Novell Kerberos* KDC

1.5

www.novell.com

ADMINISTRATION GUIDE

April 8, 2008

Novell®

Legal Notices

Novell, Inc., makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc., makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

Any products or technical information provided under this Agreement may be subject to U.S. export controls and the trade laws of other countries. You agree to comply with all export control regulations and to obtain any required licenses or classification to export, re-export or import deliverables. You agree not to export or re-export to entities on the current U.S. export exclusion lists or to any embargoed or terrorist countries as specified in the U.S. export laws. You agree to not use deliverables for prohibited nuclear, missile, or chemical biological weaponry end uses. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software. Novell assumes no responsibility for your failure to obtain any necessary export approvals.

Novell, Inc., has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the Novell Legal Patents Web page (http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

For Novell trademarks, see the Novell Trademark and Service Mark list (http://www.novell.com/company/legal/trademarks/tmlist.html)

All third-party trademarks are the property of their respective owners.

"Commercial use" means use of a name in a product or other for-profit manner. It does NOT prevent a commercial firm from referring to the MIT trademarks in order to convey information (although in doing so, recognition of their trademark status should be given).

Portions of src/lib/crypto have the following copyright:

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. FundsXpress makes no representations about the suitability of this software for any purpose. It is provided "as is" without express or implied warranty.

THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The following copyright and permission notice applies to the OpenVision Kerberos Administration system located in kadmin/create, kadmin/dbutil, kadmin/passwd, kadmin/server, lib/kadm5, and portions of lib/rpc:

Warning: Retrieving the OpenVision Kerberos Administration system source code, as described below, indicates your acceptance of the following terms. If you do not agree to the following terms, do not retrieve the OpenVision Kerberos administration system.

You may freely use and distribute the Source Code and Object Code compiled from it, with or without modification, but this Source Code is provided to you "AS IS" EXCLUSIVE OF ANY WARRANTY, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY OTHER WARRANTY, WHETHER EXPRESS OR IMPLIED. IN NO EVENT WILL OPENVISION HAVE ANY LIABILITY FOR ANY LOST PROFITS, LOSS OF DATA OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM THE USE OF THE SOURCE CODE, OR THE FAILURE OF THE SOURCE CODE TO PERFORM, OR FOR ANY OTHER REASON.

OpenVision retains all copyrights in the donated Source Code. OpenVision also retains copyright to derivative works of the Source Code, whether created by OpenVision or by a third party. The OpenVision copyright notice must be preserved if derivative works are made based on the donated Source Code.

OpenVision Technologies, Inc. has donated this Kerberos Administration system to MIT for inclusion in the standard Kerberos 5 distribution. This donation underscores our commitment to continuing Kerberos technology development and our gratitude for the valuable work which has been performed by MIT and the Kerberos community.

Portions contributed by Matt Crawford <crawdad@fnal.gov> were work performed at Fermi National Accelerator Laboratory, which is operated by Universities Research Association, Inc., under contract DE-AC02-76CHO3000 with the U.S. Department of Energy.

The implementation of the Yarrow pseudo-random number generator in src/lib/crypto/yarrow has the following copyright:

WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The implementations of GSSAPI mechglue in GSSAPI-SPNEGO in `src/lib/gssapi`, including the following files:

lib/gssapi/generic/gssapi_err_generic.et
lib/gssapi/mechglue/g_accept_sec_context.c
lib/gssapi/mechglue/g_acquire_cred.c
lib/gssapi/mechglue/g_canon_name.c
lib/gssapi/mechglue/g_compare_name.c
lib/gssapi/mechglue/g_context_time.c
lib/gssapi/mechglue/g_delete_sec_context.c
lib/gssapi/mechglue/g_dsp_name.c
lib/gssapi/mechglue/g_dsp_status.c
lib/gssapi/mechglue/g_dup_name.c
lib/gssapi/mechglue/g_exp_sec_context.c
lib/gssapi/mechglue/g_export_name.c
lib/gssapi/mechglue/g_glue.c
lib/gssapi/mechglue/g_imp_name.c
lib/gssapi/mechglue/g_imp_sec_context.c
lib/gssapi/mechglue/g_init_sec_context.c
lib/gssapi/mechglue/g_initialize.c
lib/gssapi/mechglue/g_inquire_context.c
lib/gssapi/mechglue/g_inquire_cred.c
lib/gssapi/mechglue/g_inquire_names.c
lib/gssapi/mechglue/g_process_context.c
lib/gssapi/mechglue/g_rel_buffer.c
lib/gssapi/mechglue/g_rel_cred.c
lib/gssapi/mechglue/g_rel_name.c
lib/gssapi/mechglue/g_rel_oid_set.c
lib/gssapi/mechglue/g_seal.c
lib/gssapi/mechglue/g_sign.c
lib/gssapi/mechglue/g_store_cred.c
lib/gssapi/mechglue/g_unseal.c
lib/gssapi/mechglue/g_userok.c
lib/gssapi/mechglue/g_utils.c
lib/gssapi/mechglue/g_verify.c
lib/gssapi/mechglue/gssd_pname_to_uid.c
lib/gssapi/mechglue/mglueP.h
lib/gssapi/mechglue/oid_ops.c
lib/gssapi/spnego/gssapiP_spnego.h
lib/gssapi/spnego/spnego_mech.c

are subject to the following license:

Contents

About This Guide

This guide describes Novell® Kerberos KDC and provides information on how to administer it. It is divided into the following sections:

Audience

The guide is intended for Novell eDirectory™ or Kerberos administrators.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. Please use the User Comment feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Additional Documentation

- Novell eDirectory 8.8 documentation (http://www.novell.com/documentation/edir88/index.html)
- Kerberos documentation (http://web.mit.edu/kerberos/www/)

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux* or UNIX*, should use forward slashes as required by your software.

Overview

1

This section introduces Kerberos and Novell® Kerberos KDC.

- Section 1.1, “Overview of Kerberos,” on page 13
- Section 1.2, “Understanding the Novell Kerberos KDC,” on page 14

1.1 Overview of Kerberos

Kerberos is a standard protocol that provides a means of authenticating entities on a network and is based on a trusted third-party model. It involves shared secrets and uses symmetric key cryptography. Kerberos was developed at the Massachusetts Institute of Technology (MIT).

MIT created Kerberos as a solution to network security problems. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server have used Kerberos to prove their identity, they can also encrypt all of their communication to assure privacy and data integrity.

Kerberos is a solution to your network security problems. Kerberos provides authentication over the network by using cryptography, and secures information systems across the entire enterprise.

This section introduces you to Kerberos and its concepts:

- Section 1.1.1, “Kerberos Terminology,” on page 13
- Section 1.1.2, “How Does Kerberos Work,” on page 14

1.1.1 Kerberos Terminology

The following table lists the definitions of some commonly used Kerberos terminologies.

Table 1-1 *Kerberos Terminologies*

Terminology	Definition
Key (also referred to as Secret Key)	Encryption key shared by a principal and the KDC, distributed outside the system, with a long lifetime. In the case of a user’s principal, the key is derived from a password.
Principal	Entity in the network. Each entity corresponds to a principal.
Realm	Logical grouping of principals.
Service	Resource provided to network clients, such as mail server.
Session key	Temporary encryption key used between two principals, with a lifetime limited to the duration of a single login “session”.
Service ticket	Required to access services in the network.
Ticket	Record that helps a client authenticate itself to a server. It contains information such as client’s identity, a session key, a time stamp, and other information—all sealed using the server’s secret key.

Terminology	Definition
Ticket Granting Ticket (TGT)	Initial ticket obtained after a successful login. This ticket is used to get the service ticket to access a service.

1.1.2 How Does Kerberos Work

Kerberos uses the concept of a central server called the Key Distribution Center (KDC). The KDC contains the identities and keys of every principal in the network that must service within its realm. This principal information is stored in a local database within the KDC. In Novell Kerberos KDC, the principal and realm information is stored in Novell eDirectory™

A typical KDC provides the following basic services:

- **Authentication Server (AS):** Issues authentication credentials known as Ticket Granting Tickets (TGT) to users while logging in.
- **Ticket Granting Server (TGS):** Issues service tickets to the users in response to their requests accompanied by TGT so that they can access various services in the realm.

Kerberos provides the following additional services and utilities to manage KDC and Kerberos principals:

- **Kerberos Administration Server:** Server component for maintaining Kerberos principals, policies, and service key tables (keytabs). This server responds to the requests from the kadmin utility.
- **Kerberos Administration Utilities:** Client component (such as, kadmin, kadmin.local, and kdb5_ldap_util) for maintaining Kerberos realms, principals, policies, and service key tables.
- **Kerberos Password Server:** Server component of the Kerberos Password utility for changing passwords of Kerberos principals.
- **Kerberos Client Utilities:** Utilities such as kinit and kpasswd, which are used for various operations like login and changing passwords.

For more information on the Kerberos solution developed by the MIT, refer to the *Kerberos System Administrator's Guide* (http://web.mit.edu/kerberos/www/).

1.2 Understanding the Novell Kerberos KDC

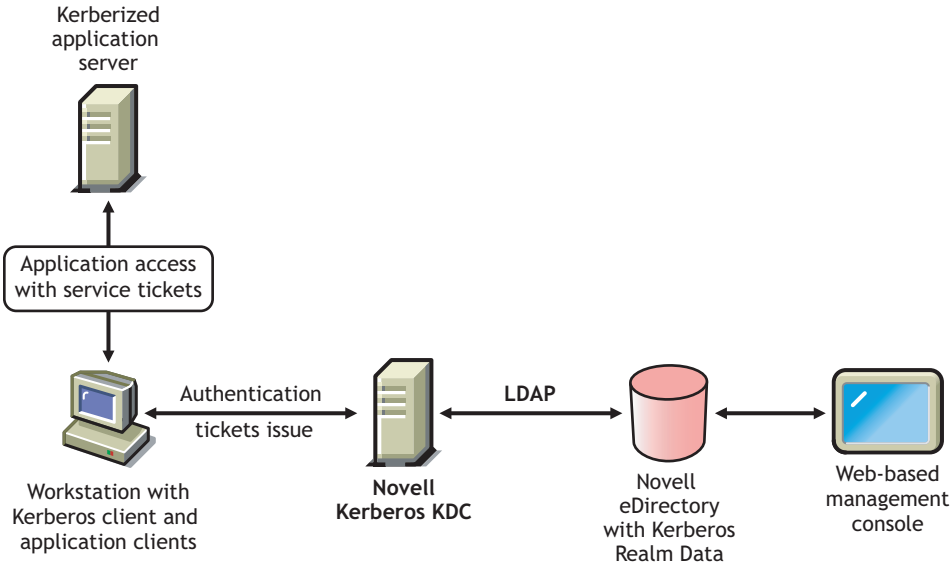
Traditional Kerberos implementations store relevant Kerberos information pertaining to a realm in a database. Database propagation between KDCs are handled by vendor-specific protocols. The Kerberos database is managed using vendor-specific administration utilities.

Novell® Kerberos KDC provides the ease of single point of management for deployments with both Kerberos and Novell eDirectory™, and gives the advantage of eDirectory replication and security capabilities. It moves Kerberos-specific data to eDirectory and provides Kerberos services using a KDC that accesses data stored in eDirectory. Additionally, because authentication requests lead to database operations that are mostly read-only in nature, eDirectory is well suited to replace the traditional database component.

