed below:

```
/usr/sbin/pamkrbval
```

-v[erbose]	verbose output
-a	{pa32 | pa64 |ia32 | ia64} Depending on the architecture on which the validation need to be done this option needs to be set. The flags available are as listed below: pa32 for PA 32-bit architecture pa64 for PA 64-bit architecture ia32 for Itanium® 32-bit architecture

	ia64 for Itanium® 64-bit architecture Depending on this flag, $ISA in the `module_path` will be expanded as explained above.
`-c`	Use this option when Common Internet File System (CIFS) is configured on the system.

Return Value

The `pamkrbval` tool returns the following exit codes:

`0`	Successful configuration validation
`1`	Warnings were found during configuration validation
`2`	Errors were detected during configuration validation

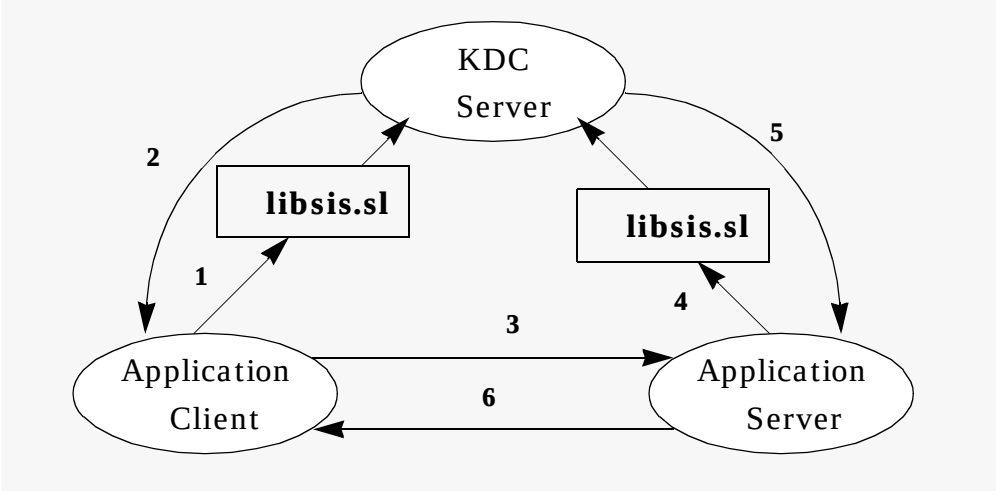
Secure Internet Services

If you want to authenticate users on remote systems without sending the password in clear text over the network, you can use the built-in support that HP provides for the following secure Internet services applications:

- `ftp`
- `rcp`
- `rlogin`
- `telnet`
- `remsh`

In Figure 2-3, SIS invokes the `libsis.sl` library. When SIS is enabled at the application client, the password is not sent to the application server. Instead, SIS uses an encrypted ticket each time the user requests a remote service.

Figure 2-3 SIS uses Kerberos Client Library Directly



As shown in Figure 2-2:

1. The application client requests for credentials from the KDC
2. The application client obtains credentials for the remote host (the application server)

3. Using the credentials, the application client creates an authenticator and sends the authenticator and service ticket to the remote host.
4. The kerberized telnet server on the remote host verifies the user identity by decrypting the service ticket.

To turn on SIS, issue the following command at the HP-UX command prompt:

```
inetsvcs_sec enable
```

NOTE The library, `libsis.sl`, is supported upto the HP-UX 11i v1.5 release. From the HP-UX 11i v1.6 release onwards the library, `libkrb5.so` is supported.

KRB5 Client Software

This section presents an overview of the KRB5-Client software, which consists of libraries, header files, manpages, and Kerberos utilities. The section is divided into two parts. The following subsection, “Libraries and Header Files”, discusses the libraries and header files supplied with the KRB5-Client software. The second subsection,“Kerberos Utilities” on page 56, discusses the Kerberos utilities.

Libraries and Header Files

This section lists and describes the libraries and header files supplied with the KRB5-Client software.

You can use the KRB5-Client libraries to develop secure client/server applications for either 32-bit or 64-bit environments on any of the HP-UX 11.X platforms.

The client libraries are based on MIT Kerberos V5 1.1.1 release. This release is compatible with MIT Kerberos 1.2 and interoperable with Microsoft Windows 2000.

NOTE On HP-UX 11i v3, the KRB5-Client libraries are based on MIT Kerberos V5 1.3.5 release. These KRB5-Client libraries support the DES, AES, 3DES and RC4 encryption types.

The Kerberos Client Library (`libkrb5.so`) replaces the KRB-Support Library (`libsis.sl`) from the HP-UX 11i v1.6 release onwards.

The KRB5-Client libraries support Data Encryption Standard (DES) as specified in RFC 1510 of the IETF.

Table 2-6 lists and describes the Kerberos client libraries.

Table 2-6 Kerberos Client Libraries on HP-UX 11i v3

Architecture	32-bit	64-bit	Functionality
PA-RISC	/usr/lib/libkrb5.sl ->/usr/lib/libkrb5.1	/usr/lib/pa20_64/ libkrb5.sl -> /usr/lib/pa20_64/ librb5.1	Authenticates users, verifies tickets, creates authenticator, and manages the context
	/usr/lib/ libcom_err.sl -> /usr/lib/libcom_err.1	/usr/lib/pa20_64/ libcom_err.sl -> /usr/lib/pa20_64/ libcom_err.1	Prints appropriate error messages to stderr, based on the error code returned by the Kerberos APIs
	/usr/lib/ libk5crypto.sl -> /usr/lib/libk5crypto.1	/usr/lib/pa20_64/ libk5crypto.sl -> /usr/lib/pa20_64/ libk5crypto.1	Encrypts (using DES, 3DES, AES, and RC4 cryptographic algorithms) and decrypts all communication between users to ensure privacy and data integrity
	/usr/lib/gss/ libgssapi_krb5.sl -> /usr/lib/gss/libgssapi_ krb5.1	/usr/lib/pa20_64/ gss/ libgssapi_krb5.sl -> /usr/lib/pa20_64/ gss/libgssapi_krb 5.1	Kerberos mechanism specific library used by GSSAPI (/usr/lib/libgss. sl)

The Kerberos client software also provides the following header files:

- /usr/include/profile.h
- /usr/include/krb5.h

- /usr/include/com_err.h
- /usr/include/krb5/gssapi.h

HP-UX includes DCE Kerberos and its manpages, so you must use specific manpage numbers for the Kerberos client software. For example, refer to `man 1 kinit` for the Kerberos manpages and to `man 1m kinit` for the DCE manpage. The default is the Kerberos manpage.

Refer to `/usr/share/man/man3.Z/libkrb5.3` for more information on the `libkrb5` library. See "Kerberos Utilities" on page 56 for information on other Kerberos utilities.

NOTE IPv6 support for Kerberos Clients has been enabled only for the Itanium® binaries on HP-UX 11i v2 and PA-RISC and Itanium binaries on HP-UX 11i v3 systems.

Kerberos Utilities

The HP-UX implementation of Kerberos utilities is compatible with the MIT reference implementation.

On HP-UX 11i onwards, the Kerberos utilities are part of the OS core. On HP-UX 11.0, they are bundled with PAM Kerberos from the quarterly distributed Application CD.

All the utilities, except `ktutil`, are available for all users.The `ktutil` utility is restricted for administrator use only.

The kinit Utility

Description The `kinit` utility obtains the Kerberos ticket-granting ticket for the requesting principal and stores it in the credential cache file.

Synopsis

```
/usr/bin/kinit -l lifetime [principal]
/usr/bin/kinit -s start_time [principal]
/usr/bin/kinit -v [principal]
/usr/bin/kinit -p [principal]
/usr/bin/kinit -f [principal]
/usr/bin/kinit -r renewable_life [principal]
```

```
/usr/bin/kinit -R [principal]
/usr/bin/kinit -k [-t keytab_file][principal]
/usr/bin/kinit -c [cache_name] [principal]
/usr/bin/kinit -S service_name [principal]
```

Options

`-l lifetime` The `-l` option requests a ticket with the lifetime of the value defined in lifetime. The value for `life_time` must be followed immediately by one of the following delimiters:

- s - seconds
- m- minutes
- h- hours
- d- days

For example: `kinit -l 90m` for 90 minutes

You cannot mix units; a value of `3h30m` will result in an error.

If the `-l` option is not specified, the default ticket lifetime (configured by each site) is used. Specifying a ticket lifetime longer than the maximum ticket life (configured by each site) results in a ticket with the maximum lifetime.

`-s start_time` The `-s` option requests a postdated ticket, valid starting at `start_time`. Postdated tickets are issued with the invalid flag set, and need to be passed back to the KDC before use.

`-v` The `-v` option requests that the TGT in the cache be passed to the KDC for validation. If the ticket is within its requested time range, the cache is replaced with the validated ticket.

`-p` The `-p` option requests a proxiable ticket.

`-f` The `-f` option requests a forwardable ticket.

`-r renewable_life` The `-r` option requests renewable tickets, with a total lifetime of `renewable_life`. The duration is in the same format as the `-l` option, with the same delimiters.

`-R` The `-R` option requests renewal of the TGT. You cannot renew an expired ticket even if the ticket is still within its renewable life.

`-k [-t keytab_file]` The `-k` option requests a host ticket obtained from a key in the local host's keytab file. You can specify the name and location of the keytab file with the `-t keytab_file` option; otherwise the default name and location will be used.

The default credentials cache can vary between systems. If the `KRB5CCNAME` environment variable is set, its value is used to name the default ticket cache. Any existing contents of the cache are destroyed by `kinit`.

`-c [cache_filename]` The `-c` option uses `cache_name` as the credentials (ticket) cache name and location; otherwise, the default cache name and location will be used.

`-S service_name` The `-s` option specifies an alternate service name to get initial tickets.

`Principal` The `Principal` uses the principal name from an existing cache, if there is one.

The `kinit` utility supports the [appdefaults] section. The relationships specified here can be over-ridden by the command-line options. The following relationships are supported by `kinit` in the [appdefaults] section:

forwardable This relationship specifies if an user can obtain a forwardable ticket. Valid values with which it can be set are `true`, `false`, `yes`, `y`, `no`, `n`, `on`, and `off`.

proxiable This relationship specifies if a user can obtain a proxiable ticket. Valid values to which it can be set are `true`, `false`, `yes`, `y`, `no`, `n`, `on`, and `off`.

tkt_lifetime This relationship specifies the lifetime of the ticket to be obtained. The unit of lifetime is either seconds, minutes, hours or days.

renew_lifetime This relationship specifies the renewable life of the ticket to be obtained. The unit of lifetime is either seconds, minutes, hours or days.

NOTE For DCE operations use `/opt/dce/bin/kinit`.

Reference To view the kinit manpage, issue the following command:

```
$ man 1 kinit
```

The klist Utility

Description The `klist` utility lists the Kerberos principal and Kerberos tickets held in a credentials cache, or the keys held in a keytab file.

Synopsis

```
/usr/bin/klist [-e]
/usr/bin/klist [-c] [cache_name]
/usr/bin/klist [-f] [cache_name]
/usr/bin/klist [-s] [cache_name]
/usr/bin/klist [-k] [keytab_name]
/usr/bin/klist [-t] [keytab_name]
/usr/bin/klist [-K] [keytab_name]
```

Options

`-e` The `-e` option displays the encryption types of the session key and the ticket for each credential in the credential cache, or each key in the keytab file.

`-c` The `-c` option lists tickets held in a credentials cache. This is the default if neither `-c` nor `-k` is specified.

`-f` The `-f` option shows the flags present in the credentials, using the following abbreviations:

- F - forwardable
- f - forwarded
- P - Proxiable
- p - proxy
- D - postDateable
- d - postdated
- R- Renewable

- I - Initial
- i - invalid

`-s`	The `-s` option sets exit status without `klist` output.
`-k`	The `-k` option lists keys held in a keytab file.
`-t`	The `-t` option displays the time entry timestamps for each keytab entry in the keytab file.
`-K`	The `-K` option displays the value of the encryption key in each keytab entry in the keytab file.