Novell Kerberos KDC integrates Kerberos Authentication, Administration, and Password Servers with eDirectory as data store. Administration is possible both using the traditional command line tools and Novell's Web-based framework, iManager.

The Novell Kerberos KDC is derived from the MIT implementation of Kerberos (http://web.mit.edu/kerberos). It is interoperable with the Kerberos implementations from other vendors like Microsoft* Active Directory*.

Figure 1-1 *Kerberos Authentication Using Novell Kerberos KDC*



This section provides information about the following:

- Section 1.2.1, “Novell Kerberos KDC Features,” on page 15
- Section 1.2.2, “Novell Kerberos KDC Components,” on page 15
- Section 1.2.3, “Changes to the MIT KDC Code Base,” on page 16

1.2.1 Novell Kerberos KDC Features

Novell Kerberos KDC provides the following features:

- A standard authentication method to leverage your existing eDirectory deployment.
- An iManager interface to manage multiple Kerberos realms.
- Universal Password integration that enables you to use the same password to log in to both eDirectory and KDC.
- The login restrictions of the users can be enforced for Kerberos authentications.

NOTE: Kerberos 4 is not supported.

1.2.2 Novell Kerberos KDC Components

This section introduces you to the components of Novell Kerberos KDC.

- “Key Distribution Center (KDC)” on page 16
- “Kerberos Administration Server” on page 16
- “Kerberos Password Server” on page 16

- "kdb5_ldap_util and kadmin" on page 16
- "Kerberos LDAP Extensions" on page 16
- "Kerberos Password Agent" on page 16

Key Distribution Center (KDC)

KDC provides authentication and ticket granting services to Kerberos clients. The principal and realm information is stored in eDirectory. Novell Kerberos KDC accesses this information by using secure LDAP connections.

Kerberos Administration Server

The Administration server services administrative requests such as principal management and key tab operations. This server acts like any another kerberized service on the network and requires the corresponding service ticket to perform any operation.

Kerberos Password Server

The Password server provides the necessary functionality to change principals' passwords from standard Kerberos Change Password clients. Users who want to use this service to change their passwords need to authenticate to KDC first and get the service ticket for this Password server. Although the wire-level protocol for this change password is still not a standard, this server complies with the Internet Draft on the Kerberos Change Password Protocol.

kdb5_ldap_util and kadmin

kdb5_ldap_util and kadmin are tools for managing the Kerberos realm and principals in eDirectory. For more information on these utilities refer to Chapter 3, "Managing the Novell Kerberos KDC," on page 27.

Kerberos LDAP Extensions

Kerberos LDAP extensions service the requests for storing and retrieving various Kerberos-specific keys from eDirectory, for example, the master key of a Realm. The keys are stored in eDirectory in a secure form.

Kerberos Password Agent

Kerberos Password Agent keeps the Kerberos password in sync with the Universal Password. Therefore, it needs to be deployed when Universal Password integration is required. It synchronizes the Kerberos password with Universal Password whenever the Universal Password is set in eDirectory.

1.2.3 Changes to the MIT KDC Code Base

- Tight integration of Kerberos and eDirectory identities, including a single password by means of Universal Password.
- Separate Password server instead of the Administration server playing that role.
- New kdb5_ldap_util utility to configure the Novell KDC.
- Modifications to kdb5_ldap_util to work with eDirectory.

- Additions to the `krb5.conf` configuration file to include eDirectory configuration.
- Kerberos 4 is not supported.

Kerberos KDC Integration with eDirectory

2

Novell® Kerberos KDC, provides the ease of single point of management for deployments with both Kerberos and Novell eDirectory™, and gives the advantage of eDirectory replication and security capabilities. It moves Kerberos-specific data to eDirectory and provides Kerberos services by using a KDC that accesses data stored in eDirectory.

In a Kerberos system, the entities in a network are called principals and a logical grouping of principals is called a realm.

In Novell Kerberos KDC, the realms and principals of Kerberos are mapped to eDirectory as shown in the following table:

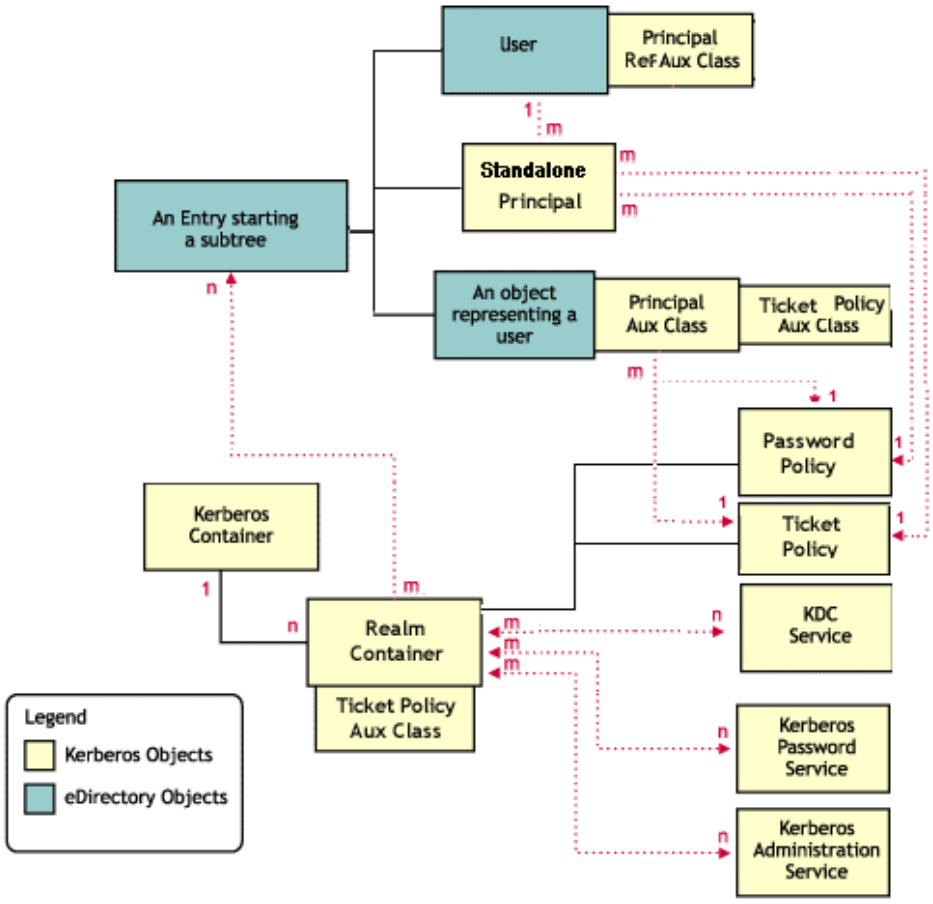
Table 2-1 *Kerberos Mapping With eDirectory*

Kerberos Term	Mapping to eDirectory
Realm	Can be mapped to one or more subtrees or containers For example, if eDirectory has an HR container, you can create an HRREALM realm that references to the HR container. The principals of HRREALM can be located under this container.
Principal	Can be mapped to an existing directory object or created as separate object. For example, if an eDirectory tree has FTP as a service object and John as an user object, you can add the following principals: ◆ A user principal, John. ◆ A service principal, FTP. A Kerberos- specific object class, krbPrincipalAux, is added to the objects.

You can create realms in eDirectory and add principals to these realms. You can associate these realms and principals to eDirectory containers and users or service objects. For information on creating realms, adding principals, and managing them, refer to Chapter 3, “Managing the Novell Kerberos KDC,” on page 27.

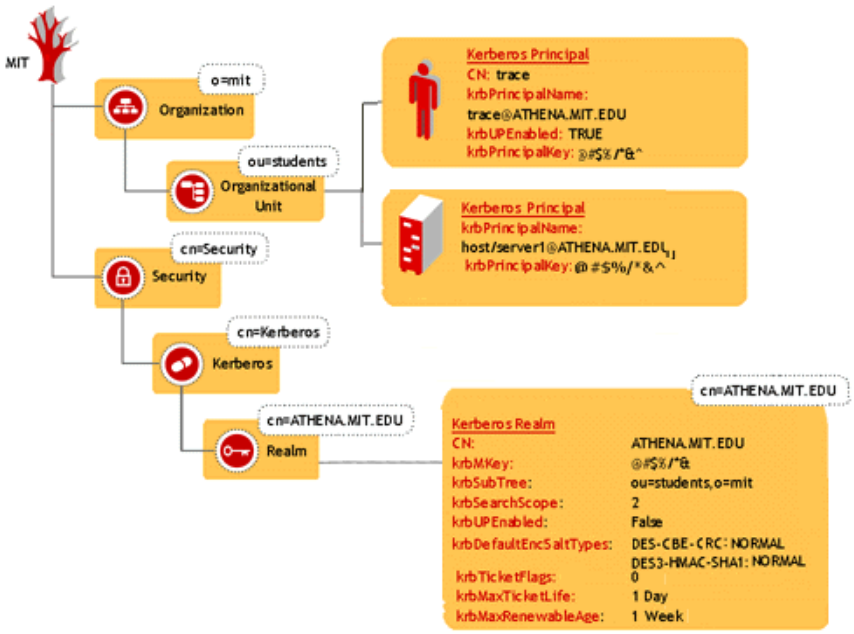
You need to create the realms under the Kerberos container, which can be located anywhere in the eDirectory tree. This helps you easily administer the Kerberos objects.

Figure 2-1 *Kerberos Integration with eDirectory*



The following diagram illustrates how the Kerberos data is mapped in eDirectory:

Figure 2-2 *eDirectory and Kerberos Mapping*



This section provides information on:

- Section 2.1, “Kerberos Container,” on page 21
- Section 2.2, “Realm Container,” on page 22
- Section 2.3, “Principal Objects,” on page 23
- Section 2.4, “Kerberos Service Objects,” on page 24
- Section 2.5, “Ticket Policy Object,” on page 24
- Section 2.6, “Password Policy Object,” on page 25
- Section 2.7, “Administrative Considerations for Integrating Kerberos with eDirectory,” on page 25

2.1 Kerberos Container

The Kerberos container contains only the realm objects. All the realm objects in the tree are placed in this container. This makes Kerberos administration easier. The Kerberos container can be located anywhere in the tree, but the location of the container is stored in the security container.

The Kerberos container can be

- In the security container under the root (default);
- Under any sub-tree or container

IMPORTANT: Make sure that the Kerberos container is replicated on all eDirectory servers that Kerberos services (KDC, Administration, and Password services) are configured with.

The following table describes the Kerberos container attribute:

Table 2-2 *Kerberos Container Attribute*

Attribute	Description
Container name	Name of the container object.

The security container contains an attribute that gives the location of the Kerberos container.

Table 2-3 *Security Container Attribute*

Attribute	Description
Container reference	DN of the Kerberos container.

2.2 Realm Container

The realm container stores the realm name and related realm information for Kerberos authentication, administration, and password management servers to process requests. This object contains the ticket policy, password policy, and principal objects, and internal principals such as krbtgt, kadmin/admin, kadmin/changepw, and kadmin/history.