Reference To view the `klist` manpage, issue the following command:

```
$ man 1 klist
```

The kdestroy Utility

Description The `kdestroy` utility destroys the user's active Kerberos authorization tickets by writing zeros to the specified credentials cache that contains them. If the credential cache is not specified, the default credential cache is destroyed.

A user's credentials are not automatically removed by exiting from a SHELL or logging out. You need to remove the credential cache files manually before logging out using the `kdestroy` command.

If you use the `csh` shell, you can include `kdestroy` in the `.logout` file in your home directory. Additionally, the system administrator can remove expired credential cache files using either a `start` script or a `cron` job to recover disk space and prevent maliciously access to the network credentials.

Synopsis

```
/usr/bin/kdestroy [-q]
/usr/bin/kdestroy [-c] [cache_name]
```

Options

`-q`	The `-q` option suppresses beeps if it fails to destroy the user's tickets.
`-c`	The `-c` option uses `cache_name` as the credentials (ticket) cache name and location; if cache_name is not specified, the default cache name and location are used.

Reference To view the `kdestroy` manpage, issue the following command:

```
$ man 1 kdestroy
```

The kpasswd Utility

Description

The kpasswd utility changes a user's Kerberos password.

If the optional parameter principal is not used, kpasswd uses the principal name from an existing cache if there is one. If not, the principal is derived from the identity of the user by invoking kpasswd.

The kpasswd utility prompts for the current Kerberos password that is used to obtain a changepw ticket from the KDC for the user's Kerberos REALM. If kpasswd successfully obtains the changepw ticket, the user is prompted twice for a new password to make the password change.

Use kpasswd for your MIT KDC server only, not for Microsoft 2000 KDC. Also, note that kpasswd only changes Kerberos passwords on the KDC, not the UNIX password. Use the UNIX passwd command to change your UNIX password on the /etc/passwd file.

Synopsis

```
/usr/bin/kpasswd [principal]
```

Reference

To view the kpasswd manpage, issue the following command:

```
$ man 1 kpasswd
```

The ktutil Utility

Description

The ktutil utility maintains the keytab files. It is restricted only for system administrator's use.

Synopsis

```
/usr/sbin/ktutil
ktutil: list (Alias: l)
ktutil: read_kt keytab (Alias: rkt)
ktutil: read_st srvtab (Alias: rst)
ktutil: write_kt keytab (Alias: wkt)
ktutil: write_st srvtab (Alias: wst)
ktutil: clear_list (Alias: clear)
ktutil: delete_entry slot (Alias: delete)
ktutil: list_requests (Alias: lr or ?)
```

```
ktutil: quit (Alias: exit or q)
```

Options

`list` (Alias: `l`) The `l` option displays the current keylist.

`read_kt keytab` (Alias: `rkt`) The `rkt` option reads the Kerberos V5 keytab file keytab into the current keylist.

`read_st srvtab` (Alias: `rst`) The `rst` option reads the Kerberos V4 server KEYTAB file server `keytab` into the current keylist.

`write_kt keytab` (Alias: `wkt`) The `wkt` option writes the current keylist into the Kerberos V5 keytab file keytab.

`write_st srvtab` (Alias: `wst`) The `wst` option writes the current keylist into the Kerberos V4 server keytab file.

`clear_list` (Alias: `clear`) The `clear` option clears the current keylist.

`delete_entry slot` (Alias: `delete`) The `delete` option deletes the entry in slot number *`slot`* from the current keylist.

`list_requests` (Alias: `lr or ?`) The `list_request` option displays a list of available commands.

Reference

To view the `ktutil` manpage, issue the following command:

```
$ man 1 ktutil
```

The kvno Utility

Description

The `kvno` utility acquires a service ticket for the specified Kerberos principals to return key version numbers of Kerberos principals.

The `kvno` utility uses the environment variable `KRB5CCNAME`, which records the location of the credentials (ticket) cache.

Synopsis

```
/usr/bin/kvno [-e etype] service1, [service2,...]
```

Options

`-e etype` Specifies the encryption type which will be requested for the session key of all the services named on the command line. This is useful in certain backward compatibility situations. The value of `etype` can be one `DES-CBC-CRC`, `DES-CBC-RAW` or `DES-CBC-MD5`.

`[service1], [service2]` Service name(s) or principal name(s).

To view the `kvno` manpage, issue the following command:

```
$ man 1 kvno
```

HP Kerberos Server

Kerberos Server ensures secure communication in a networked environment by leveraging individual trust relationships. It then brokers that trust across enterprise wide, distributed client-server networks.

Table 2-7 lists the various versions of Kerberos Server available for different HP-UX operating systems.

Table 2-7 Versions of Kerberos Server on HP-UX Operating Systems

Kerberos Server Version	HP-UX OS Version
Kerberos Server v 2.1	HP-UX 11.0 and HP-UX 11i v1
Kerberos Server v3.1	HP-UX 11i v2
Kerberos Server v3.12	HP-UX 11i v3

Kerberos Server Version 3.12 Features

This version of the Kerberos Server offers the following features on HP-UX 11i v3:

- "Graphical User Interface (GUI) Based Administration tool" on page 65
- "Multithreaded Server" on page 65
- "High Availability" on page 65
- "Dynamic Propagation" on page 66
- "Scalability" on page 66
- "Windows 2000(R) Interoperability" on page 66
- "Choice of C-Tree or LDAP Database" on page 67
- "Auto-Configuration Tool" on page 67

This version of the Kerberos server integrates with the existing Kerberos clients on HP-UX 11i, thus providing the end user a full fledged security solution.

Kerberos server v3.12 supersedes the earlier MIT based Kerberos server (version 1.0), on HP-UX 11i. This version of the Kerberos server offers many enhancements when compared to the previous version.

For information on previous Kerberos Server versions, see the Release Notes at www.docs.hp.com/en/internet.html#Kerberos.

Graphical User Interface (GUI) Based Administration tool

Use the GUI to create and manage principals in the Kerberos Realms. This includes both the remote administrator, `kadmin_ui`, and the local administrator, `kadminl_ui`. Following are the functions you can perform using the GUI:

- create, modify and delete principals
- alter principal account key type settings
- assign administrative permissions
- modify the default group principals
- extract keys of principals to service key table files
- change the principal's password
- add a new realm or delete existing realms

Multithreaded Server

Kerberos server version 3.12 is a pre-threaded concurrent server. This feature enables the server to service multiple user requests in the KDC, thus enhancing the performance of the server. The server uses kernel space threads.

High Availability

The Kerberos server daemon (`kdcd`) is constantly monitored by a parent process. If the child process dies or crashes, the parent process automatically spawns a new server daemon. This provides for high availability in the case of mission critical applications.

In addition, it allows for multiple secondary security servers to be configured. The secondary security server services authentication requests, once it has been configured to authenticate and receive information propagated from the primary security server. This enables load balancing for the primary server, with automatic incremental propagation, without any performance degradation.

The secondary security server also provides redundancy against a single point of failure. The Kerberos Server also allows administrators to organize realms according to the types of users or services.

Dynamic Propagation

In Kerberos server version 1.0, the entire database had to be periodically dumped and propagated. This resulted in heavy network traffic and thus reduced performance.

It is important that secondary servers are configured to act as authentication servers. This allows the primary server to be available for tasks other than authentication. When a secondary server is configured, both the servers must be synchronized with each other. If entries are updated on the primary server, they must be updated on the secondary server as well. The databases on the primary and the secondary servers are synchronized by a mechanism called 'propagation'. The `kpropd` daemon running on the primary server ensures that the data is synchronized with the other secondary server.

Kerberos Server version 3.12 also supports hierarchical propagation. The primary server need not propagate the database to all the secondary servers in the realm, except for a designated secondary server. This designated secondary server will then propagate the database to the other secondary servers available in the realm. This is possible by defining such a propagation hierarchy in the configuration files.

Scalability

This version of the Kerberos Server is highly scalable, and has been tested to support up to 2,000,000 (two million) users in the database. In addition, it supports simultaneous requests from multiple clients and ensures that these queries are not lost even when the system is heavily loaded.

Windows 2000(R) Interoperability

To enable the user to work in a mixed platform environment, this version of the Kerberos Server is interoperable with the Windows 2000 Server(R) and client. A Kerberos Server in the Windows 2000(R) environment can talk to the HP-UX Kerberos server, for cross-realm authentication.

Choice of C-Tree or LDAP Database

Kerberos server version 3.12 allows you to use a C-Tree or an LDAP database as the backend database. By integrating the Kerberos principals with the corresponding users in the LDAP directory, you store data in a common repository. For more information, see *Kerberos Server Version 3.12 Administrator's Guide (5991-7686)* on www.docs.hp.com.

Auto-Configuration Tool

An automated tool named, `krbsetup`, has been provided to auto-configure your Kerberos Server. Using this tool, you can configure, unconfigure, start, and stop the `kdcd` and the `kadmind` daemons. This tool is installed in the following directory:

```
/opt/krb5/sbin
```

The `krbsetup` tool automatically creates your configuration files, `krb.conf` and `krb.realms`, `kpropd.ini` files and places them in the `/opt/krb5` directory. The sections in the configuration files is set to its default values. If you want to customize these sections, you must manually edit the configuration files and restart the `kdcd` and `kadmind` daemons using this tool.

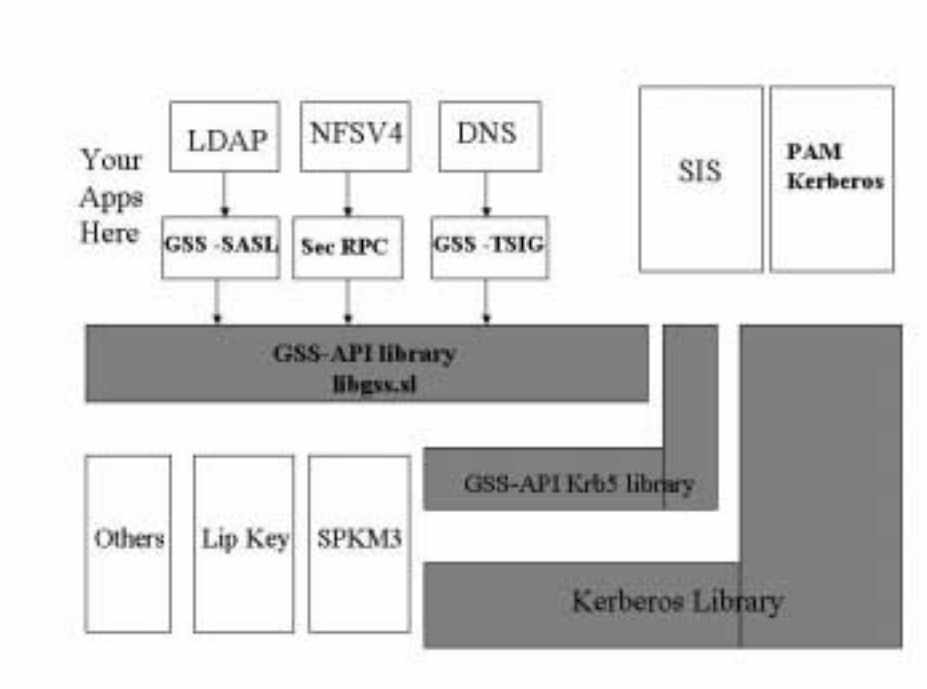
You can also use the `krbsetup` tool to configure your secondary security servers.

Generic Security Service Application Programming Interface (GSS-API)

The GSS-API provides authentication, integrity, and confidentiality services to the calling application.

Figure 2-4 shows the `libgss.sl` shared library, which is independent of underlying security mechanisms. The figure illustrates how the underlying security mechanisms -- such as Kerberos, Simple Public Key Management (SPKM) -- work with respect to the GSS-API library. If you are developing applications using GSS-APIs, you do not have to change the application's code whenever the underlying security mechanism is changed. Instead, you can change the underlying security mechanism at runtime using the configuration options.

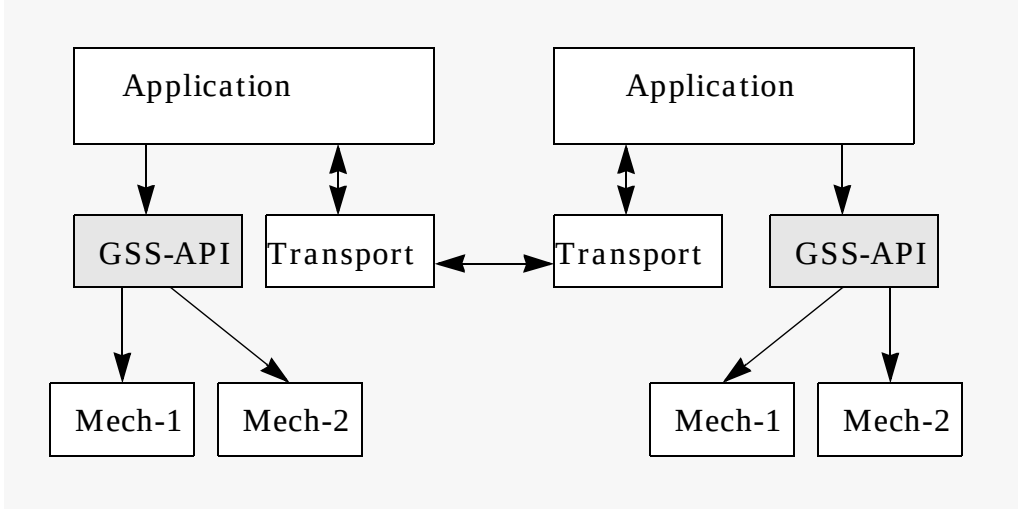
Figure 2-4 GSS-API Library



With an Open System architecture, GSS-API provides portability in a heterogeneous environment. It contains all the GSS-APIs specified in RFC 2743. It is implemented as a package of C-language interfaces as defined in *RFC 2744, Generic Security Service API: C-bindings*. The Kerberos Version 5 GSS-API Mechanism is explained in RFC 1964.

GSS-API provides secure communication between two peers with a security context established by an exchange of tokens. As shown in Figure 2-5, GSS-API is independent of communication protocols. The GSS-API libraries on the two hosts are responsible for creating and processing the tokens, but the application is responsible for transporting the tokens between the client and the server.

Figure 2-5 GSS-API Operation



It is the GSS-API caller's responsibility to transfer GSS-API-provided data element to the peer end to parse communicated messages, and to separate GSS-API related data elements from caller-provided data. GSS-API provides either context level tokens or per-message tokens for the caller to transport and get the results.

GSS-API filesets are listed in Table 2-8 and Table 2-9.

Table 2-8 GSS-API Libraries

Library Availability	Functionality
• Itanium® 32 - `/usr/lib/hpux32/libgss.so` • PA-RISC 32 - `/usr/lib/libgss.sl` • Itanium® 64 - `/usr/lib/hpux64/libgss.so` • PA-RISC 64 - `/usr/lib/pa20_64/libgss.sl`	This is the front-end GSS-API library, which has all the GSS-APIs

Table 2-9 Additional files in the GSS-API product

File Types	File Names
Header files	`/usr/include/gssapi/gssapi.h` `/usr/include/gssapi.h` (a link to `/usr/include/gssapi/gssapi.h`) `/usr/include/gssapi_ext.h` (a link to `/usr/include/gssapi/gssapi_ext.h`)
Configuration	`/etc/gss/mech`
	`/etc/gss/qop`
	`/etc/gss/gsscred.conf`
Examples	In the `/usr/contrib/gssapi/sample` directory
	`/usr/contrib/gssapi/sample/README` - README file for the samples
manpages	The English-language manpages for all the GSS-APIs are under `/usr/share/man/man3.Z`. (The manpages in the Japanese manpage filesets are also available in English.)

Following lists the services that the GSS-API interface provides:

• “Credential Management Services” on page 71

- "Context Level Services" on page 71
- "Authentication Services" on page 72
- "Confidentiality Service" on page 72
- "Support Services" on page 72

Credential Management Services

Credential management function calls acquire and release credentials by principals.

Applications are responsible for establishing a security mechanism based on the initial credentials. GSS-API mechanisms are responsible for management of credentials on the local machine.

The GSS-API function calls for credential management are:

- `gss_acquire_cred`: Obtain credentials for use
- `gss_release_cred`: Release credentials after use
- `gss_add_cred`: Adds credential elements incrementally
- `gss_inquire_cred`: Display information about credentials

Context Level Services

Context level function calls manage security context between peers. A context's initiator calls `gss_init_sec_context()`, resulting in generalization of a token that the caller passes to the target. The target then passes the token to `gss_accept_sec_context()`. It can take multiple exchanges of tokens to establish the security context depending on the options used.

The GSS-API context level function calls are:

- `gss_init_sec_context`: Initiate outbound security context
- `gss_accept_sec_context`: Accept inbound security context
- `gss_delete_sec_context`: Remove context that is no longer needed
- `gss_export_sec_context`: Transfer context to other process
- `gss_import_sec_context`: Import context from other process
- `gss_inquire_context`: Display information about context

- `gss_context_time`: Indicate validity time remaining in context

Authentication Services

Two sets of per-message calls provide security to the context. The `gss_get_mic()` and `gss_verify_mic()`function calls provide data origin authentication and data integrity services. The `gss_wrap()` and `gss_unwrap()` function calls support caller requested confidentiality. For more information, see “Confidentiality Service” on page 72.

The `gss_get_mic()` function call generates a token. The peer that receives the application data along with the message token verifies the communication using `gss_verify_mic()`.

Each deployment can select their own configurable Quality Of Protection (QOP) options.

In summary, per-message calls that authenticate messages are:

- `gss_get_mic`: Apply integrity check, receive as token separate from message
- `gss_verify_mic`: Validate integrity check token along with message.

Confidentiality Service

GSS-APIs provide confidentiality with the `gss_wrap()` and `gss_unwrap()` functions. The output of `gss_wrap()` is passed to the remote peer encapsulated and optionally encrypted with the associated token. This data element is an input to `gss_unwrap()` at the target, where it is decapsulated or optionally decrypted.

In summary, the APIs for confidentiality service include the following:

- `gss_wrap`: Sign, optionally encrypt, encapsulate
- `gss_unwrap`: Decapsulate, decrypt if needed, validate integrity check.

Support Services

GSS-API support services include the following APIs:

- `gss_display_status`: Translate status codes into printable format
- `gss_indicate_mechs`: Indicate supported mech_type on local system

- `gss_compare_name`: Compare two names
- `gss_display_name`: Translate name to printable format
- `gss_import_name`: Convert printable name to normalized form
- `gss_release_name`: Free storage of name
- `gss_release_buffer`: Free storage of general GSS-allocated object
- `gss_release_OID_set`: Free storage of OID set object
- `gss_create_empty_OID_set`: Create empty OID set
- `gss_add_OID_set_member`: Add member to OID set
- `gss_test_OID_set_member`: Test if OID is a member of a OID set
- `gss_inquire_names_for_mech`: Indicate name types supported
- `gss_inquire_mechs_for_name`: Indicates mechanisms supporting name type
- `gss_canonicalize_name`: Translate name to per mechanism form
- `gss_export_name`: Externalize per-mechanism name
- `gss_duplicate_name`: Duplicate name object
- `gss_inquire_cred_by_mech`: Provides per-mechanism information about a credential
- `gss_process_context_token`:
- `gss_wrap_size_limit`: Determines a token-size limit for `gss_wrap` in a context

3 Configuring the Kerberos Environment

This chapter describes the files and procedures that are used to configure Kerberos on HP-UX.

It contains the following sections:

- "Configuration Files for Kerberos Clients" on page 77
- "Configuration Files for GSS-API" on page 82
- "Configuring the Kerberos Server" on page 85
  - "Configuring Your Microsoft Windows 2000 KDC" on page 85
- "Configuring the Kerberos Client" on page 87
- "Configuring for PAM Kerberos" on page 88

Configuration Files for Kerberos Clients

Table 3-1 lists and describes the files that you use to configure a Kerberos server or a Kerberos client using PAM Kerberos. Samples of all the configuration files shown in the table are listed in the Appendices.

Table 3-1 Kerberos Configuration Files

Purposes	Kerberos Server	Kerberos Client	PAM Kerberos
Configure Kerberos as the module for authentication and password management	N/A	N/A	`/etc/pam.conf`
Specify the defaults and the location of the Kerberos server	`/opt/krb5/krb.conf`	`/etc/krb5.conf`	`/etc/krb5.conf`
Associate the Kerberos services with the ports	`/etc/services`	`/etc/services`	`/etc/services`
Specify Kerberos configuration information including defaults used to issue Kerberos tickets	`/opt/krb5/krb.conf` `/opt/krb5/krb.realms`	N/A	N/A

pam.conf

The configuration file `/etc/pam.conf` controls the behavior of the PAM modules. The `pam.conf` file contains a listing of system entry services, each of which is paired with its corresponding service module. When a service is requested, its associated module is invoked.

Each entry has the following format:

```
<service_name> <module_type> <control_flag> <module_path> <options>
```

The following is a sample entry for PAM Kerberos in the `pam.conf` file on HP-UX 11.0 and 11i v1:

```
login  auth required  /usr/lib/security/libpam_krb5.1 debug

ftp   auth required  /usr/lib/security/libpam_unix.1
```

The following is a sample entry for PAM Kerberos in the `pam.conf` file on HP-UX 11i v2 and HP-UX 11i v3:

```
login  auth required  libpam_krb5.so.1 debug

ftp   auth required   libpam_unix.1
```

As mentioned in Chapter 2, "Introduction to the Kerberos Products and GSS-API," on page 31 the PAM Kerberos module provides functionality for the authentication (*auth*), and password management (*password*) modules.

Using either the `required`, `optional`, or `sufficient` option, the *`control_flag`* field determines the priority and behavior of the modules stacked for a *`module_type`*. For example,

```
login  auth sufficient  /usr/lib/security/libpam_krb5.1 debug

login  auth required  /usr/lib/security/libpam_unix.1
```

The PAM Kerberos options are `renewable=<time>`, `forwardable`, `proxiable`, `use_first_pass`, `try_first_pass`, `ignore`, and `debug`.

For more information, see the *pam. conf(4)* and the *pam_krb5(5)* manpages.

Appendix A, "Sample pam.conf File," on page 105 contains a sample `/etc/pam.conf` file.

In the HP-UX 11i version, a sample `pam.conf` file for Kerberos is available as `/etc/pam.krb5`.

krb5.conf

The krb5.conf file specifies the defaults for the REALM and Kerberos applications, mappings of the hostnames onto Kerberos REALMs, and the location of KDCs for Kerberos REALMs. Application clients depend on the configuration file /etc/krb5.conf to locate the REALM's KDC.

The [libdefaults] section of the krb5.conf file specifies various parameters for the Kerberos library. In order for the utility klist to work with PAM Kerberos, it must include "ccache_type = 2."

```
[libdefaults]
default_realm = KDC.SUBDOMAIN.DOMAIN.COM
default_tkt_enctypes = DES-CBC-CRC
default_tgs_enctypes = DES-CBC-CRC
ccache_type = 2
```

The [realms] section of the krb5.conf file specifies the KDC server and the Kerberos admin server, kadmind that manages the administration interface to KDC.

The default ports used by Kerberos are port 88 for the KDC, port 749 for the kadmin service, and port 751 for kpasswd. You can optionally choose to run on other ports, as long as the ports are specified in each host's /etc/services, and in the krb5.conf files.

```
[realms]
   KDC.SUBDOMAIN.DOMAIN.COM = {
      kdc = hostname.subdomain.domain.com:88
      admin_server = hostname.subdomain.domain.com:749

      }
```

To configure for multiple Kerberos REALMs, list them in the order of priority, as in the following example:

```
[libdefaults]
default_realm = KDC1.SUBDOMAIN.DOMAIN.COM
default_tkt_enctypes = DES-CBC-CRC
default_tgs_enctypes = DES-CBC-CRC
ccache_type = 2


[realms]
   KDC1.SUBDOMAIN.DOMAIN.COM = {
      kdc = hostname1.subdomain.domain.com:88
      admin_server = hostname1.subdomain.domain.com:749
      }
   KDC2.SUBDOMAIN.DOMAIN.COM = {
      kdc = hostname2.subdomain.domain.com:88
      admin_server = hostname2.subdomain.domain.com:749
```

```
      }
[domain_realm]
      .subdomain.domain.com = KDC1.SUBDOMAIN.DOMAIN.COM
      .subdomain.domain.com = KDC2.SUBDOMAIN.DOMAIN.COM
```

The ldapux_multidomain Option

The `ldapux_multidomain` option needs to be set to `1` by the administrator if the realm name of the user needs to be obtained from the W2K multidomain. See the *ldapux (5)* manpage for more information to configure W2K multidomain.