2.2.1 Realm Container Attributes

The following table describes the realm container attributes:

Table 2-4 *Realm Container Attributes*

Attribute	Description
Realm name	Name of the realm. This is unique within an eDirectory tree.
Default encryption salt types	The default encryption salt types supported by the realm.
Master key	Realm-specific master key.
Search scope	Scope for searching the principals under the specified subtree.
Universal Password enabled	Specifies whether to use the Universal Password of the user as the Kerberos password.
Login policy enabled	Specifies whether the login restrictions of the user must be enforced.

2.2.2 Realm Container Associations

The following table describes the objects that you can associate the realm container to:

Table 2-5 Realm Container Associations

Associate To	Description
Subtrees	Reference to container objects under which the principals of the realm are placed.
Principal container reference	Reference to the container under which the standalone principals are created.
KDC servers	List of references to the KDC service objects that can service the realm.
Administration servers	List of references to the administration service objects that can service the realm.
Password servers	List of references to the password service objects that can service the realm.

2.3 Principal Objects

A principal is a fundamental entity in Kerberos. All the services, clients, and users are represented as principals in Kerberos. Principals are associated with keys.

2.3.1 Principal Attributes

The following table describes the principal attributes:

Table 2-6 Principal Attributes

Attribute	Description
Principal name	Name of the principal. This is used to uniquely identify a principal within a realm.
Principal expiration	The time when the principal expires.
Password expiration	The time when the principal's password expires.
Principal (secret) key	A set of all the secret keys that are associated with a principal. The version, type, and other information about the keys are stored in this attribute.
Universal Password enabled	Specifies whether to use the Universal Password of the user as the Kerberos password.
Last Password change	The time when the principal password was last changed.

2.3.2 Principal Associations

The following table describes the object you can associate a principal to:

Table 2-7 Principal Associations

Associate to	Description
Ticket policy	Reference to a ticket policy object that is applicable to a particular principal.
Password policy	Reference to a Kerberos password policy object that is applicable to a particular principal.

2.4 Kerberos Service Objects

Each service of Novell Kerberos KDC (KDC server, Administration server, and Password server) uses a representative object in eDirectory. This has two purposes:

- To treat the service as a client of eDirectory and provide necessary authorization
- To store any configuration related to the service

When each service comes up, it makes an LDAP bind to eDirectory as the corresponding service object, using the stashed password or stored certificate on the local system. All subsequent operations happen based on the rights provided to that object.

2.4.1 Kerberos Service Attributes

The following table describes the Kerberos service attributes:

Table 2-8 Kerberos Service Attributes

Attribute	Description
Service Name	Name of the Kerberos server.
Host server	This attribute holds the host name, transport protocol and port for a Kerberos service.

2.4.2 Kerberos Service Associations

The following table describes the object you can associate a Kerberos service to:

Table 2-9 Kerberos Service Associations

Associate To	Description
Realm references	List of references to the realm objects.

2.5 Ticket Policy Object

This is a Kerberos ticket policy object. This object is located under the realm container.

2.5.1 Ticket Policy Attributes

The following table describes the policy attributes:

Table 2-10 *Policy Attributes*

Attribute	Description
Ticket policy name	Name of the ticket policy.
Ticket flags	Various ticket flags that can be allowed for a principal.
Maximum ticket lifetime	Maximum lifetime of a ticket for a principal in seconds.
Maximum ticket renewable lifetime	Maximum renewable lifetime for a principal's ticket in seconds.

2.6 Password Policy Object

This is a Kerberos password policy object. This object is located under the realm container.

2.6.1 Password Policy Attributes

The following table describes the password policy attributes:

Table 2-11 *Password Policy Attributes*

Attribute	Description
Password Policy Name	Name of the Password Policy.
Maximum password life	Maximum lifetime of a principal's password.
Minimum password life	Minimum lifetime of a principal's password.
Password minimum characters	Minimum number of character classes allowed in a password.
Password minimum length	Minimum length of the password.
Password history length	Number of previous versions of passwords that are stored.

2.7 Administrative Considerations for Integrating Kerberos with eDirectory

Administrators are advised to consider the following points while deploying Novell Kerberos KDC, because it involves an integration between Kerberos and eDirectory paradigms:

- Multiple realms can be configured in a single tree.
- Only one Kerberos identity can be stored on the user object. To associate multiple Kerberos identities, a Kerberos principal can be created separately and linked to the user object.

- Overall, the need for more than one administrator is avoided. The benefits of having such a single point of management are ease of administration and reduced administrative costs.
- If separate eDirectory container administrators are present, each has an additional responsibility of administrating the Kerberos data.

Managing the Novell Kerberos KDC

3

This section provides the following information on managing the Novell® Kerberos KDC:

3.1 The krb5.conf Configuration File

The `krb5.conf` file contains Kerberos configuration information, including the locations of KDCs and admin servers for the Kerberos realms of interest, defaults for the current realm and for Kerberos applications, and mappings of hostnames onto Kerberos realms. You should install your `krb5.conf` file in the /etc directory. You can override the default location by setting the environment variable KRB5_CONFIG. While managing Novell Kerberos KDC, when you do not specify any of the mandatory parameters, the values are taken from the `/etc/krb5.conf` file.

For a sample configuration file, refer to Appendix A, “Sample krb5.conf File,” on page 91.

Table 3-1 *krb5.conf Configuration File Details*

Parameter	Description
libdefaults	
default_realm	Identifies the default Kerberos realm.
realms	
max_life	Specifies the maximum lifetime of the ticket issued.
max_renewable_life	Specifies the maximum time period during which a valid ticket can be renewed.
acl_file	Filename and path of the ACL file.
dict_file	Filename and path of the DICT file.
kdc	KDC server name for the realm.
admin_server	Administration server name for the realm.

Parameter	Description
kpasswd_server	Password server name for the realm.
database_module	Database module configuration tag (refer to the one used in the `dbmodules` section.)
domain_realm	Domain-realm mappings provides translation from a domain name or hostname to a Kerberos realm.
logging	
kdc	Filename and path of the KDC log file.
admin_server	Filename and path of the Administration server log file.
kpasswd_server	Filename and path of the Password server log file.
dbdefaults	
database_module	Database module configuration tag (refer to the one used in `dbmodules` section.)
dbmodules	
db_module_dir	Directory in which the LDAP plug-in module (kldap) is present.
db_library	The library name should be set to kldap.
ldap_kdc_dn	KDC service object DN.
ldap_kadmind_dn	Administration service object DN.
ldap_kpasswdd_dn	Password service object DN.
ldap_root_certificate_file	Path of the trusted root certificate file.
ldap_service_password_file	Path of the service password stash file.
ldap_servers	List of LDAP servers.
ldap_conns_per_server	Number of LDAP connections to be used by KDC, Administration server, or Password server. This parameter value must be set to 2. In this release, the Kerberos servers require two connections and do not use more than two connections at a time.

3.2 Configuration and Administration Utilities

Use the kdb5_ldap_util utility to manage realms, Kerberos services, and ticket policies.

Use the kadmin utility to manage principals, password policies, and keytab entries.

You can also use iManager to configure and administer the Novell Kerberos KDC.

3.2.1 The kdb5_ldap_util Utility

This utility has the following syntax:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri][-t trusted_cert]
cmd [cmd_options]
```

The kdb5_ldap_util parameters are described below:

Table 3-2 *kdb5_ldap_util Parameters*

Parameter	Description
-D	Distinguished name of the user who has sufficient rights to authenticate to the LDAP server and configure Kerberos services.
-w	Userdn password. We do not recommend that you use this option because the password is visible when you enter it through command line.
-H	URI of the LDAP server.
-t	Filename that contains the trusted root certificate of the LDAP server.

The command options include the following:

Table 3-3 *kdb5_ldap_util Command Options*

Command	Description
create	Creates a realm.
modify	Modifies a realm.
view	Displays the attributes of a realm.
destroy	Destroys a realm.
list	Lists all the realms.
create_service	Creates a KDC, Administration, or Password service.
modify_service	Modifies a KDC, Administration, or Password service.
view_service	Displays the service details.
destroy_service	Deletes the service.
list_service	Lists all the services.
create_policy	Creates a ticket policy.
modify_policy	Modifies a ticket policy.
view_policy	Displays the ticket policy details.
destroy_policy	Deletes a ticket policy.
list_policy	Lists all the ticket policies.
setsrvpw	Sets a password for the service objects such as KDC, Administration, and Password server in the stash file and eDirectory™.
setsrvcert	Configures the service to use the issued certificate instead of the password for authentication to the LDAP server.
ldapxtn_info	Updates the ldapExtensionInfo attribute on the LDAP server object.
setmasterkey	Sets the master key password.

3.2.2 The kadmin Utility

You can use the kadmin or kadmin.local utilities to manage principals, keys, and password policies. In the Novell Kerberos KDC, kadmin.local is used to access the database (eDirectory) remotely, unlike MIT Kerberos.

kadmin is a client utility and contacts the Administration server, which in turn contacts eDirectory for any administration request.

kadmin.local directly contacts eDirectory for completing the administration request.

The syntax for using this utility is as follows:

```
kadmin [-r realm] [-p principal] [-q query] [-s admin_server[:port]]
[-w password] [[-c ccache]|[-k [-t keytab]]]
kadmin.local [-r realm] [-p principal] [-q query] [-x db_args]  [-e
"enc:salt ..."] [-m]
cmd [cmd_options]
```

The kadmin and kadmin.local parameters are described below:

Table 3-4 *kadmin and kadmin.local Parameters*

Parameter	Description
-r	Kerberos realm of the database. By default, the default_realm parameter of the `krb5.conf` file is used.
-p	Principal to authenticate to the administration server.
-q	Passes the query directly to kadmin, which performs the query and then exits.
-s	The admin server that kadmin should contact.
-c	Indicates to use credentials_cache as the credentials cache. The credentials_cache should contain a service ticket for the kadmin/admin service; it can be acquired with the kinit(1) program. If this option is not specified, kadmin requests a new service ticket from the KDC, and stores it in its own temporary ccache.
-k	Uses a keytab to decrypt the KDC response instead of prompting for a password on the keyboard. In this case, the default principal is host/hostname. If there is not a keytab specified with the t option, then the default keytab is used.
-t	Uses a keytab to decrypt the KDC response. This can only be used with the -k option.

Parameter	Description
-x	Database-specific parameters. • -x host=<hostname> Specifies the LDAP server to connect to by a LDAP URI. The same as the ldap_servers parameter in the configuration file. • -x binddn=<bind_dn> DN of the object used by the administration server to bind to the LDAP server. The object should have the read and write rights on the realm container, subtrees, and principal container configured for the realm. The binddn equates to ldap_kadmin_dn in the configuration file. • -x bindpwd=<bind_password> Password for the binddn. You are recommended not to use this option. Instead, you can securely store the password in a file by using the `setsrvpw` command of kdb5_ldap_util. This option overrides the password that is read from the ldap_service_password_file. • -x cert=<certificate_file> The trusted root certificate file for the LDAP server. The same as the ldap_root_certificate_file parameter from the configuration file.
-e	Sets the list of encryption types and salt types to be used for any new keys created.
-m	Do not authenticate using a keytab. This option causes kadmin to prompt for the master database password.
-w	Uses the password specified and does not prompt for it. **NOTE:** Placing the password for a Kerberos principal with administration access into a shell script can be dangerous if unauthorized users get read access to the script.

The command options include the following:

Table 3-5 *kadmin and kadmin.local Command Options*

Command Option	Description
add_principal, addprinc, ank	Adds a principal.
delete_principal, delprinc	Deletes a principal.
modify_principal, modprinc	Modifies a principal.
change_password, cpw	Sets the principal password.
get_principal, getprinc	Displays the attributes of a principal.
list_principals, listprincs, get_principals, getprincs	Lists all the principals.
add_policy, addpol	Adds a password policy.
modify_policy, modpol	Modifies a password policy.
delete_policy, delpol	Deletes a password policy.
get_policy, getpol	Displays the attributes of a password policy.

Command Option	Description
list_policies, listpols, get_policies, getpols	Lists the password policies.
ktadd	Adds entries to a keytab.
ktremove	Removes entries from a keytab.

3.3 Managing Realms

You can manage realms by using the kdb5_ldap_util utility.

This section provides information about the following:

- Section 3.3.1, “Creating a Realm,” on page 32
- Section 3.3.2, “Modifying a Realm,” on page 34
- Section 3.3.3, “Viewing a Realm,” on page 36
- Section 3.3.4, “Destroying a Realm,” on page 36
- Section 3.3.5, “Listing Realms,” on page 37

3.3.1 Creating a Realm

You can use one of the following methods to create a realm:

- “Command Line” on page 32
- “iManager” on page 33

Command Line

Use the following syntax to create a realm:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri] [-t
trusted_cert]

create  [-subtrees subtree_dn_list] [-sscope search_scope]
        [-containerref container_reference_dn]
        [-kdcdn kdc_service_list][-admindn admin_service_list]
        [-pwddn passwd_service_list][-defencsalttypes enc_salt_types]
        [-maxtktlife max_ticket_life]
        [-maxrenewlife max_renewable_ticket_life]
        [-ticket_flags] [-up] [-lp] [-k mkeytype]
        [-m|-P password][-sf stashfilename][-r realm]
```