The appdefaults Section

The appdefaults section denotes the default values used by Kerberos V5 applications.

Each tag in the `[appdefaults]` section names a Kerberos V5 application. The value of the tag is a subsection with relations that define the default behaviors for that application. For example:

```
[appdefaults]
kinit = {
   forwardable = true
}
```

You can find the list of options for each application in the respective application manpages. The application defaults specified in this section are overridden by those specified in the `[realms]` section.

See the *krb5.conf(4)* manpage for more information.

Appendix B, “Sample krb5.conf File,” on page 111 contains a sample copy of the `/etc/krb5.conf` file.

In the HP-UX 11i version of the operating system, a sample `krb5.conf` file is available as `/etc/krb5.conf.sample`.

The services File

The `services` file contains entries that allow client applications to establish socket connections to the KDC or to the application servers. A Kerberos client requires the following entries in the `/etc/services` file:

```
#
# PAM Kerberos services
#
```

```
kerberos5    88/udp   kdc              # Kerberos authentication
kerberos5    88/tcp   kdc              # Kerberos authentication
kerberos-adm 749/tcp  kerberos_adm     # Kerberos admin/changepw
kerberos-cpw 751/tcp  kerberos_master  # Kerberos changepw
krb5_prop    754/tcp                  # Kerberos slave propogation
```

For more information on services, see *services(4)*.

Configuration Files for GSS-API

Following configuration files are essential for proper functioning of GSS-API:

- "The mech File" on page 82
- "The /etc/gss/qop File" on page 83
- "The gsscred.conf File" on page 84

NOTE IPv6 support for GSS-API has been enabled only for the Itanium® binaries on HP-UX 11i v2 and HP-UX 11i v3 systems.

The mech File

The mechanism file, or `mech` file (`/etc/gss/mech`) specifies the underlying security mechanism. Table 3-2 lists and describes the entries in the `mech` file.

Table 3-2 Entries in the mech file

Column	Description
First column	Contains the names of the back-end security mechanism that support GSSAPI.
Second column	Contains the Object Identifier (OID).

Table 3-2 Entries in the mech file (Continued)

Column	Description
Third column	Contains the name of the shared library that implements the back-end security mechanism for GSSAPI. The back-end library must be placed in the `/usr/lib/gss` path for 32-bit and the `/usr/lib/pa20_64/gss` path for 64-bit versions on PA-RISC based systems. The back-end library has to be placed in the `/usr/lib/hpux32/gss` path for 32-bit and the `/usr/lib/hpux64/gss` path for 64-bit versions on Itanium based systems.
Fourth column	This is an optional field. In HP-UX 11i v3, this field lists the `krb5` kernel module.

You can use the `GSSAPI_MECH_CONF` environment variable to change the path of the mechanism file (`/etc/gss/mech`) file.

Example mech File on HP-UX 11.0 and HP-UX 11i v1

```
# Mechanism Name   Object Identifier      Shared Library
#
krb5_mech          1.2.840.113554.1.2.2  libgssapi_krb5.sl
```

Example mech File on HP-UX 11i v2

```
# Mechanism Name    Object Identifier    Shared Library
#
krb5_mech          1.2.840.113554.1.2.2  libgssapi_krb5.so
```

Example mech File on HP-UX 11i v3

```
# Mechanism Name   Object Identifier      Shared Library     Kernel Module
#
krb5_mech          1.2.840.113554.1.2.2  libgssapi_krb5.so  krb5
```

The /etc/gss/qop File

The `/etc/gss/qop` file contains information about the GSSAPI-based Quality Of Protection (QOP) for each underlying security mechanism.

QOP values are used with the Kerberos V5 GSS-API mechanism as input to `gss_wrap()` and `gss_get_mic()` in order to select alternate integrity and confidentiality algorithms.

Table 3-3 shows the format of the `/etc/gss/qop` file:

Table 3-3 Format of the /etc/gss/qop file

Column	Description
First column	Specifies the string name of QOP.
Second column	Contains its QOP value (32-bit integer).
Third column	Contains names of the security mechanism.

Following is a sample `/etc/gss/qop` file:

```
# QOP string                    QOP Value       Mechanism Name
#
GSS_KRB5_INTEG_C_QOP_DES_MD5 0                  krb5_mech
```

The gsscred.conf File

Use the `gsscred.conf` file to determine the underlying `gsscred` backend used to store the `gsscred` table. In HP-UX, it must contain an entry only as files.

Following is a sample `/etc/gss/gsscred.conf` file:

```
# gsscred configuration file
#
# Valid gsscred backend mechanisms are
# files
#
files
```

Configuring the Kerberos Server

You can configure a Kerberos *client* in the same way whether your KDC server is a Kerberos server on HP-UX 11i or a Microsoft 2000 KDC server. However, for a Microsoft Windows 2000 KDC server or the Kerberos server on HP-UX 11i, the *server* configuration procedures are different. To configure a Microsoft Windows 2000 KDC server or Kerberos server on HP-UX 11i, you must follow the KDC Server configuration instructions accompanied with your server software.

You can configure your Kerberos server with C-Tree or LDAP as the backend database. For instructions on configuring HP's Kerberos Server, see *Kerberos Server Version 3.12 Administrator's Guide (5991-7686)* available on www.docs.hp.com.

Configuring Your Microsoft Windows 2000 KDC

To configure your Microsoft Windows 2000 KDC, complete the following steps:

1. Use the Active Directory Management tool to create a new account for the UNIX host:
   - From *Administrators Tools*, select *Active Directory Users and Computers*.
   - Select the *Users* folder, select *Action* from the top menu, click *New*, then click *User*.
   - Add the name of a UNIX host as a user by entering the *hostname* as the user name, and host/*hostname* as user logon name.
2. Create a keytab file for the Kerberos client on Microsoft Windows 2000 KDC.
   - Locate ktpass on Microsoft Windows 2000
   - Use ktpass to create the KEYTAB file and set up the account for the UNIX host.

     ```
     C:> ktpass -princ host/hostname@NT-DNS-REALM-NAME
     -mapuser hostname -pass your-password -out
     hostname.keytab
     ```

     where:

— *hostname* is the unix host DNS name.

— *NT-DNS-REALM-NAME* is the uppercase name of the Windows 2000 domain. All domain names should be in upper case.

— *your-password* is the password for this principal, *hostname*.

This step creates an account in the name of host/hostname.subdomain.domain.com.

3. Follow step 3 under "Configuring the Kerberos Client" on page 87 to merge the KEYTAB file at the Kerberos client system.
4. For each user in the Kerberos client, create a Kerberos principal in the KDC Server:
   - From **Administrators Tools**, select the *Active Directory Users and Computers*.
   - Select the *Users* folder, select *Action* from the top menu, click *New*, then click *User*.
   - Add the name of each UNIX user by entering the user's first and last name, login name, and user's password.

Configuring the Kerberos Client

To configure the Kerberos Client, complete the following steps:

1. Edit the configuration files, `/etc/krb5.conf` and `/etc/services` as described in "Configuration Files for Kerberos Clients" on page 77.
2. All Kerberos systems need a KEYTAB file (`/etc/krb5.keytab`) to authenticate themselves to the KDC. Create a KEYTAB file for each KDC client on your KDC Server.
3. Transfer (`ftp`) the KEYTAB file from the KDC Server to the client without overwriting any keys installed for other applications. For example, use `/tmp/hostname.keytab` as the temporary destination filename. Use the Kerberos utility `ktutil` to merge the KEYTAB data.

   The following example shows how to merge the keytab using `ktutil`:

   ```
   $ /usr/sbin/ktutil

   ktutil: rkt /tmp/hostname.key

   ktutil: list
   ```

   You can view the KEYTAB file using `klist` command. For example:

   ```
   # klist -k

   Keytab name: FILE:/etc/krb5.keytab

   KVNO Principal

   ----------------------------------------------------------
   -
   2 host/hostname.domain.com@KDC.SUBDOMAIN.DOMAIN.COM
   ```

4. If the UNIX users do not exist, add the equivalent KDC users as UNIX users in the UNIX `/etc/passwd` password file. When creating a credential file for a user, the user's entry in the `/etc/passwd` is accessed for its `UID` number.
5. Synchronize the KDC client's clock to the KDC server's clock (within two minutes).

Configuring for PAM Kerberos

If you want to run PAM Kerberos, after you complete KDC client configuration from the previous section, you must edit the PAM configuration files `for PAM Kerberos`. Using the `/etc/pam.krb5` file as an example, edit the `/etc/pam.conf` as described in “Configuration Files for Kerberos Clients” on page 77.

4 Troubleshooting Kerberos Related Products

This chapter explains the error messages that you can encounter while using the Kerberos client products.

It contains the following sections:

- “Troubleshooting PAM Kerberos” on page 91
- “Troubleshooting the Kerberos Client Utilities” on page 94
- “Troubleshooting GSS-API” on page 96
- “Troubleshooting Using the pamkrbval Tool” on page 100

Troubleshooting PAM Kerberos

The PAM Kerberos module returns debug and error messages that are logged using the `syslog` utility. Use the appropriate `syslog` log levels to gather more information about error scenarios.

Debug logging is enabled using the `debug` option in the `/etc/pam.conf` file for Kerberos PAM module, as shown in following example:

```
login  auth sufficient  /usr/lib/security/libpam_krb5.1 debug
```

When using the debug option, make sure you designate a log file for debugging by modifying the `/etc/syslog.conf` file. For example:

```
*.debug<tab>/var/adm/syslog/pam.log
```

You can instruct the `syslog` daemon, `/etc/syslogd`, to re-read its configuration file by sending it a HANGUP signal as follows:

```
kill -HUP ‘cat /var/run/syslog.pid‘
```

The `syslog` also contains all the authentication messages for ARPA services such as `ftp` and `telnet`. For more information, see the *syslogd(1M)* manpage.

In addition, the syslog contains PAM error codes from the `/usr/include/security/pam_appl.h` include file. Table 4-1 provides a list of error codes with the suggested corrective actions.

Table 4-1 Error Codes and Corrective Actions

Error No.	PAM Error Code	Meaning	Reason/ Corrective Actions
1	PAM_SYSTEM_ERR	System error	Generic System Error. See `syslog` outputs for specific information.
2	PAM_BUF_ERR	Memory buffer error	Ensure that sufficient system memory is available for all processes.
3	PAM_PERM_DENIED	No permission	Check the permissions/ACLs.

Table 4-1 Error Codes and Corrective Actions (Continued)

Error No.	PAM Error Code	Meaning	Reason/ Corrective Actions
4	PAM_AUTH_ERR	Authentication failure	The user's password may be wrong, or the host machine identity is not present, or the credential cache may not be writable.
5	PAM_CRED_INSUFFICIENT	Cannot access authentication data: insufficient credentials	
6	PAM_AUTHINFO_UNAVAIL	Authentication service not available	KDC Server is down or not reachable.
7	PAM_USER_UNKNOWN	User unknown to Kerberos service	Ensure that the user is present in Kerberos KDC.
8	PAM_CRED_UNAVAIL	Cannot retrieve user credentials	KRB5CCNAME is not set or the credential file does not exist or the user is not permitted to use the credential cache.
9	PAM_CRED_EXPIRED	User credentials expired	Credential expired. Re-initialize the credentials.
10	PAM_CRED_ERR	Failure setting user credentials	Check user's permissions to write to credential cache.
11	PAM_ACCT_EXPIRED	User account has expired	Ensure that the user's account is valid.
12	PAM_AUTHTOK_ERR	Authentication token manipulation error.	Check the password entered.