For example:

```
kdb5_ldap_util -r ATHENA.MIT.EDU -D cn=admin,o=org -H ldaps://ldap-
server1.mit.edu create
-sscope 2 -kdcdn cn=service-kdc,o=org:cn=service-kdc2,o=org
-defencsalttypes des3-cbc-shal:normal -subtrees o=org
```

Output of the above command:

```
Password for "cn=admin,o=org":
Initializing database for realm ’ATHENA.MIT.EDU’
```

```
Enter KDC database master key:
Re-enter KDC database master key to verify:
```

Table 3-6 *Parameters for Creating a Realm*

Parameter	Description
-subtrees	Subtrees list where principals of the realm are placed.
-sscope	Scope for searching the principals under the specified subtree. The parameter sscope specifies the search scope for searching the principals under the subtree specified. The possible values are 1 or one (one level), 2 or sub (subtree).
-containerref	DN of the container object in which the principals of a realm will be created.
-kdcdn	List of KDC Service objects serving the realm. The list contains the DNs of the KDC Service objects separated by a colon (:).
-admindn	List of Administration Service objects serving the realm. The list contains the DNs of the Administration Service objects separated by a colon (:).
-pwddn	List of Password service objects serving the realm. The list contains the DNs of the Password service objects separated by a colon (:).
-maxtktlife	Maximum ticket life for principals in this realm.
-maxrenewlife	Maximum renewable life of tickets for principals in this realm.
-ticket_flags	Indicates the ticket flags. If this option is not set, there are no restrictions set and all ticket options are allowed.
-defencsalttypes	List of key:salt strings that specifies the default key/salt combinations for the realm. This value takes precedence over the value specified in the configuration file.
-up	Use the Universal Password of the user as the Kerberos password for the principals in the realm.
-lp	Enforce the login restrictions of the user to which the principals are attached or linked.
-k	Encryption type of the master key in the database. The default is the type given in the `krb5.conf` file.
-m	The Master password should be read from the keyboard rather than from a file or disk.
-P	Master database password.
-sf	Stash file of the master database password.
-r	Kerberos realm of the database. By default, the default_realm parameter of the `krb5.conf` file is used.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Click *Kerberos Management* > *New Realm*.

Refer to the iManager online help for more information.

3.3.2 Modifying a Realm

You can modify the realm by using one of the following methods:

- “Command Line” on page 34
- “iManager” on page 35

Command Line

Use the following syntax to modify a realm:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

modify  [-subtrees subtree_dn_list] [-sscope search_scope]
      [-containerref container_reference_dn]
      [-kdcdn kdc_service_list] [-clearkdcdn kdc_service_list]
      [-addkdcdn kdc_service_list] [-admindn admin_service_list]
      [-clearadmindn admin_service_list]
      [-addadmindn admin_service_list][-pwddn passwd_service_list]
      [-clearpwddn passwd_service_list]
      [-addpwddn passwd_service_list][-defencsalttype enc_salt_type]
      [-maxtktlife max_ticket_life|-clearmaxtklife]
      [-maxrenewlife max_renewable_ticket_life|-clearmaxrenewlife]
      [-ticket_flags] [-up|-clearup] [-lp|clearlp] [-r realm]
```

For example:

```
kdb5_ldap_util -r ATHENA.MIT.EDU -D cn=admin,o=org modify -clearkdcdn
cn=service-kdc1,o=org:cn=service-kdc2,o=org -addkdcdn cn=service-
kdc3,o=org:cn=service-kdc4,o=org -subtrees
ou=users,o=org:ou=services,o=org
```

Output of the above command:

```
Password for "cn=admin,o=org":
```

Table 3-7 *Parameters for Modifying a Realm*

Parameter	Description
-subtrees	Subtrees list containing principals in the realm.
-sscope	Scope for searching the principals under the specified subtree. The parameter sscope specifies the search scope for searching the principals under the subtree specified. The possible values are 1 or one (one level), 2 or sub (subtree).
-containerref	DN of the container object in which the principals of a realm will be created.
-kdcdn	List of KDC service objects serving the realm. The list contains the DNs of the KDC Service objects separated by a colon (:). This list replaces the existing list.

Parameter	Description
-clearkdcdn	List of KDC service objects that need to be removed from the list. The list contains the DNs of the KDC service objects separated by a colon (:).
-addkdcdn	List of KDC service objects that need to be added to the list. The list contains the DNs of the KDC service objects separated by a colon (:).
-admindn	List of Administration service objects serving the realm. The list contains the DNs of the Administration service objects separated by a colon (:). This list replaces the existing list.
-clearadmindn	List of Administration service objects that need to be removed from the list. The list contains the DNs of the Administration service objects separated by a colon (:).
-addadmindn	List of Administration service objects that need to be added to the list. The list contains the DNs of the Administration service objects separated by a colon (:).
-pwddn	List of Password service objects serving the realm. The list contains the DNs of the Password service objects separated by a colon (:). This list replaces the existing list.
-clearpwddn	List of Password service objects that need to be removed from the list. The list contains the DNs of the Administration service objects separated by a colon (:).
-addpwddn	List of Password service objects that need to be added to the list. The list contains the DNs of the Password service objects separated by a colon (:).
-defencsalttypes	List of key:salt strings that specifies the default key/salt combinations for the realm. This value takes precedence over the value specified in the configuration file.
-maxtktlife	Maximum ticket life for principals in this realm.
-clearmaxtklife	Clears the maximum ticket life value set for the realm in the directory.
-maxrenewlife	Maximum renewable life of tickets for principals in this realm.
-clearmaxrenewlife	Clears the maximum renewable ticket life value set for the realm in the directory.
-ticket_flags	Indicates the ticket flags. If this option is not set, there are no restrictions set and all tickets options are allowed.
-up	Uses the Universal Password of the user as the Kerberos password for the principals in the realm.
-clearup	Specifies not to use the Universal Password of the user as the Kerberos password.
-lp	Enforce the login restrictions of the users.
-clearlp	Exempt the login restrictions of the users.
-r	Kerberos realm of the database. By default, the default_realm parameter of the `krb5.conf` file is used.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Click *Kerberos Management* > *Edit Realm*.

Refer to the iManager online help for more information.

Modifying the Subtree for a Realm

If you modify the subtree list for a realm and the existing subtree is left out, then all the principals in that subtree are excluded from realm.

Modifying the Search Scope for a Realm

If you modify the scope for a realm, then the principals created previously under the old scope still exist, but might be excluded from the realm.

For example, if your subtree is "o=mit" that has a container "ou=students,o=mit" and you change the search scope from "sub" to "one", the Kerberos principal objects that were created under "ou=students,o=mit" still exist and are excluded from the realm.

3.3.3 Viewing a Realm

Use the following syntax to view realms:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

view      [-r realm]
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu view
-r ATHENA.MIT.EDU
```

Output of the above command:

```
Password for "cn=admin,o=org":
Realm Name: ATHENA.MIT.EDU
Subtree: ou=users,o=org
Subtree: ou=servers,o=org
SearchScope: ONE
Maximum ticket life: 0 days 01:00:00
Maximum renewable life: 0 days 10:00:00
Ticket flags: DISALLOW_FORWARDABLE
```

Table 3-8 *Parameters for Viewing a Realm*

Parameter	Description
-r	Kerberos realm of the database. By default, the default_realm parameter of the `krb5.conf` file is used.

3.3.4 Destroying a Realm

You can use one of the following methods to destroy a realm :

- “Command Line” on page 37
- “iManager” on page 37

Command Line

Use the following syntax to destroy a realm:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

destroy [-f] [-r realm]
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
destroy -r ATHENA.MIT.EDU
```

Output of the above command:

```
Password for "cn=admin,o=org":
Deleting KDC database of 'ATHENA.MIT.EDU', are you sure?
(type 'yes' to confirm)? yes
OK, deleting database of 'ATHENA.MIT.EDU'...
** Database of 'ATHENA.MIT.EDU' destroyed.
```

The principals associated with this realm are also deleted.

Table 3-9 *Parameters for Destroying a Realm*

Parameter	Description
-f	If specified, does not prompt the user for confirmation.
-r	Kerberos realm of the database. By default, the default_realm parameter of the `krb5.conf` file is used.

iManager

1. In Novell iManager, click the *Roles and Task*s button  .
2. Click *Kerberos Management* > *Delete Realm*.

Refer to the iManager online help for more information.

3.3.5 Listing Realms

Use the following syntax to list realms:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

list
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org|-H ldaps://ldap-server1.mit.edu list
```

Output of the above command:

```
Password for "cn=admin,o=org":
NOVELL.COM
ATHENA.MIT.EDU
MEDIA-LAB.MIT.EDU
```

3.4 Managing Services

You can manage the KDC, Administration, and Password services by using the `kdb5_ldap_util` command. This section provides information about the following:

- Section 3.4.1, “Creating a Service,” on page 38
- Section 3.4.2, “Modifying a Service,” on page 39
- Section 3.4.3, “Viewing a Service,” on page 40
- Section 3.4.4, “Listing Services,” on page 41
- Section 3.4.5, “Destroying a Service,” on page 42
- Section 3.4.6, “Setting a Password for Service Objects,” on page 43
- Section 3.4.7, “Setting the Server Certificate,” on page 43

3.4.1 Creating a Service

You can use one of the following methods to create a service:

- “Command Line” on page 38
- “iManager” on page 39

Command Line

Use the following syntax to create a service using `kdb5_ldap_util`:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

create_service  {-kdc|-admin|-pwd} [-servicehost service_host_list]
       [-realm realm_list][-randpw|-fileonly] [-f filename] service_dn
```

The service is created in eDirectory and appropriate rights are assigned over the realm, subtrees and principal container.

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
create_service -kdc -randpw -f /home/andrew/conf_keyfile cn=service-
kdc,o=org
```

Output of the above command is similar to the following:

```
Password for "cn=admin,o=org":
File does not exist. Creating the file /home/andrew/conf_keyfile...
```

The following table describes the configuration parameters of create_service option of the kdb5_ldap_util command:

Table 3-10 *create_service Parameters*

Parameter	Description
-kdc	KDC service.