Table 4-1 Error Codes and Corrective Actions (Continued)

Error No.	PAM Error Code	Meaning	Reason/ Corrective Actions
13	PAM_AUTHTOK_RECOVERY_ERR	Authentication information cannot be recovered.	Old password is not correct.
14	PAM_TRY_AGAIN	Preliminary check by password service failed.	Try again.
15	OTHER Errors		See the *syslog(1M)* manpage for more specific information.

Troubleshooting the Kerberos Client Utilities

Kerberos utilities, `kdestroy`, `kinit`, `klist`, and `kpasswd` can return the following errors. Table 4-2 provides a list of errors with their meaning and suggested corrective actions for each error.

Table 4-2 Kerberos Client Error Codes

Error No.	Error	Meaning	Reason/Corrective Action
1	`kdestroy`: No credentials cache file found while destroying cache. Ticket cache not destroyed!	The credentials cache file was not found.	The credential file may have been deleted. Recreate the credentials (TGT) using `kinit`.
2	`kinit`: Key table entry not found while getting initial credentials.	The local keytab file does not contain the key for the principal whose credentials are being requested.	Add the principal key entry to the keytab file.
3	`kinit`: Client not found in Kerberos database while getting initial credentials.	The principal whose credentials are being requested does not exist in the Kerberos database.	Verify that there is a principal entry available for the client in the Kerberos database. If there is no entry, you must create it.
4	`klist`: No credentials cache file found	No credentials cache file was found.	This could be due to the deletion of credentials. Recreate the credentials using `kinit` or if the credential file is different from the one indicated by `klist`; then export the `KRB5CCNAME` environment variable to specify the correct filename.

Table 4-2 Kerberos Client Error Codes (Continued)

Error No.	Error	Meaning	Reason/Corrective Action
5	`klist`: No such file or directory while starting keytab scan	The keytab file was not found. (The default location of the keytab file is `/etc/krb5.keytab`.)	Verify the keytab file. If the keytab file does not exist, create the keytab file with specific entries. If the keytab file location is different from the default location, then use *-t* option to specify the correct location.
6	`klist`: Bad format in credentials cache while setting cache flags	The credential cache file is not in the proper format.	Reinitialize the credentials by using `kinit`.
7	`kpasswd`: New passwords do not match - password not changed.	The principal whose password you want to change does not exist in the Kerberos database.	Create a principal entry in Kerberos database.
8	`kpasswd`: Unknown credential cache type while reading principal name from credential cache.	The credential cache file is of an unknown type.	Credential file may be corrupted. Obtain the credentials using `kinit`.
9	KDC has no support for encryption type while getting credentials.	Encryption type requested for the session key is not supported.	Use the supported encryption type.
10	`kpasswd`: when `kpasswd` gets the principal from the cache file, it finds the principle in bad format.	Check user's permissions to write to credential cache.	Credential file may be corrupted. Obtain the credentials using `kinit`.

You can find Kerberos V5 Library Error Codes from Appendix A of MIT's *Kerberos V5 System Administrator's Guide.*

Troubleshooting GSS-API

This section provides troubleshooting tips for GSS-API.

Error Codes

It is the responsibility of the application programmer to check for the major and minor status values. For debugging purposes, HP recommends using the `gss_display_status()` function call for getting the textual representation of a GSS-API status code that can be displayed to a user or used for logging.

Major and Minor Status Values

Major status values are generic API routine errors or calling errors defined in RFC 2744.

Minor status values indicate mechanism-specific errors. Minor status values usually contain more detailed information about the error. They are not, however, portable between GSS-API implementations.

When designing portable applications, use major status values for handling errors. Use minor status values to debug applications and to display error and error-recovery information to users.

Common GSS-API Errors

Table 4-3 lists common GSS-API errors and their meanings:

Table 4-3 Common GSS-API Errors

Error No.	Name	Meaning
1	GSS_S_BAD_MECH	The required mechanism is unsupported.
2	GSS_S_BAD_NAME	The name passed is invalid.

Table 4-3 Common GSS-API Errors (Continued)

Error No.	Name	Meaning
3	GSS_S_BAD_NAMETYPE	The name type passed is unsupported.
4	GSS_S_BAD_BINDINGS	The channel bindings are incorrect.
5	GSS_S_BAD_STATUS	A status value is invalid.
6	GSS_S_BAD_SIG	A token has an invalid signature.
7	GSS_S_NO_CRED	No credentials are supplied.
8	GSS_S_NO_CONTEXT	No context established.
9	GSS_S_DEFECTIVE_TOKEN	Invalid token.
10	GSS_S_DEFECTIVE_CREDENTIAL	Invalid credential.
11	GSS_S_CREDENTIALS_EXPIRED	The referenced credentials expired.
12	GSS_S_CONTEXT_EXPIRED	The context expired.
13	GSS_S_FAILURE	The routine failed.
14	GSS_S_BAD_QOP	The quality of protection requested cannot be provided.
15	GSS_S_UNAUTHORIZED	The operation is forbidden by local security policy.

Calling Error Values

Table 4-4 lists the calling error values and their meanings:

Table 4-4 Calling Errors

Error No.	Name	Meaning
1	GSS_S_CALL_INACCESSIBLE_READ	Cannot read a required input parameter.
2	GSS_S_CALL_INACCESSIBLE_WRITE	Cannot write a required output parameter.
3	GSS_S_BAD_STRUCTURE	Cannot structure parameter correctly.

Supplementary Bits

Table 4-5 lists the supplementary bit values and their meanings:

Table 4-5 Supplementary Bits

Bit No.	Name	Meaning
0	GSS_S_CONTINUE_NEEDED	Call the routine again to complete its function.
1	GSS_S_DUPLICATE_TOKEN	The token is a duplicate of an earlier token.
2	GSS_S_OLD_TOKEN	The token's validity period expired; the routine cannot verify that the token is not a duplicate of an earlier token.
3	GSS_S_UNSEQ_TOKEN	A later token has been processed.

Other Common Causes of Errors

Other common causes of errors include the following:

- If KRB5-Client product is not installed, you can get an error trying to use `gssapi` with `/etc/gss/mech` configured to `krb5_mech`.
- Improper permissions of the `libgssapi_krb5.sl` / `libgssapi_krb5.so` library.
- Specifying the full path of the backend library in the `/etc/gss/mech` (for example, when using the 64-bit library, one should not specify the library path as `/usr/lib/pa20_64/gss/libgssapi_krb5.sl`, but only as `libgssapi_krb5.sl`; then the 64-bit `libgss.sl` library will take care of linking it).
- Absence of GSS-API configuration files.
- In case of GSSAPI-SSPI interoperability, the entries must use the DES-CBC-MD5 encryption type instead of the default DES-CBC-CRC.

NOTE There is a sample GSS-API client-server application in the `/usr/contrib/gssapi/sample` directory that you can use for troubleshooting.

You can find additional GSS-API Error Codes from the Appendix A of MIT's *Kerberos V5 System Administrator's Guide*.

Troubleshooting Using the pamkrbval Tool

This section provides tips for troubleshooting with the `pamkrbval` tool. When you use the `pamkrbval` tool for troubleshooting, you can get error messages when validating the `keytab` file.

NOTE Use the `pamkrbval` command with the `-c` option to troubleshoot CIFS-related issues.

Table 4-6 lists various errors that can occur and provides methods to troubleshoot the errors.

Table 4-6 Error Messages that Appear During keytab Validation

Error/Warning Messages	Reason for Message	Troubleshooting
`[WARNING] : host/example.com@EXAMPLE.COM found on KDC but not found in keytab file` `[FAIL] : The keytab validation Failed`	The keytab validation has failed because the key table entry is not found in the client's keytab file. There is a host principal present at the KDC.	Extract the keytab entry for the host principal on your system.

`[WARNING] : Client not found in Kerberos database`
`[WARNING] : The keytab entry for the host service principal host/example.com@EXAMPLE.COM is invalid`
`[FAIL] : The keytab validation Failed`
	The keytab validation has failed because there is no keytab entry in the client's keytab file and KDC.	You must create the keytab entry on the Kerberos server and extract this keytab entry on your system.

Table 4-6 Error Messages that Appear During keytab Validation

Error/Warning Messages	Reason for Message	Troubleshooting

`[WARNING] : Key incorrect`
`[WARNING] : The keytab entry for the host service principal host/example.com@EXAMPLE.COM is invalid`
`[FAIL] : The keytab validation Failed`
	There is a key mismatch between the client and the server.	Get the new keytab entry with the correct key from the Kerberos server.

`/pamkrbval: Cannot contact any KDC for requested realm while getting TGT` `[FAIL]: The keytab validation failed`	The KDC is not accessible.	Check that the KDC daemons are running.
`[LOG] : The keytab entry for host/cherry.example.com is not found in keytab file /etc/krb5.keytab` `[FAIL]: The keytab validation failed`	The keytab entry for the host service principal is not available.	• You must create the keytab entry on the Kerberos server and extract this keytab entry on your system. • Regenerate the keytab file in the CIFS environment and check that the service key for host/fqdn is present in the file. Execute the following command to regenerate the keytab file: `net ads keytab create –U administrator`

Table 4-6 Error Messages that Appear During keytab Validation

Error/Warning Messages	Reason for Message	Troubleshooting
`[LOG] : The keytab entry for host/cherry is not found in keytab file /etc/krb5.keytab [FAIL]: The keytab validation failed`	The keytab entry for the host service principal is not available. This error only occurs in the CIFS environment.	Regenerate the keytab file in the CIFS environment and check that the service key for host/simple hostname is present in the file. Execute the following command to regenerate the keytab file: `net ads keytab create –U administrator`
`[LOG] : Key table entry not found in keytab file /etc/krb5.keytab, ignoring keytab entry validation [IGNORE]: The keytab validation is ignored, assuming success`	The keytab entry for the host service principal is not available.	• Extract the key from the Kerberos Server using the `kadminl` tool and copy it to your system. • In a CIFS environment, regenerate the keytab file and check that the service key for host/fqdn is present in the file. Execute the following command to regenerate the keytab file: `net ads keytab create –U administrator`

Table 4-6 Error Messages that Appear During keytab Validation

Error/Warning Messages	Reason for Message	Troubleshooting
`pamkrbval: Key version number for principal in key table is incorrect while reading request [FAIL]: The keytab validation failed`	The key has been changed on the server but has not been updated in the user's system.	• Extract the key from the Kerberos Server using the `kadminl` tool and copy it to your system. • In a CIFS environment, update the keytab file with the current service key by regenerating the keytab file again. Execute the following command to regenerate the keytab file: `net ads keytab create –U administrator`
`pamkrbval: Decrypt integrity check failed While getting TGT [FAIL]: The keytab validation failed`	The key has been changed on the server but has not been updated in the user's system.	• Extract the key from the Kerberos Server using the `kadminl` tool and copy it to your system. • In a CIFS environment, update the keytab file with the current service key by regenerating the keytab file again. Execute the following command to regenerate the keytab file: `net ads keytab create –U administrator`

A Sample pam.conf File

The file presented below is `/etc/pam.krb5`, a sample `pam.conf` file that comes with PAM Kerberos.