Parameter	Description
-admin	Administration service.
-pwd	Password service.
-servicehost	List of entries separated by a colon (:) where each entry consists of the hostname or IP address of the server hosting the service, transport protocol, and the port number of the service separated by a pound sign (#). For example, server1#tcp#88:server2#udp#89.
-realm	List of realms that can be serviced by the Kerberos service being created. The list contains the names of the realms separated by a colon (:).
-randpw	Generate and set a random password. This option cannot be specified with the -fileonly option. This option does not work when Universal Password is enabled in eDirectory.
-fileonly	Stores the password only in a file and not in eDirectory. The -randpw option cannot be used if this option is specified.
-f	Complete path of the service password file where the Service object password is stashed. The default path is `/usr/local/var/service_passwd`.
servicedn	DN of the Kerberos service to be created.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *New Service*.

Refer to the iManager online help for more information.

NOTE: A service object must always be associated to a realm. During realm association, the service object is assigned the necessary rights to access the realm. A service can be associated to a realm either during Realm creation or modification, or Service creation or modification.

3.4.2 Modifying a Service

You can use one of the following methods to modify a service:

- “Command Line” on page 39
- “iManager” on page 40

Command Line

Use the following syntax to modify a service:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

modify_service  [-servicehost service_host_list |
        [-clearservicehost service_host_list]
        [-addservicehost service_host_list]]
        [-realm realm_list | [-clearrealm realm_list]
        [-addrealm realm_list]] service_dn
```

This command modifies the attributes of a service and assigns appropriate rights over the realm, subtrees and principal container.

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu -w
passwd modify_service -realm ATHENA.MIT.EDU cn=service-kdc,o=org
```

Output of the above command is similar to the following:

```
Password for "cn=admin,o=org":
Changing rights for the service object. Please wait ... done
```

The following table describes the modify_service parameters:

Table 3-11 *modify_service Parameters*

Parameter	Description
-servicehost	List of entries separated by a colon (:) where each entry consists of host name or IP address of the server hosting the service, the transport protocol, and the port number of the service separated by a pound sign (#). For example, server1#tcp#88:server2#udp#89. This service configuration parameter is not supported in this release.
-clearservicehost	List of servicehost entries to be removed from the existing list. The entries are separated by colon, where each entry consists of host name or IP address of the server hosting service, the transport protocol, and the port number of the service separated by a pound sign (#). This service configuration parameter is not supported in this release.
-addservicehost	List of servicehost entries to be added to the existing list. The entries are separated by colon, where each entry consists of host name or IP address of the server hosting service, the transport protocol, and the port number of the service separated by a pound sign (#). This service configuration parameter is not supported in this release.
-realm	List of realms that are associated with this service. The list contains the names of the realms separated by a colon (:). This list replaces the existing list.
-clearrealm	List of realms to be removed from the existing list. The list contains the names of the realms separated by a colon (:).
-addrealm	List of realms to be added to the existing list. The list contains the names of the realms separated by a colon (:).
servicedn	DN of the Kerberos service to be modified.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Edit Service*.

Refer to the iManager online help for more information.

3.4.3 Viewing a Service

Use the following syntax to view a service:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

view_service    service_dn
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
view_service cn=kdc-service1,o=org
```

Output of the above command is similar to the following:

```
Password for "cn=admin,o=org":
Service dn: cn=service-kdc,o=org
Service type: kdc
Service host list:
Realm DN list: cn=NOVELL.COM,cn=kerberos,o=novell
```

Table 3-12 *view_service Parameters*

Parameter	Description
servicedn	DN of the Kerberos service to be viewed.

3.4.4 Listing Services

Use the following syntax to list the services:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

list_service [-basedn base_dn]
```

Table 3-13 *list_service Parameters*

Parameter	Description
-basedn	Base DN for searching the services. The basedn option is made available to limit the search to a particular subtree.

This command lists the name of all existing services.

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
list_service
```

The output of the above command is similar to the following:

```
Password for "cn=admin,o=org":
cn=service-kdc,o=org
cn=service-adm,o=org
cn=service-pwd,o=org
```

3.4.5 Destroying a Service

You can use one of the following methods to destroy a service:

- “Command Line” on page 42
- “iManager” on page 42

Command Line

Use the following syntax to destroy a service:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]

destroy_service [-force] [-f stashfilename] service_dn
```

Table 3-14 *destroy_service Parameters*

Parameter	Description
destroy_service	Destroys an existing server.
-force	If specified, does not prompt for user's confirmation, but forces destruction of the service.
-f	Complete path of the service password file from where the entry corresponding to the service_dn needs to be removed.

The -f option becomes necessary if you have chosen to use a stash file of your choice while creating the service or setting the password for it. If this option is not provided, the entry for the service to be destroyed is looked up in the default stash file. Therefore, the service object is destroyed, but the entry might remain in the stash file of your choice.

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
destroy_service cn=service-kdc,o=org
```

Output of the above command is similar to the following:

```
Password for "cn=admin,o=org":
This will delete the service object 'cn=service-kdc,o=org', are you
sure?
(type 'yes' to confirm)? yes
** service object 'cn=service-kdc,o=org' deleted.
```

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Click *Kerberos Management* > *Delete Service*.

Refer to the iManager online help for more information.

3.4.6 Setting a Password for Service Objects

You can set a password for service objects such as the KDC, Administration, and Password server and store it in a file. The -fileonly option stores the password in a file and not in the eDirectory object.

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
               [-t trusted_cert]
setsrvpw  [-randpw|-fileonly] [-f filename] service_dn
```

For example:

```
kdb5_ldap_util setsrvpw -fileonly -f /home/andrew/conf_keyfile
cn=service-kdc,o=org
```

If you do not specify a filename, the default path `/usr/local/var/service_passwd` is used. When you set the service object password for the first time, the service object DN and the encrypted password are stored in the service password filename. During subsequent setting of the password for the same service object, the entry corresponding to the service object is located by comparing case insensitivity in the file, and it is replaced with the new password.

`kdb5_ldap_util` does not store the password in plain text format in the file. It is encrypted by using a unique machine-dependent key and then stored in the file.

IMPORTANT: The password file should not be edited manually. It must be modified using only the kdb5_ldap_util utility.Also, because passwords in this file are encrypted with a unique machine-dependent key, the password file becomes unusable if it is moved to a different machine.

The following table describes the configuration parameters:

Table 3-15 *setsrvpw Parameters*

Parameter	Description
-randpw	Generates and sets a random password. You can specify this option if you want to store the password both in eDirectory and a file. You cannot use the -fileonly option when you specify -randpw.
-fileonly	Stores the password only in a file and not in eDirectory. You cannot use the -randpw option when you specify -fileonly.
-f	Complete path of the service password file.
servicedn	DN of the service object whose password is to be set.

3.4.7 Setting the Server Certificate

This section describes the steps to configure the Kerberos services (KDC, Administration and Password servers) for authenticating to eDirectory using LDAP SASL EXTERNAL (CertMutual) authentication.

To set up certificate-based authentication:

1. Create a new directory. For example, `kerbcert`.
2. Change directory:

```
cd kerbcert/
```

3 Create a file called `openssl.cnf` in the `kerbcert` directory with the following contents:

```
[ req ]
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
CN=service-kdc.O=org
```

Replace CN=service-kdc.O=org with the FDN of the service object in eDirectory.

NOTE: The attribute names CN, OU, O must be in uppercase. The components of the FDN must be separated by "." (dot) and not by "," (comma).

---

4 Create a private key and certificate signing request (CSR):

4a Enter the following command:

```
openssl req -newkey rsa:1024 -keyout key.pem -out req.pem -
config openssl.cnf
```

The private key is written to `key.pem` and the certificate signing request to `req.pem`. For more information, refer to the OpenSSL Web site (http://www.openssl.org/docs/apps/openssl.html).

4b Specify the password at the prompt.

This password protects the private key.

5 Use iManager to connect to the eDirectory tree and issue a certificate as described in the *Novell Certificate Server 2.21 Administration Guide* (http://www.novell.com/documentation/crt221ad/index.html).

When prompted for the certificate signing request, specify the `req.pem` file path.

Export the issued certificate in Base 64 format (.b64) into a file called `cert.b64` in the new directory (`kerbcert` in our example).

6 Concatenate the files `key.pem` and `cert.b64` into a single `cert-key.pem` file as follows:

```
cat key.pem cert.b64 > cert-key.pem
```

7 Configure the service to use the issued certificate for authentication instead of the password as follows:

```
kdb5_ldap_util setsrvcert -f path_of_the_password_stash_file -cert
cert-key.pem service_dn
```

service_dn should be the FDN specified in the `openssl.cnf` file (CN=service-kdc.O=org as per our example). The components of the FDN must be separated by a comma.

Enter the password, when you are prompted to do so. This password is same as the one you created in Step 4b.

The service is now configured to use certificate-based authentication instead of password-based authentication.

Before starting the service, configure eDirectory to accept certificate-based authentication as follows:

1 Use iManager to modify the LDAP server SSL/TLS configuration.

Change *Client Certificate* from `Not requested` to `Requested` as described in Section 14.6 Authentication and Security in the *Novell eDirectory 8.8 Administration Guide* (http://www.novell.com/documentation/edir88/index.html).

2 Check whether the SASL EXTERNAL mechanism is installed as follows:

```
ldapsearch -x -h ldaphost -b "" -s base | grep
'supportedSASLMechanisms'
```

The SASL mechanisms supported by eDirectory are listed. Check if the EXTERNAL mechanism is in the list. If not, the mechanism must be installed as described in Section 14.6 Authentication and Security in *Novell eDirectory 8.8 Administration Guide* (http://www.novell.com/documentation/edir88/index.html).

3.5 Managing Ticket Policies

The ticket policy objects stored in eDirectory can be attached to Kerberos principals. Policy-related attributes can also be associated directly with the principals or realms but are not explained here.