On HP-UX 11.0 and HP-UX 11i v1

```
#
# PAM configuration
#
# Authentication management
#
login     auth  sufficient /usr/lib/security/libpam_krb5.1
login     auth  required   /usr/lib/security/libpam_unix.1  try_first_pass
su        auth  sufficient /usr/lib/security/libpam_krb5.1
su        auth  required   /usr/lib/security/libpam_unix.1  try_first_pass
dtlogin   auth  sufficient /usr/lib/security/libpam_krb5.1
dtlogin   auth  required   /usr/lib/security/libpam_unix.1  try_first_pass
dtaction  auth  sufficient /usr/lib/security/libpam_krb5.1
dtaction  auth  required   /usr/lib/security/libpam_unix.1  try_first_pass
ftp       auth  sufficient /usr/lib/security/libpam_krb5.1
ftp       auth  required   /usr/lib/security/libpam_unix.1  try_first_pass
OTHER     auth  required   /usr/lib/security/libpam_unix.1
#
# Account management
#
login      account   required   /usr/lib/security/libpam_krb5.1
login      account   required   /usr/lib/security/libpam_unix.1
su         account   required   /usr/lib/security/libpam_krb5.1
su         account   required   /usr/lib/security/libpam_unix.1
dtlogin    account   required   /usr/lib/security/libpam_krb5.1
dtlogin    account   required   /usr/lib/security/libpam_unix.1
dtaction   account   required   /usr/lib/security/libpam_krb5.1
dtaction   account   required   /usr/lib/security/libpam_unix.1
ftp        account   required   /usr/lib/security/libpam_krb5.1
ftp        account   required   /usr/lib/security/libpam_unix.1
OTHER      account   required   /usr/lib/security/libpam_unix.1
#
# Session management
#
login      session   required   /usr/lib/security/libpam_krb5.1
login      session   required   /usr/lib/security/libpam_unix.1
dtlogin    session   required   /usr/lib/security/libpam_krb5.1
dtlogin    session   required   /usr/lib/security/libpam_unix.1
dtaction   session   required   /usr/lib/security/libpam_krb5.1
dtaction   session   required   /usr/lib/security/libpam_unix.1
OTHER      session   required   /usr/lib/security/libpam_unix.1
#
# Password management
#
login      password   required   /usr/lib/security/libpam_krb5.1
```

```
login      password   required   /usr/lib/security/libpam_unix.1
passwd     password   required   /usr/lib/security/libpam_krb5.1
passwd     password   required   /usr/lib/security/libpam_unix.1
dtlogin    password   required   /usr/lib/security/libpam_krb5.1
dtlogin    password   required   /usr/lib/security/libpam_unix.1
dtaction   password   required   /usr/lib/security/libpam_krb5.1
dtaction   password   required   /usr/lib/security/libpam_unix.1
OTHER      password   required   /usr/lib/security/libpam_unix.1
```

On HP-UX 11i v2 and HP-UX 11i v3

```
#
# PAM configuration
#
# Notes: This pam.conf file is intended as an example only.
# If the path to a library is not absolute, it is assumed to be
# relative to one of the following directories:
# /usr/lib/security         (PA 32-bit)
# /usr/lib/security/pa20_64 (PA 64-bit)
# /usr/lib/security/hpux32  (IA 32-bit)
# /usr/lib/security/hpux64  (IA 64-bit)
# The IA file name convention is normally used; for example:
# libpam_unix.so.1
# For PA libpam_unix.so.1 is a symbolic link to the PA library:
# ln -s libpam_unix.1 libpam_unix.so.1
# Also note that the use of pam_hpsec(5) is mandatory for some of the
# services. See pam_hpsec(5).
# Authentication management
#
login      auth sufficient   libpam_krb5.so.1
login      auth required     libpam_unix.so.1
try_first_pass
su         auth sufficient   libpam_krb5.so.1
su         auth required     libpam_unix.so.1
try_first_pass
dtlogin    auth sufficient   libpam_krb5.so.1
dtlogin    auth required     libpam_unix.so.1
try_first_pass
dtaction   auth sufficient   libpam_krb5.so.1
dtaction   auth required     libpam_unix.so.1
try_first_pass
ftp        auth sufficient   libpam_krb5.so.1
ftp        auth required     libpam_unix.so.1
try_first_pass
OTHER      auth sufficient   libpam_unix.so.1
#
# Account management
#
login      account required   libpam_krb5.so.1
login      account required   libpam_unix.so.1
su         account required   libpam_krb5.so.1
su         account required   libpam_unix.so.1
dtlogin    account required   libpam_krb5.so.1
dtlogin    account required   libpam_unix.so.1
dtaction   account required   libpam_krb5.so.1
```

```
dtaction   account required   libpam_unix.so.1
ftp        account required   libpam_krb5.so.1
ftp        account required   libpam_unix.so.1
OTHER      account sufficient libpam_unix.so.1
#
# Session management
#
login      session required   libpam_krb5.so.1
login      session required   libpam_unix.so.1
dtlogin    session required   libpam_krb5.so.1
dtlogin    session required   libpam_unix.so.1
dtaction   session required   libpam_krb5.so.1
dtaction   session required   libpam_unix.so.1
OTHER      session sufficient libpam_unix.so.1
#
# Password management
#
login      password sufficient libpam_krb5.so.1
login      password required   libpam_unix.so.1
passwd     password sufficient libpam_krb5.so.1
passwd     password required   libpam_unix.so.1
dtlogin    password sufficient libpam_krb5.so.1
dtlogin    password required   libpam_unix.so.1
dtaction   password sufficient libpam_krb5.so.1
dtaction   password required   libpam_unix.so.1
OTHER       password sufficient    libpam_unix.so.1
```

B Sample krb5.conf File

The following is a `/etc/krb5.conf.sample` file, which is provided with KRB5-Client from HP-UX 11i v2 onwards. You can modify this file for use as your own `krb5.conf` file. Replace the underlined

KDC.SUBDOMAIN.DOMAIN.COM and hostname.subdomain.domain.com with the name of your Kerberos REALM and hostname.

```
[libdefaults]
      default_realm = KDC.SUBDOMAIN.DOMAIN.COM
      default_tkt_enctypes = DES-CBC-CRC
      default_tgs_enctypes = DES-CBC-CRC
      ccache_type = 2
      checksum_type = 1

[realms]
    KDC.SUBDOMAIN.DOMAIN.COM = {
      kdc = hostname.subdomain.domain.com:88
      admin_server = hostname.subdomain.domain.com:749
      kpasswd_server = hostname.subdomain.domain.com
      }
[domain_realm]
      .subdomain.domain.com = KDC.SUBDOMAIN.DOMAIN.COM

[logging]
      kdc = FILE:/var/adm/krb5kdc.log
      admin_server = FILE:/var/adm/kadmin.log
      default = FILE:/var/adm/krb5lib.log
```

C Sample krb.conf File

The following is a sample `krb.conf.sample` file available in the following directory:

`/opt/krb5/example`

Copy this sample file to /opt/krb5/krb.conf file and modify it to reflect the hostnames and realm name of your realm.

Replace the underlined Your_Realm_Name, Your_Secondary_Server1, Your_Secondary_Server2 and hostname.subdomain.domain.com with the name of your Kerberos REALM, Primary and Secondary Servers hostnames.

```
Your_Realm_Name
Your_Realm_Name Your_Secondary_Server1
Your_Realm_Name Your_Secondary_Server2
Your_Realm_Name host.subdomain.domain.com admin server
```

Given below is an example with a brief explanation of the krb.conf file.

```
BAMBI.COM
BAMBI.COM fox.bambi.com
BAMBI.COM goat.bambi.com
BAMBI.COM deer.bambi.com admin server #
```

Where:

- the realm name is
  - BAMBI.COM
- the primary security server is
  - deer.bambi.com
- the secondary security server 1 is
  - fox.bambi.com
- the secondary security server 2 is
  - goat.bambi.com

D Sample krb.realms File

The following is a sample `krb.realms.sample` file available in the following directory:

`/opt/krb5/example`

Replace the underlined Your_Realm_Name, Your_Primary_Security_Server, Your_Secondary_Server_Server and Your_Domain_Name with the name of your Kerberos REALM, primary and secondary servers hostnames.

```
Your_Primary_Security_Server Your_Realm_Name      #
.Your Secondary Security Server Your Realm Name  #
*.Your Domain Name Your Realm Name                    #
```

Given below is an example with a brief explanation of the krb.realms file.

```
deer.bambi.com BAMBI.COM    # map host directly
.fox.bambi.com BAMBI.COM    # all hosts in domain
*.bambi.com BAMBI.COM       # all the other hosts belonging
                                to the domain and sub-domains
```

Line one of the krb.realms file maps the host admin.bambi.com to the BAMBI.COM realm.

Line two of the krb.realms file maps all hosts in the fox.bambi.com domain to the BAMBI.COM realm.

NOTE

The preceding dot in this line identifies the first field as a domain name rather than a hostname.

Typically, this line is not required as the realm name, by default, is the upper-case equivalent of the domain name.

Line three of the krb.realms file maps all hosts in the domain and sub-domains with the root name bambi.com to the BAMBI.COM realm.

E Kerberos Error Messages

The following is a list of Kerberos Error Messages that you might encounter while using the Kerberos server.

NOTE The error codes are denoted in capital letters, followed by their respective error message.

Kerberos V5 Library Error Codes

This is the Kerberos v5 library error code table. Protocol error codes are ERROR_TABLE_BASE_krb5 + the protocol error code number; other error codes start at ERROR_TABLE_BASE_krb5 + 128.

1. KRB5KDC_ERR_NONE: No error
2. KRB5KDC_ERR_NAME_EXP: Client's entry in database has expired
3. KRB5KDC_ERR_SERVICE_EXP: Server's entry in database has expired
4. KRB5KDC_ERR_BAD_PVNO: Requested protocol version not supported
5. KRB5KDC_ERR_C_OLD_MAST_KVNO: Client's key is encrypted in an old master key
6. KRB5KDC_ERR_S_OLD_MAST_KVNO: Server's key is encrypted in an old master key
7. KRB5KDC_ERR_C_PRINCIPAL_UNKNOWN: Client not found in Kerberos database
8. KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN: Server not found in Kerberos database
9. KRB5KDC_ERR_PRINCIPAL_NOT_UNIQUE: Principal has multiple entries in Kerberos database
10. KRB5KDC_ERR_NULL_KEY: Client or server has a null key
11. KRB5KDC_ERR_CANNOT_POSTDATE: Ticket is ineligible for postdating
12. KRB5KDC_ERR_NEVER_VALID: Requested effective lifetime is negative or too short
13. KRB5KDC_ERR_POLICY: KDC policy rejects request
14. KRB5KDC_ERR_BADOPTION: KDC can't fulfill requested option
15. KRB5KDC_ERR_ETYPE_NOSUPP: KDC has no support for encryption type

16. KRB5KDC_ERR_SUMTYPE_NOSUPP: KDC has no support for checksum type
17. KRB5KDC_ERR_PADATA_TYPE_NOSUPP: KDC has no support for padata type
18. KRB5KDC_ERR_TRTYPE_NOSUPP: KDC has no support for transited type
19. KRB5KDC_ERR_CLIENT_REVOKED: Clients credentials have been revoked
20. KRB5KDC_ERR_SERVICE_REVOKED: Credentials for server have been revoked
21. KRB5KDC_ERR_TGT_REVOKED: TGT has been revoked
22. KRB5KDC_ERR_CLIENT_NOTYET: Client not yet valid - try again later
23. KRB5KDC_ERR_SERVICE_NOTYET: Server not yet valid - try again later
24. KRB5KDC_ERR_KEY_EXP: Password has expired
25. KRB5KDC_ERR_PREAUTH_FAILED: Pre-authentication failed
26. KRB5KDC_ERR_PREAUTH_REQUIRED: Additional pre-authentiaction required
27. KRB5KDC_ERR_SERVER_NOMATCH: Requested server and ticket don't match
28. KRB5PLACEHOLD_27: KRB5 error code 27
29. KRB5PLACEHOLD_28: KRB5 error code 28
30. KRB5PLACEHOLD_29: KRB5 error code 29
31. KRB5PLACEHOLD_30: KRB5 error code 30
32. KRB5KRB_AP_ERR_BAD_INTEGRITY: Decrypt integrity check failed
33. KRB5KRB_AP_ERR_TKT_EXPIRED: Ticket expired
34. KRB5KRB_AP_ERR_TKT_NYV: Ticket not yet valid
35. KRB5KRB_AP_ERR_REPEAT: Request is a replay
36. KRB5KRB_AP_ERR_NOT_US: The ticket isn't for us

37. KRB5KRB_AP_ERR_BADMATCH: Ticket/authenticator don't match
38. KRB5KRB_AP_ERR_SKEW: Clock skew too great
39. KRB5KRB_AP_ERR_BADADDR: Incorrect net address
40. KRB5KRB_AP_ERR_BADVERSION: Protocol version mismatch
41. KRB5KRB_AP_ERR_MSG_TYPE: Invalid message type
42. KRB5KRB_AP_ERR_MODIFIED: Message stream modified
43. KRB5KRB_AP_ERR_BADORDER: Message out of order
44. KRB5KRB_AP_ERR_ILL_CR_TKT: Illegal cross-realm ticket
45. KRB5KRB_AP_ERR_BADKEYVER: Key version is not available
46. KRB5KRB_AP_ERR_NOKEY: Service key not available
47. KRB5KRB_AP_ERR_MUT_FAIL: Mutual authentication failed
48. KRB5KRB_AP_ERR_BADDIRECTION: Incorrect message direction
49. KRB5KRB_AP_ERR_METHOD: Alternative authentication method required
50. KRB5KRB_AP_ERR_BADSEQ: Incorrect sequence number in message
51. KRB5KRB_AP_ERR_INAPP_CKSUM: Inappropriate type of checksum in message
52. KRB5PLACEHOLD_51: KRB5 error code 51
53. KRB5PLACEHOLD_52: KRB5 error code 52
54. KRB5PLACEHOLD_53: KRB5 error code 53
55. KRB5PLACEHOLD_54: KRB5 error code 54
56. KRB5PLACEHOLD_55: KRB5 error code 55
57. KRB5PLACEHOLD_56: KRB5 error code 56
58. KRB5PLACEHOLD_57: KRB5 error code 57
59. KRB5PLACEHOLD_58: KRB5 error code 58
60. KRB5PLACEHOLD_59: KRB5 error code 59
61. KRB5KRB_ERR_GENERIC: Generic error (see e-text)

62. KRB5KRB_ERR_FIELD_TOOLONG: Field is too long for this implementation
63. KRB5PLACEHOLD_62: KRB5 error code 62
64. KRB5PLACEHOLD_63: KRB5 error code 63
65. KRB5PLACEHOLD_64: KRB5 error code 64
66. KRB5PLACEHOLD_65: KRB5 error code 65
67. KRB5PLACEHOLD_66: KRB5 error code 66
68. KRB5PLACEHOLD_67: KRB5 error code 67
69. KRB5PLACEHOLD_68: KRB5 error code 68
70. KRB5PLACEHOLD_69: KRB5 error code 69
71. KRB5PLACEHOLD_70: KRB5 error code 70
72. KRB5PLACEHOLD_71: KRB5 error code 71
73. KRB5PLACEHOLD_72: KRB5 error code 72
74. KRB5PLACEHOLD_73: KRB5 error code 73
75. KRB5PLACEHOLD_74: KRB5 error code 74
76. KRB5PLACEHOLD_75: KRB5 error code 75
77. KRB5PLACEHOLD_76: KRB5 error code 76
78. KRB5PLACEHOLD_77: KRB5 error code 77
79. KRB5PLACEHOLD_78: KRB5 error code 78
80. KRB5PLACEHOLD_79: KRB5 error code 79
81. KRB5PLACEHOLD_80: KRB5 error code 80
82. KRB5PLACEHOLD_81: KRB5 error code 81
83. KRB5PLACEHOLD_82: KRB5 error code 82
84. KRB5PLACEHOLD_83: KRB5 error code 83
85. KRB5PLACEHOLD_84: KRB5 error code 84
86. KRB5PLACEHOLD_85: KRB5 error code 85
87. KRB5PLACEHOLD_86: KRB5 error code 86
88. KRB5PLACEHOLD_87: KRB5 error code 87
89. KRB5PLACEHOLD_88: KRB5 error code 88

90. KRB5PLACEHOLD_89: KRB5 error code 89
91. KRB5PLACEHOLD_90: KRB5 error code 90
92. KRB5PLACEHOLD_91: KRB5 error code 91
93. KRB5PLACEHOLD_92: KRB5 error code 92
94. KRB5PLACEHOLD_93: KRB5 error code 93
95. KRB5PLACEHOLD_94: KRB5 error code 94
96. KRB5PLACEHOLD_95: KRB5 error code 95
97. KRB5PLACEHOLD_96: KRB5 error code 96
98. KRB5PLACEHOLD_97: KRB5 error code 97
99. KRB5PLACEHOLD_98: KRB5 error code 98
100. KRB5PLACEHOLD_99: KRB5 error code 99
101. KRB5PLACEHOLD_100: KRB5 error code 100
102. KRB5PLACEHOLD_101: KRB5 error code 101
103. KRB5PLACEHOLD_102: KRB5 error code 102
104. KRB5PLACEHOLD_103: KRB5 error code 103
105. KRB5PLACEHOLD_104: KRB5 error code 104
106. KRB5PLACEHOLD_105: KRB5 error code 105
107. KRB5PLACEHOLD_106: KRB5 error code 106
108. KRB5PLACEHOLD_107: KRB5 error code 107
109. KRB5PLACEHOLD_108: KRB5 error code 108
110. KRB5PLACEHOLD_109: KRB5 error code 109
111. KRB5PLACEHOLD_110: KRB5 error code 110
112. KRB5PLACEHOLD_111: KRB5 error code 111
113. + KRB5PLACEHOLD_112: KRB5 error code 112
114. KRB5PLACEHOLD_113: KRB5 error code 113
115. KRB5PLACEHOLD_114: KRB5 error code 114
116. KRB5PLACEHOLD_115: KRB5 error code 115
117. KRB5PLACEHOLD_116: KRB5 error code 116

118. KRB5PLACEHOLD_117: KRB5 error code 117
119. KRB5PLACEHOLD_118: KRB5 error code 118
120. KRB5PLACEHOLD_119: KRB5 error code 119
121. KRB5PLACEHOLD_120: KRB5 error code 120
122. KRB5PLACEHOLD_121: KRB5 error code 121
123. KRB5PLACEHOLD_122: KRB5 error code 122
124. KRB5PLACEHOLD_123: KRB5 error code 123
125. KRB5PLACEHOLD_124: KRB5 error code 124
126. KRB5PLACEHOLD_125: KRB5 error code 125
127. KRB5PLACEHOLD_126: KRB5 error code 126
128. KRB5PLACEHOLD_127: KRB5 error code 127
129. KRB5_ERR_RCSID: $Id: admin.texinfo,v 1.7 1996/09/09 18:29:25 jcb Exp $
130. KRB5_LIBOS_BADLOCKFLAG: Invalid flag for file lock mode
131. KRB5_LIBOS_CANTREADPWD: Cannot read password
132. KRB5_LIBOS_BADPWDMATCH: Password mismatch
133. KRB5_LIBOS_PWDINTR: Password read interrupted
134. KRB5_PARSE_ILLCHAR: Illegal character in component name
135. KRB5_PARSE_MALFORMED: Malformed representation of principal
136. KRB5_CONFIG_CANTOPEN: Can't open/find configuration file
137. KRB5_CONFIG_BADFORMAT: Improper format of configuration file
138. KRB5_CONFIG_NOTENUFSPACE: Insufficient space to return complete information
139. KRB5_BADMSGTYPE: Invalid message type specified for encoding
140. KRB5_CC_BADNAME: Credential cache name malformed
141. KRB5_CC_UNKNOWN_TYPE: Unknown credential cache type
142. KRB5_CC_NOTFOUND: Matching credential not found
143. KRB5_CC_END: End of credential cache reached

144. KRB5_NO_TKT_SUPPLIED: Request did not supply a ticket
145. KRB5KRB_AP_WRONG_PRINC: Wrong principal in request
146. KRB5KRB_AP_ERR_TKT_INVALID: Ticket has invalid flag set
147. KRB5_PRINC_NOMATCH: Requested principal and ticket don't match
148. KRB5_KDCREP_MODIFIED: KDC reply did not match expectations
149. KRB5_KDCREP_SKEW: Clock skew too great in KDC reply
150. KRB5_IN_TKT_REALM_MISMATCH: Client/server realm mismatch in initial ticket request
151. KRB5_PROG_ETYPE_NOSUPP: Program lacks support for encryption type
152. KRB5_PROG_KEYTYPE_NOSUPP: Program lacks support for key type
153. KRB5_WRONG_ETYPE: Requested encryption type not used in message
154. KRB5_PROG_SUMTYPE_NOSUPP: Program lacks support for checksum type
155. KRB5_REALM_UNKNOWN: Cannot find KDC for requested realm
156. KRB5_SERVICE_UNKNOWN: Kerberos service unknown
157. KRB5_KDC_UNREACH: Cannot contact any KDC for requested realm
158. KRB5_NO_LOCALNAME: No local name found for principal name
159. KRB5_MUTUAL_FAILED: Mutual authentication failed
160. KRB5_RC_TYPE_EXISTS: Replay cache type is already registered
161. KRB5_RC_MALLOC: No more memory to allocate (in replay cache code)
162. KRB5_RC_TYPE_NOTFOUND: Replay cache type is unknown
163. KRB5_RC_UNKNOWN: Generic unknown RC error
164. KRB5_RC_REPLAY: Message is a replay
165. KRB5_RC_IO: Replay I/O operation failed XXX

166. KRB5_RC_NOIO: Replay cache type does not support non-volatile storage
167. KRB5_RC_PARSE: Replay cache name parse/format error
168. KRB5_RC_IO_EOF: End-of-file on replay cache I/O
169. KRB5_RC_IO_MALLOC: No more memory to allocate (in replay cache I/O code)
170. KRB5_RC_IO_PERM: Permission denied in replay cache code
171. KRB5_RC_IO_IO: I/O error in replay cache i/o code
172. KRB5_RC_IO_UNKNOWN: Generic unknown RC/IO error
173. KRB5_RC_IO_SPACE: Insufficient system space to store replay information
174. KRB5_TRANS_CANTOPEN: Can't open/find realm translation file
175. KRB5_TRANS_BADFORMAT: Improper format of realm translation file
176. KRB5_LNAME_CANTOPEN: Can't open/find lname translation database
177. KRB5_LNAME_NOTRANS: No translation available for requested principal
178. KRB5_LNAME_BADFORMAT: Improper format of translation database entry
179. KRB5_CRYPTO_INTERNAL: Cryptosystem internal error
180. KRB5_KT_BADNAME: Key table name malformed
181. KRB5_KT_UNKNOWN_TYPE: Unknown Key table type
182. KRB5_KT_NOTFOUND: Key table entry not found
183. KRB5_KT_END: End of key table reached
184. KRB5_KT_NOWRITE: Cannot write to specified key table
185. KRB5_KT_IOERR: Error writing to key table
186. KRB5_NO_TKT_IN_RLM: Cannot find ticket for requested realm
187. KRB5DES_BAD_KEYPAR: DES key has bad parity
188. KRB5DES_WEAK_KEY: DES key is a weak key
189. KRB5_BAD_ENCTYPE: Bad encryption type

190. KRB5_BAD_KEYSIZE: Key size is incompatible with encryption type
191. KRB5_BAD_MSIZE: Message size is incompatible with encryption type
192. KRB5_CC_TYPE_EXISTS: Credentials cache type is already registered.
193. KRB5_KT_TYPE_EXISTS: Key table type is already registered.
194. KRB5_CC_IO: Credentials cache I/O operation failed XXX
195. KRB5_FCC_PERM: Credentials cache file permissions incorrect
196. KRB5_FCC_NOFILE: No credentials cache file found
197. KRB5_FCC_INTERNAL: Internal file credentials cache error
198. KRB5_CC_WRITE: Error writing to credentials cache file
199. KRB5_CC_NOMEM: No more memory to allocate (in credentials cache code)
200. KRB5_CC_FORMAT: Bad format in credentials cache
201. KRB5_INVALID_FLAGS: Invalid KDC option combination (library internal error) [for dual tgt library calls]
202. KRB5_NO_2ND_TKT: Request missing second ticket [for dual tgt library calls]
203. KRB5_NOCREDS_SUPPLIED: No credentials supplied to library routine
204. KRB5_SENDAUTH_BADAUTHVERS: Bad sendauth version was sent
205. KRB5_SENDAUTH_BADAPPLVERS: Bad application version was sent (via sendauth)
206. KRB5_SENDAUTH_BADRESPONSE: Bad response (during sendauth exchange)
207. KRB5_SENDAUTH_REJECTED: Server rejected authentication (during sendauth exchange)
208. KRB5_PREAUTH_BAD_TYPE: Unsupported pre-authentication type
209. KRB5_PREAUTH_NO_KEY: Required preauthentication key not supplied