3.5.1 Creating a Ticket Policy

You can use one of the following methods to add a Ticket policy:

Command Line

Use the following command to add a ticket policy:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
         [-t trusted_cert]
create_policy [-maxtktlife max_ticket_life] [-maxrenewlife
max_renewable_ticket_life] [ticket_flags] [-r realm] policy_name
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
create_policy
-r ATHENA.MIT.EDU -maxtktlife "1 day" -maxrenewlife "1 week"
-allow_postdated +needchange -allow_forwardable usertktpolicy
```

Refer to the following table for the description of the parameters:

Table 3-16 *create_policy Parameters*

Parameter	Description	
-maxtklife	Maximum ticket life for principals.	
-maxrenewlife	Maximum renewable life of tickets for principals.	
ticket_flags	Specifies the ticket flags. If this option is not specified, by default, none of the flags are set. This means that all the ticket options are allowed and no restrictions is set.	
	The various flags are:	
	{-|+}allow_postdated	Allows or prohibits principals from obtaining postdated tickets.
	{-|+}allow_forwardable	Allows or prohibits principals from obtaining forwardable tickets.
	{-|+}allow_renewable	Allows or prohibits principals from obtaining renewable tickets.
	{-|+}allow_proxiable	Allows or prohibits principals from obtaining proxiable tickets.
	{-|+}allow_dup_skey	Disables or enables user-to-user authentication for principals, by prohibiting or allowing obtaining of a session key for another user.
	{-|+}requires_preauth	Makes principals require or not require pre-authentication before being allowed to kinit.
	{-|+}requires_hwauth	Makes principals require or not require pre-authentication by using a hardware device before being allowed to kinit.
	{-|+}allow_svr	Allows or prohibits issuance of service tickets for this principal.
	{-|+}allow_tgs_req	(-)allow_tgs_req specifies that a Ticket-Granting Service (TGS) request for a service ticket for this principal is not permitted. (+) allow_tgs_req clears this flag. The default is (+) allow_tgs_req. In effect, (-)allow_tgs_req sets the KRB5_KDB_DISALLOW_TGT_BASED flag on the principal in the database.
	{-|+}allow_tix	Allows or prohibits issuance of any tickets for principals. The default is "+allow_tix".
	{-|+}needchange	Makes principals require or not require a password change.
	{-|+}password_ changing_service	Used to set or unset principals as password changing services.
-r	Realm to which the ticket policy belongs. By default, the default_realm parameter of the `krb5.conf` file is used.	
policy_name	Name of the policy.	

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *New Policy*.

Refer to the iManager online help for more information.

3.5.2 Modifying a Ticket Policy

You can use one of the following methods to modify a ticket policy:

- “Command Line” on page 47
- “iManager” on page 47

Command Line

Use the following command to modify a ticket policy:

```
kdb5_ldap_util [-D user_dn] [-w passwd]] [-H ldap_uri]
           [-t trusted_cert]
modify_policy  [-maxtktlife max_ticket_life] [-maxrenewlife
max_renewable_ticket_life] [ticket_flags] [-r realm] policy_name
```

For more information on the parameters, refer to Table 3-16 on page 46.

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
modify_policy -maxtktlife "1 day" -maxrenewlife "1 week"
+allow_postdated -requires_preauth usertktpolicy
```

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Edit Policy*.

Refer to the iManager online help for more information.

3.5.3 Destroying a Ticket Policy

You can use one of the following methods to destroy a ticket policy:

- “Command Line” on page 47
- “iManager” on page 48

Command Line

Use the following command to destroy a ticket policy:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
          [-t trusted_cert]
destroy_policy  [-force] [-r realm] policy_name
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
destroy_policy   -r ATHENA.MIT.EDU usertktpolicy
This will delete the policy object 'usertktpolicy', are you sure?
(type 'yes' to confirm)? Yes
* policy object 'usertktpolicy' deleted.
```

Table 3-17 destroy_policy Parameters

Parameter	Description
-force	Forces the deletion of the policy object. If you do not specify this option, you are prompted for confirmation while deleting the policy. Enter YES to confirm the deletion.
-r	Realm to which the ticket policy belongs. By default, the default_realm parameters of the `krb5.conf` file is used.
policy_name	Name of the policy.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Delete Policy*.

Refer to the iManager online help for more information.

3.5.4 Viewing a Ticket Policy

Use the following command to view a ticket policy:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
           [-t trusted_cert]
view_policy [-r realm] policy_name
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
view_policy -r ATHENA.MIT.EDU usertktpolicy
```

The expected output is:

```
Ticket Policy: usertktpolicy
Maximum ticket life: 0 days 01:00:00
Maximum renewable life: 0 days 10:00:00
Ticket flags: DISALLOW_FORWARDABLE REQUIRES_PWCHANGE
```

Table 3-18 view_policy Parameters

Parameter	Description
-r	Realm to which the ticket policy belongs. By default, the default_realm parameters of the `krb5.conf` file is used.
policy_name	Name of the policy.

3.5.5 Listing Ticket Policies

Use the following command to list policies:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
           [-t trusted_cert]
list_policy [-r realm]
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
list_policy
```

The expected output is as follows:

```
usertktpolicy
tempusertktpolicy
krbtktpolicy
```

Table 3-19 *view_policy Parameters*

Parameter	Description
-r	Realm for which the ticket policies are to be listed. By default, the default_realm parameters of the `krb5.conf` file is used.

3.6 Updating Kerberos LDAP Extension Information

You can use the `kdb5_ldap_util` utility to update the ldapExtensionInfo attribute on the LDAP server object as follows:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
          [-t trusted_cert]
ldapxtn_info -add|-clear
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
ldapxtn_info -add
```

Table 3-20 *ldapxtn_info Parameters*

Parameter	Description
-add	Adds Kerberos LDAP extension information (OIDs for Kerberos LDAP Extensions) to ldapExtensionInfo on the LDAP server object.
-clear	Removes Kerberos LDAP extension information (OIDs for Kerberos LDAP Extensions) from ldapExtensionInfo on the LDAP server object.

3.7 Setting the Master Key

If the master key of a realm in eDirectory is corrupted, you can use `kdb5_ldap_util` to reset it. Ensure that the master key is reset with the same master password and key type, which was provided while creating the realm. Otherwise, all the principals in the realm will be unusable.

If you change the master key of a realm, then the existing principals cannot access any Kerberos services in the network, because their secret keys were encrypted with the old master key. If you want to change the master key, you must delete and reset the keys for all the principals in the realm.

You can reset the master key as follows:

```
kdb5_ldap_util [-D user_dn [-w passwd]] [-H ldap_uri]
           [-t trusted_cert]
setmasterkey [-k mkeytype] [-m|-P password] [-r realm]
```

For example:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu
setmasterkey -r ATHENA.MIT.EDU
```

Table 3-21 *setmasterkey Parameters*

Parameter	Description
-k	Specifies the key type of the master key for the realm. If not specified, the default value is used. The default value is DES3_HMAC_SHA1.
-m	Specifies that the master password should be read from the keyboard.
-P	Specifies the master password. We do not recommend that you use this.
-r	Specifies the Kerberos realm of the database. By default, the default_realm parameter of the configuration file (`/etc/krb5.conf`) is used.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Set Master Key*.

Refer to the iManager online help for more information.

NOTE: Enter the same master password that was provided during creation of the realm.

3.8 Managing Principals

You can manage principals through kadmin. This section explains the following:

3.8.1 Adding a Principal

Principals can be created under the realm subtrees, principal container, or realm container. Prinicipals can be created in any of the following ways:

- Attached to an existing LDAP object. The LDAP object should exist within the subtree or principal container.
- Created as a separate principal object, which can be optionally linked to an LDAP object. The principal can be created under a specific container by providing the option during principal creation. Otherwise, it is created under principal container (if it is configured) or the realm container. If a container is provided, it must be within the subtree or principal container.

You can use one of the following methods to add a principal :

- “Command Line” on page 51
- “iManager” on page 54

Command Line

To create a principal, enter the following at the kadmin prompt:

```
add_principal [options] principal
```

options are:

```
[-x db_princ_args] [-expire expdate] [-pwexpire pwexpdate] [-maxlife
maxlife] [-maxrenewlife maxrenewlife] [-kvno kvno] [-policy policy]
[{+|-}attribute]
```

attributes are:

```
allow_postdated allow_forwardable allow_tgs_req allow_renewable
allow_proxiable allow_dup_skey allow_tix requires_preauth
requires_hwauth needchange allow_svr password_changing_service
```

Table 3-22 *add_principal Parameters*

Parameter	Description
-x	Denotes the database-specific options. The following are the options for LDAP as the back end: ◆ -x dn=<dn> LDAP object that will contain the Kerberos principal being created. ◆ -x linkdn=<linkdn> LDAP object to which the newly created Kerberos principal object will point to. ◆ -x containerdn=<container_dn> Container under which the Kerberos principal is to be created. ◆ -x tktpolicy=<policyname> Associates a ticket policy object to the Kerberos principal. ◆ -x up=<on|off> Specifies if the Kerberos User Principal associated with the eDirectory user object will make use of the Universal Password.
-expire	Expiration date of the principal
-pwexpire	Password expiration date
-maxlife	Maximum ticket life for the principal
-maxrenewlife	Maximum renewable life of tickets for the principal.
-kvno	Explicitly sets the key version number.
-policy	Password policy used by this principal. If no policy is supplied, and if the default policy exists and the -clearpolicy is also not specified, then the default policy is used; otherwise, the principal has no password policy, and a warning message will be printed.
-clearpolicy	Prevents the default policy from being assigned when (-) policy is not specified. This option has no effect if the default policy does not exist.
{-|+}allow_postdated	(-) allow_postdated prohibits this principal from obtaining postdated tickets. (Sets the KRB5_KDB_DISALLOW_POSTDATED flag.) (+) allow_postdated clears this flag.
{-|+}allow_forwardable	(-) allow_forwardable prohibits this principal from obtaining forwardable tickets. (Sets the KRB5_KDB_DISALLOW_FORWARDÂABLE flag.) (+) allow_forwardable clears this flag.
{-|+}allow_renewable	(-) allow_renewable prohibits this principal from obtaining renewable tickets. (Sets the KRB5_KDB_DISALLOW_RENEWABLE flag.) (+) allow_renewable clears this flag.
{-|+}allow_proxiable	(-) allow_proxiable prohibits this principal from obtaining proxiable tickets. (Sets the KRB5_KDB_DISALLOW_PROXIABLE flag.) (+) allow_proxiable clears this flag.

Parameter	Description
{-|+}allow_dup_skey	(-) allow_dup_skey disables user-to-user authentication for this principal by prohibiting this principal from obtaining a session key for another user. (Sets the KRB5_KDB_DISALLOW_DUP_SKEY flag.) (+) allow_dup_skey clears this flag.
{-|+}requires_preauth	(+) requires_preauth requires this principal to preauthenticate before being allowed to kinit. (Sets the KRB5_KDB_REQUIRES_PRE_AUTH flag.) (-) requires_preauth clears this flag.
{-|+}requires_hwauth	(+) requires_hwauth requires this principal to preauthenticate by using a hardware device before being allowed to kinit. (Sets the KRB5_KDB_REQUIRES_HW_AUTH flag.) (-) requires_hwauth clears this flag.
{-|+}allow_svr	(-) allow_svr prohibits the issuance of service tickets for this principal. (Sets the KRB5_KDB_DISALLOW_SVR flag.) (+) allow_svr clears this flag.
{-|+}allow_tgs_req	(-) allow_tgs_req specifies that a Ticket-Granting Service (TGS) request for a service ticket for this principal is not permitted. (+) allow_tgs_req clears this flag. The default is (+) allow_tgs_req . In effect, (-) allow_tgs_req sets the KRB5_KDB_DISALLOW_TGT_BASED flag on the principal in the database.
{-|+}allow_tix	(-) allow_tix forbids the issuance of any tickets for this principal. (+) allow_tix clears this flag. The default is (+) allow_tix . In effect, (-) allow_tix sets the KRB5_KDB_DISALLOW_ALL_TIX flag on the principal in the database.
{-|+}needchange	(+) needchange sets a flag in attributes field to force a password change; (-) needchange clears it. The default is (-) needchange . In effect, (+) needchange sets the KRB5_KDB_REQUIRES_PWCHANGE flag on the principal in the database.
{-|+}password_changing_service	(+) password_changing_service sets a flag in the attributes field marking this as a password change service principal. (-) password_changing_service clears the flag. This flag intentionally has a long name. The default is (-) password_changing_service. In effect, (+) password_changing_service sets the KDB_PWCHANGE_SERVICE flag on the principal in the database.

Creating a Principal

To create a user principal with default values, enter the following at the kadmin prompt:

```
kadmin: addprinc jennifer
```

The output of the above command is similar to the following:

```
WARNING: no policy specified for "jennifer@ATHENA.MIT.EDU";
defaulting to no policy.