210. KRB5_PREAUTH_FAILED: Generic pre-authentication failure
211. KRB5_RCACHE_BADVNO: Unsupported replay cache format version number
212. KRB5_CCACHE_BADVNO: Unsupported credentials cache format version number
213. KRB5_KEYTAB_BADVNO: Unsupported key table format version number
214. KRB5_PROG_ATYPE_NOSUPP: Program lacks support for address type
215. KRB5_RC_REQUIRED: Message replay detection requires rcache parameter
216. KRB5_ERR_BAD_HOSTNAME: Hostname cannot be canonicalized
217. KRB5_ERR_HOST_REALM_UNKNOWN: Cannot determine realm for host
218. KRB5_SNAME_UNSUPP_NAMETYPE: Conversion to service principal undefined for name type
219. KRB5KRB_AP_ERR_V4_REPLY: Initial Ticket response appears to be Version 4 error
220. KRB5_REALM_CANT_RESOLVE: Cannot resolve KDC for requested realm
221. KRB5_TKT_NOT_FORWARDABLE: Requesting ticket can't get forwardable tickets
222. KRB5_FWD_BAD_PRINCIPAL: Bad principal name while trying to forward credentials
223. KRB5_GET_IN_TKT_LOOP: Looping detected inside krb5_get_in_tkt
224. KRB5_CONFIG_NODEFREALM: Configuration file does not specify default realm

Kerberos V5 Magic Numbers Error Codes

This is the Kerberos v5 magic numbers error code table.

1. KV5M_NONE: Kerberos V5 magic number table
2. KV5M_PRINCIPAL: Bad magic number for krb5_principal structure
3. KV5M_DATA: Bad magic number for krb5_data structure
4. KV5M_KEYBLOCK: Bad magic number for krb5_keyblock structure
5. KV5M_CHECKSUM: Bad magic number for krb5_checksum structure
6. KV5M_ENCRYPT_BLOCK: Bad magic number for krb5_encrypt_block structure
7. KV5M_ENC_DATA: Bad magic number for krb5_enc_data structure
8. KV5M_AUTHDATA: Bad magic number for krb5_authdata structure
9. KV5M_TRANSITED: Bad magic number for krb5_transited structure
10. KV5M_ENC_TKT_PART: Bad magic number for krb5_enc_tkt_part structure
11. KV5M_TICKET: Bad magic number for krb5_ticket structure
12. KV5M_AUTHENTICATOR: Bad magic number for krb5_authenticator structure
13. KV5M_TKT_AUTHENT: Bad magic number for krb5_tkt_authent structure
14. KV5M_CREDS: Bad magic number for krb5_creds structure
15. KV5M_LAST_REQ_ENTRY: Bad magic number for krb5_last_req_entry structure
16. KV5M_PA_DATA: Bad magic number for krb5_pa_data structure
17. KV5M_KDC_REQ: Bad magic number for krb5_kdc_req structure
18. KV5M_ENC_KDC_REP_PART: Bad magic number for krb5_enc_kdc_rep_part structure
19. KV5M_KDC_REP: Bad magic number for krb5_kdc_rep structure

20. KV5M_ERROR: Bad magic number for krb5_error structure
21. KV5M_AP_REQ: Bad magic number for krb5_ap_req structure
22. KV5M_AP_REP: Bad magic number for krb5_ap_rep structure
23. KV5M_AP_REP_ENC_PART: Bad magic number for krb5_ap_rep_enc_part structure
24. KV5M_RESPONSE: Bad magic number for krb5_response structure
25. KV5M_SAFE: Bad magic number for krb5_safe structure
26. KV5M_PRIV: Bad magic number for krb5_priv structure
27. KV5M_PRIV_ENC_PART: Bad magic number for krb5_priv_enc_part structure
28. KV5M_CRED: Bad magic number for krb5_cred structure
29. KV5M_CRED_INFO: Bad magic number for krb5_cred_info structure
30. KV5M_CRED_ENC_PART: Bad magic number for krb5_cred_enc_part structure
31. KV5M_PWD_DATA: Bad magic number for krb5_pwd_data structure
32. KV5M_ADDRESS: Bad magic number for krb5_address structure
33. KV5M_KEYTAB_ENTRY: Bad magic number for krb5_keytab_entry structure
34. KV5M_CONTEXT: Bad magic number for krb5_context structure
35. KV5M_OS_CONTEXT: Bad magic number for krb5_os_context structure
36. KV5M_ALT_METHOD: Bad magic number for krb5_alt_method structure
37. KV5M_ETYPE_INFO_ENTRY: Bad magic number for krb5_etype_info_entry structure
38. KV5M_DB_CONTEXT: Bad magic number for krb5_db_context structure
39. KV5M_AUTH_CONTEXT: Bad magic number for krb5_auth_context structure
40. KV5M_KEYTAB: Bad magic number for krb5_keytab structure

41. KV5M_RCACHE: Bad magic number for krb5_rcache structure
42. KV5M_CCACHE: Bad magic number for krb5_ccache structure
43. KV5M_PREAUTH_OPS: Bad magic number for krb5_preauth_ops
44. KV5M_PASSWD_PHRASE_ELEMENT: Bad magic number for passwd_phrase_element

ANSI.1 Error Codes

1. ASN1_BAD_TIMEFORMAT: ASN.1 failed call to system time library
2. ASN1_MISSING_FIELD: ASN.1 structure is missing a required field
3. ASN1_MISPLACED_FIELD: ASN.1 unexpected field number
4. ASN1_TYPE_MISMATCH: ASN.1 type numbers are inconsistent
5. ASN1_OVERFLOW: ASN.1 value too large
6. ASN1_OVERRUN: ASN.1 encoding ended unexpectedly
7. ASN1_BAD_ID: ASN.1 identifier doesn't match expected value
8. ASN1_BAD_LENGTH: ASN.1 length doesn't match expected value
9. ASN1_BAD_FORMAT: ASN.1 badly-formatted encoding
10. ASN1_PARSE_ERROR: ASN.1 parse error

GSSAPI Error Codes

Generic GSSAPI Errors:

1. GSS_KRB5_S_G_BAD_SERVICE_NAME: /* "No @ in SERVICE-NAME name string" */
2. GSS_KRB5_S_G_BAD_STRING_UID: /* "STRING-UID-NAME contains nondigits" */
3. GSS_KRB5_S_G_NOUSER: /* "UID does not resolve to username" */
4. GSS_KRB5_S_G_VALIDATE_FAILED: /* "Validation error" */
5. GSS_KRB5_S_G_BUFFER_ALLOC: /* "Couldn't allocate gss_buffer_t data" */
6. GSS_KRB5_S_G_BAD_MSG_CTX: /* "Message context invalid" */
7. GSS_KRB5_S_G_WRONG_SIZE: /* "Buffer is the wrong size" */
8. GSS_KRB5_S_G_BAD_USAGE: /* "Credential usage type is unknown" */
9. GSS_KRB5_S_G_UNKNOWN_QOP: /* "Unknown quality of protection specified" */

Kerberos 5 GSSAPI Errors:

1. GSS_KRB5_S_KG_CCACHE_NOMATCH: /* "Principal in credential cache does not match desired name" */
2. GSS_KRB5_S_KG_KEYTAB_NOMATCH: /* "No principal in keytab matches desired name" */
3. GSS_KRB5_S_KG_TGT_MISSING: /* "Credential cache has no TGT" */
4. GSS_KRB5_S_KG_NO_SUBKEY: /* "Authenticator has no subkey" */
5. GSS_KRB5_S_KG_CONTEXT_ESTABLISHED: /* "Context is already fully established" */
6. GSS_KRB5_S_KG_BAD_SIGN_TYPE: /* "Unknown signature type in token" */

7. GSS_KRB5_S_KG_BAD_LENGTH: /* "Invalid field length in token" */
8. GSS_KRB5_S_KG_CTX_INCOMPLETE: /* "Attempt to use incomplete security context" */

FATAL ERROR CODES

1. GSS_S_BAD_BINDINGS: channel binding mismatch
2. GSS_S_BAD_MECH: unsupported mechanism requested
3. GSS_S_BAD_NAME: invalid name provided
4. GSS_S_BAD_NAMETYPE: name of unsupported type provided
5. GSS_S_BAD_STATUS: invalid input status selector
6. GSS_S_BAD_SIG: token had invalid integrity check
7. GSS_S_BAD_MIC: preferred alias for GSS_S_BAD_SIG
8. GSS_S_CONTEXT_EXPIRED: specified security context expired
9. GSS_S_CREDENTIALS_EXPIRED: expired credentials detected
10. GSS_S_DEFECTIVE_CREDENTIAL: defective credential detected
11. GSS_S_DEFECTIVE_TOKEN: defective token detected
12. GSS_S_FAILURE: failure, unspecified at GSS-API level
13. GSS_S_NO_CONTEXT: no valid security context specified
14. GSS_S_NO_CRED: no valid credentials provided
15. GSS_S_BAD_QOP: unsupported QOP value
16. GSS_S_UNAUTHORIZED: operation unauthorized
17. GSS_S_UNAVAILABLE: operation unavailable
18. GSS_S_DUPLICATE_ELEMENT: duplicate credential element requested
19. GSS_S_NAME_NOT_MN: name contains multi-mechanism elements

INFORMATORY STATUS CODES

1. GSS_S_COMPLETE: normal completion
2. GSS_S_CONTINUE_NEEDED: continuation call to routine required

3. GSS_S_DUPLICATE_TOKEN: duplicate per-message token detected
4. GSS_S_OLD_TOKEN: timed-out per-message token detected
5. GSS_S_UNSEQ_TOKEN: reordered (early) per-message token detected
6. GSS_S_GAP_TOKEN: skipped predecessor token(s) detected

F Kerberos Client Environment Variables

This appendix lists and describes the various Kerberos environment variables that you may need to set while using Kerberos Client.

Kerberos Client Environment Variables

Following lists and describes the Kerberos Client environment variables:

KRB5RCACHEDIR The default replay cache directory. The placement of the replay cache file can be changed by setting the KRB5RCACHEDIR or KRB5RCACHENAME environment variable.

KRB5RCACHENAME The default replay cache name. The default is /var/tmp/rc_host_(*uid*) where <*uid*> is the user id of the process.

GSSAPI_MECH_CONF The default path of the mechanism file (/etc/gss/mech). This default path can be changed by setting the GSSAPI_MECH_CONF environment variable.

KRB5_CONFIG The default configuration file (/etc/krb5.conf). This default file can be changed by setting the KRB5_CONFIG environment variable. You can specify one or more configuration file names separated by colons.

KRB5CCNAME The default name for the credentials cache file. You can set the variable type to the following value:

[[<*cc type*>:] <*file name*>]

where:

- <*cc type*> can be FILE or MEMORY
- <*file name*> is the location of the principal's credential cache

If the FILE type is specified, subsequent operations on the associated file are readable and writable by the invoking process.

If the MEMORY type is specified, a temporary cache is created for the life of the invoking process.

If KRB5CCNAME is not defined, the default is to perform FILE-based credentials caching in /tmp/krb5cc_(uid) where <*uid*> is the user id of the process that created the cache file.

KRB5_KTNAME — The default key table name. You can set the variable type to the following value:

[[<*kt type*>:]<*file name*>]

where:

- <*kt type*> can be FILE or WRFILE
- <*file name*> is the location of the keytab file

Use the FILE type for read operations, and the WRFILE type for write operations.

If KRB5_KTNAME is not specified, the file specified by the default_keytab_name configuration entry in the configuration file is used. If the configuration entry is not specified, the default file is /etc/krb5.keytab.

Kerberos Client Environment Variables