Enter password for principal jennifer@ATHENA.MIT.EDU:  <= Type the
password.

Re-enter password for principal jennifer@ATHENA.MIT.EDU:  <=Type it
again.
```

```
Principal "jennifer@ATHENA.MIT.EDU" created.

kadmin:
```

To create a principal that is contained by an LDAP object, enter the following at the kadmin prompt:

```
kadmin: addprinc -x dn=cn=jennifer,o=mit jennifer.
```

The object cn=jennifer,o=mit must exist in the directory.

The output of the above command is similar to the following:

```
WARNING: no policy specified for
"jennifer@ATHENA.MIT.EDU";defaulting to no policy.

Enter password for principal jennifer@ATHENA.MIT.EDU:  <= Type the
password.

Re-enter password for principal jennifer@ATHENA.MIT.EDU:  <=Type it
again.

Principal "jennifer@ATHENA.MIT.EDU" created.

kadmin:
```

To create a principal under a specific LDAP container and link to an existing LDAP object, enter the following at the kadmin prompt:

```
kadmin: addprinc -x containerdn=o=mit -x linkdn=cn=david,o=mit
david
```

The output of the above command is similar to the following:

```
WARNING: no policy specified for "david@ATHENA.MIT.EDU"; defaulting
to no policy.

Enter password for principal david@ATHENA.MIT.EDU:  <= Type the
password.

Re-enter password for principal david@ATHENA.MIT.EDU:  <=Type it
again.

Principal "david@ATHENA.MIT.EDU" created.

kadmin:
```

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *New Principal*.

Refer to the iManager online help for more information.

3.8.2 Modifying a Principal

You can use one of the following methods to modify a principal:

- “Command Line” on page 55
- “iManager” on page 55

Command Line

To modify principals, enter the following at the kadmin command prompt:

```
modify_principal [options] principal
```

options are:

```
 [-x db_princ_args]* [-expire expdate] [-pwexpire pwexpdate]
[-maxlife maxtixlife] [-maxrenewlife maxrenewlife] [-kvno kvno]
[-policy policy] [-clearpolicy] [{+|-}attribute]
```

attributes are:

```
allow_postdated allow_forwardable allow_tgs_req allow_renewable
allow_proxiable allow_dup_skey allow_tix requires_preauth
requires_hwauth needchange allow_svr password_changing_service
```

For details about the parameters, refer to Table 3-22 on page 52.

For example:

```
modify_principal -x up=off -x tktpolicy=usertktpolicy -policy
pwdpolicy +requires_preauth Jennifer
```

The output of the above command is similar to the following:

```
Principal "Jennifer@MYREALM" modified.
```

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Edit Principal*.

Refer to the iManager online help for more information.

3.8.3 Deleting a Principal

You can use one of the following methods to delete a principal:

- “Command Line” on page 55
- “iManager” on page 56

Command Line

To delete a principal, enter the following at the kadmin command prompt:

```
delete_principal [-force] principal
```

If the -force option is not specified, you are prompted to confirm the deletion. If the Kerberos principal is attached to the LDAP object, the `delete_principal` command does not delete the LDAP object but only deletes the Kerberos attributes.

For example:

```
delete_principal David
```

The output of the above command is similar to the following:

```
Are you sure you want to delete the principal "David@MYREALM"? (yes/
no): yes
Principal "David@MYREALM" deleted.
Make sure that you have removed this principal from all ACLs before
reusing.
```

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Delete Principal*.

Refer to the iManager online help for more information.

3.8.4 Listing Principals

To list principals, enter the following at the kadmin prompt:

```
list_principals [expression]
```

Expression is a shell-style glob expression that can contain the characters *, ?, [, and]. All policy names matching the expression are displayed. The list_principals command has the aliases listprincs, get_principals, and getprincs.

For example:

```
list_principals princ*
```

The output of the above command is similar to the following:

```
princ@MYREALM
princ1@MYREALM
princ2@MYREALM
```

3.8.5 Getting Principal Information

To get the attributes of a principal, enter the following at the kadmin command prompt:

```
get_principal [-terse] principal
```

For example:

```
get_principal jennifer/root
```

The output of the above command is similar to the following:

```
Principal: jennifer/root@ATHENA.MIT.EDU
Expiration date: [never]
Last password change: Mon Jan 31 02:06:40 EDT 2002
Password Expiration date: [none]
Maximum ticket life: 0 days 10:00:00
```

```
Maximum renewable life: 7 days 00:00:00
Last modified: Wed Jul 24 14:46:25 EDT 2002 (joeadmin/
admin@ATHENA.MIT.EDU)
Last successful authentication: Mon Jul 29 18:20:17 EDT 2002
Last failed authentication: Mon Jul 29 18:18:54 EDT 2002
Failed password attempts: 3
Number of keys: 2
Key: vno 2, Triple DES cbc mode with HMAC/sha1, no salt
Key: vno 2, DES cbc mode with CRC-32, no salt
Attributes: DISALLOW_FORWARDABLE, DISALLOW_PROXIABLE
Policy: [none]
kadmin:
```

3.8.6 Setting Principal Password

You can use one of the following methods to set principal password:

- "Command Line" on page 57
- "iManager" on page 58

Command Line

To change the password of a principal, enter the following at the kadmin prompt:

```
change_password [-randkey] [-keepold] [-e keysaltlist] [-pw password]
principal
```

Table 3-23 *change_password Parameters*

Parameter	Description
-randkey	Sets the key of the principal to a random value.
-keepold	Keeps the previous kvno's keys. There is no easy way to delete the old keys, and this flag is usually not necessary except perhaps for TGS keys. Don't use this flag unless you are sure you want to use it. This option is not supported for this release.
-e	Uses the specified list of enctype-salttype pairs for setting the key of the principal. The quotes are necessary if there are multiple enctype-salttype pairs.
-pw	Sets the password to the specified string. We do not recommend that you use it.

For example:

```
change_password Jennifer
```

The output of the above command is similar to the following:

```
Enter password for principal "Jennifer":
Re-enter password for principal "Jennifer":
Password for "Jennifer@MYREALM" changed.
```

```
change_password -pw secret Jennifer
```

The output of the above command is similar to the following:

```
Password for "Jennifer@MYREALM" changed.
```

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Set Principal Password*.

Refer to the iManager online help for more information.

3.8.7 Extracting a Principal Key to a Keytab File

To extract the principal key to a keytab file, enter the following command at the kadmin prompt:

```
ktadd [-k[eytab] keytab] [-q] [-e keysaltlist] principal | -glob
princ-exp [...]
```

Table 3-24 *ktadd Parameters*

Parameter	Description
-keytab	Specifies the keytab file path.
-q	Displays less verbose status information.
-e	Uses the specified list of enctype-salttype pairs for setting the key of the principal. The quotes are necessary if there are multiple enctype-salttype pairs.
-principal|-glob	Add a principal or all principals matching the principal expression to the keytab.

For example:

```
ktadd -k /etc/key-tab David
```

The output of the above command is similar to the following:

```
Entry for principal David with kvno 2, encryption type Triple DES cbc
mode with HMAC/sha1 added to keytab WRFILE:/etc/key-tab.
```

3.8.8 Removing a Keytab Entry

To remove entries from a keytab, enter the following command at the kadmin prompt:

```
ktremove [-keytab keytab] [-q] principal [kvno|"all"|"old"]
```

Table 3-25 *ktremove Parameters*

Parameter	Description
-keytab	Specifies the keytab file path.
-q	Displays less verbose status information.
kvno	Removes all entries for the specified principal whose key version numbers match `kvno`.

Parameter	Description
all	Removes all entries for the specified principal.
old	Removes all entries for the specified principal, except those with the highest kvno.

For example:

```
ktremove -k /etc/key-tab user_Davud all
```

The output of the above command is similar to the following:

```
Entry for principal user_David with kvno 2 removed from keytab WRFILE:/
etc/key-tab.
```

3.9 Managing Password Policies

The policy management commands in the MIT kadmin utility were modified to work with an LDAP directory. The policies control the password of the Kerberos principals. The Kerberos password policies come into effect only when the Kerberos passwords of the principals are different from the eDirectory user passwords. When the Kerberos passwords are the same as the user's passwords, the NSPM password policy is effective.

- Section 3.9.1, “Adding a Password Policy,” on page 59
- Section 3.9.2, “Modifying a Password Policy,” on page 60
- Section 3.9.3, “Deleting a Password Policy,” on page 60
- Section 3.9.4, “Viewing Policy Values,” on page 61
- Section 3.9.5, “Listing Policies,” on page 61

3.9.1 Adding a Password Policy

You can use one of the following methods to add a password policy:

- “Command Line” on page 59
- “iManager” on page 60

Command Line

This command creates a password policy object, in the directory.

```
add_policy [-maxlife time] [-minlife time] [-minlength length] [-
minclasses number] [-history number] policy
```

Table 3-26 *add_policy Parameters*

Parameter	Description
-maxlife	Maximum lifetime of a password.
-minlife	Minimum lifetime of a password.
-minlength	Minimum length of a password.

Parameter	Description
-minclasses	Minimum number of character classes allowed in a password.
-history	Number of past keys kept for a principal. Not supported.

For example, enter the following at the kadmin prompt:

```
add_policy -maxlife "2 days" -minlength 5 guestpolicy
```

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *New Password Policy*.

Refer to the iManager online help for more information.

3.9.2 Modifying a Password Policy

You can use one of the following methods to modify the password policy:

- “Command Line” on page 60
- “iManager” on page 60

Command Line

To modify a policy, enter the following at the kadmin prompt:

```
modify_policy [-maxlife time] [-minlife time] [-minlength length] [-
minclasses number] [-history number] policy
```

For more information on the options, refer to Table 3-26 on page 59.

For example, enter the following at the kadmin prompt:

```
modify_policy -minlife "1 day" -minclasses 2 guestpolicy
```

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Edit Password Policy*.

Refer to the iManager online help for more information.

3.9.3 Deleting a Password Policy

You can use one of the following methods to delete a password policy:

- “Command Line” on page 60
- “iManager” on page 61

Command Line

This command deletes the specified policy from the directory. It fails if the policy is in use by any principal.

To delete a policy, enter the following at the kadmin prompt:

```
delete_policy [-force] policy
```

For example, enter the following at the kadmin prompt:

```
delete_policy guestpolicy
```

You are prompted to confirm the deletion as follows:

```
Are you sure you want to delete the policy "guestpolicy"? (yes/no):
```

Enter `yes` to proceed with the deletion.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *Delete Password Policy*.

Refer to the iManager online help for more information.

3.9.4 Viewing Policy Values

You can view the values of the specified policy as follows:

```
get_policy [-terse] policy
```

The -terse flag outputs the fields as quoted strings separated by tabs.

For example:

```
get_policy guestpolicy
```

This gives the following output:

```
Policy: guestpolicy
Maximum password life: 172800
Minimum password life: 86400
Minimum password length: 5
Minimum number of password character classes: 2
Number of old keys kept: 1
Reference count: 0
```

3.9.5 Listing Policies

You can list all the password policies as follows:

```
list_policies
```

This gives the following output:

```
kadmin:  listpols
test-pol
dict-only
once-a-min
test-pol-nopw

kadmin:  listpols t*
test-pol
test-pol-nopw
```

```
kadmin:
```

Integrating Universal Password

4

Universal Password is a single password to eDirectory™. It allows synchronization of passwords from eDirectory to other systems. For more information about Universal Passwords, refer to *Novell Modular Authentication Services (NMAS) 3.2 Administration Guide* (http://www.novell.com/documentation/password_management/index.html).

4.1 Configuring Universal Passwords

Novell® Kerberos KDC can be integrated with Universal Password so that there is a single password to authenticate to eDirectory and Kerberos. The eDirectory and Kerberos passwords are synchronized by using the Kerberos Password Agent.

4.1.1 Prerequisites

- ❑ Enable Universal Password in eDirectory.

  You can enable Universal Password at the tree, container, or user level.

  For more information, refer the Deploying Universal Password section in *Novell Password Management Administration Guide* (http://www.novell.com/documentation/nmas32/admin/index.html?page=/documentation/nmas32/admin/data/a20gkue.html)
- ❑ Ensure that the *Synchronize Distribution Password while setting Universal Password* option is enabled in the password policy at the level (tree, container, or user) that is in effect.

4.1.2 Integrating Universal Password with the Novell Kerberos KDC

To enable Universal Password, first enable it at the realm or user level, then set or modify the password as follows.

- “Enabling Universal Passwords at the Realm Level” on page 63
- “Enabling Universal Password at the User Level” on page 64
- “Setting or Modifying the Universal Password” on page 65

Enabling Universal Passwords at the Realm Level

Enable Universal Password at the time of creating the realm. Alternatively, after the realm is created, you can enable universal passwords by editing the realm.

You can use any of the methods to enable Universal Passwords:

- “Command Line” on page 63
- “iManager” on page 64

Command Line

```
kdb5_ldap_util -H ldaps://ldap-server1.mit.edu -D cn=admin,o=org -r
ATHENA.MIT.EDU create -subtrees o=org -up
```

```
kdb5_ldap_util -H ldaps://ldap-server1.mit.edu -D cn=admin,o=org -r
ATHENA.MIT.EDU modify -up
```

NOTE: To disable Universal Password, use the above command with the -clearup option.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Click *Kerberos Management* > *New Realm*.

   If you are modifying the realm, click *Kerberos Management* > *Edit Realm*.

Refer to the iManager online help for more information.

NOTE: If the Universal Password is configured for a principal, this configuration takes precedence over the realm level configuration.

Enabling Universal Password at the User Level

You can enable Universal Password for the principal at the time of creating the principal. Alternatively, after the principal is created, you can edit the principal.

- "Command Line" on page 64
- "iManager" on page 64

Command Line

```
add_principal -x up=on -x dn=cn=alice,o=org alice
modify_principal -x up=on alice
```

NOTE: To disable Universal Password, use the above commands with up=off or up=clr options. To use the realm level configuration when the Universal Password option is enabled for the principal, use the above command with up=clr option.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Select *Kerberos Management* > *New Principal*.

   If you are modifying the realm, click *Kerberos Management* > *Edit Principal*.

Refer to the iManager online help for more information.

Use the following table to check if Universal Password has been enabled for a user.

Table 4-1 *Checking if Universal Password Is Enabled or Disabled*

Universal Password Configuration Level		Is Universal Password Enabled?
`Realm`	`Principal`	
True	True	Yes
True	False	No

Universal Password Configuration Level		Is Universal Password Enabled?
False	True	Yes
False	False	No
True	Not Present	Yes
False	Not Present	No
Not Present	True	Yes
Not Present	False	No
Not Present	Not Present	No

Ensure that the Kerberos Password Agent is loaded when Universal Password is enabled in Kerberos and eDirectory. If the Kerberos Password Agent is not running, the passwords are not synchronized when the Universal Password is changed in eDirectory. Additionally, when the password is changed by using cpw or kpasswd in Kerberos, the principal's Kerberos key version might not be consistent.

Setting or Modifying the Universal Password

When a new principal is added, Kerberos password and Universal Password are not synchronized. The Kerberos keys are generated from the password specified while adding the principal. For the Kerberos password to be the same as the Universal Password, the Universal Password of the user must be modified after the principal is created. You can set or modify the Universal Password either in eDirectory or Kerberos.

In Kerberos, if the principal is Universal password enabled and if kpasswd or cpw is used to change the Kerberos password, it modifies the Universal Password of the user with which the principal is associated. This also leads to synchronization of passwords for all the principals that are associated with that user and that have Universal Password enabled.

4.2 Kerberos Password Agent

The Kerberos Password Agent (KPA) synchronizes the Kerberos password with Universal Password based on the configuration at the realm and user. It is sufficient to install the KPA on one of the eDirectory servers with the writable replica of the Kerberos data.

To start the KPA, enter the following:

```
kpa -l
```

To stop the KPA, enter the following:

```
kpa -u
```

The messsages logged by the Password Agent are displayed when the Misc tag is enabled in the ndstrace. The messages are also logged in the log file that is configured for the eDirectory server.

IMPORTANT: The Kerberos Password Agent is not loaded automatically when the machine or eDirectory is restarted. It msut be loaded manually.

Key Generation

The encryption types and salt type used by the Kerberos Password Agent to generate the Kerberos keys from the Universal Password are based on the following:

- If the principal has Kerberos keys, the encryption and salt types used in generating the existing keys are used to generate the new keys from the Universal Password.
- If the principal does not have the Kerberos password set, the default encryption salt types configured for the realm are used for the key generation.

If the default key types are not configured for the realm, the key types used are DES3-HMAC-SHAI:NORMAL and DES-CBC-CRC:NORMAL.

For more information on the supported encryption and salt types, refer to Appendix B, “Supported Encryption Types and Salt Types,” on page 93.

4.3 Universal Password Considerations

- If the Universal Password is enabled, the randkey option cannot be used for setting Universal Password while changing the password of a principal.
- Setting the password for a principal associated with a user object sets Universal Password as the Kerberos password for all the principals that are Universal Password enabled and associated with that user object.
- If Universal Password is enabled, the Kerberos Password Agent module should be loaded whenever the machine or eDirectory is restarted.
- The Novell Kerberos KDC does not support extended characters in a password. If the Kerberos password is integrated with Universal Password, the Universal Password also cannot have extended characters.

Enforcing Login Restrictions

5

The Enforce Login Restrictions feature of the Novell® Kerberos KDC allows you to do the following:

- Enforce the login restrictions of the eDirectory™ users for Kerberos authentications. For every authentication request by a Kerberos principal, the Novell Kerberos KDC checks for the associated eDirectory user's various login based restrictions, such as address restrictions, time restrictions, intruder detection, account disabled, login expiration, password expiration, and grace logins expiration.
- Update the login statistics for both successful and failed authentication attempts to enforce the login restrictions configured. For example, the account is locked when the configured number of bad password attempts is exceeded.

5.1 Configuring Enforce Login Restrictions

This section provides information about the following:

- Section 5.1.1, "Prerequisites," on page 67
- Section 5.1.2, "Enabling Login Restrictions," on page 67

5.1.1 Prerequisites

- ❑ Install NMAS™ Server 3.1.
- ❑ Set up intruder detection for all users in the subtrees configured for the realm.
- ❑ Set up extra login security for a user.

  For more information, refer to the Managing User Accounts (http://www.novell.com/documentation/edir88/index.html) section in the *Novell eDirectory 8.8 Administration Guide*.
- ❑ (Conditional) If Universal Password is used as the Kerberos Password, then create and assign a password policy for users.

  Ensure that the *Enable the Advanced Password Rules* option in Novell iManager is enabled in the password policy at the tree, container, or user level. For more information, refer to the Managing Passwords by Using Password Policies (http://www.novell.com/documentation/password_management/index.html) section in the *Novell Password Administration Guide*.
- ❑ Enable pre-authentication for Kerberos principals. You can enable pre-authentication at the realm or principal level. For more information, refer to the following sections:
  - Section 3.3, "Managing Realms," on page 32
  - Section 3.8, "Managing Principals," on page 50

5.1.2 Enabling Login Restrictions

You can use either of the following methods to enable the Enforce Login Restrictions feature:

- "Command Line" on page 68
- "iManager" on page 68

Command Line

Use the following syntax to enable this feature:

```
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu -r
ATHENA.MIT.EDU create -subtrees o=org -lp
kdb5_ldap_util -D cn=admin,o=org -H ldaps://ldap-server1.mit.edu -r
ATHENA.MIT.EDU modify -lp
```

To disable login restrictions, use the above command with the `-clearlp` option.

iManager

1. In Novell iManager, click the *Roles and Tasks* button  .
2. Click *Kerberos Management* > *New Realm*.

   If you are modifying the realm, click *Kerberos Management* > *Edit Realm*.

Refer to the iManager online help for more information.

5.2 Login Restrictions Considerations

- The performance of KDC is affected when users request for Ticket Granting Tickets.
- If the eDirectory user has grace logins, they are used for Kerberos authentications, and principals are not allowed to change their passwords after the grace logins are exhausted.
- Login restrictions are applicable only while the users acquire Ticket Granting Tickets, and are not applicable for subsequent service ticket requests.
- The KDC service object is granted Write rights for the ACL attribute on the subtrees of the realm.

Kerberzing eDirectory Users

6

This section provides information on enabling existing eDirectory™ users to use the Novell® Kerberos KDC.

Existing eDirectory users are typically created and managed by the `kadmin` or `kadmin.local` utility or through the Kerberos iManager plug-in. In these methods, the administrator manually kerberizes every user object.

However, you can use the kerberize tool to kerberize one or more eDirectory user objects at a time.

6.1 Prerequisites

- Install and configure Novell Kerberos KDC 1.5

  For more information on installing and configuring the Novell KDC, refer to the Novell Kerberos KDC Quick Start Guide (http://www.novell.com/documentation/kdc15/quickst/index.html?page=/documentation/kdc15/quickst/data/bookinfo.html)

- Ensure that the Novell Kerberos KDC LDAP Extension (libkrbpwd) is loaded on eDirectory.

6.2 Using the Kerberize Tool to Kerberize User Objects

The kerberize tool creates the krbPrincipalName attribute for eDirectory user objects, either with a default password (set in the configuration file) or it reads the Universal Password and sets the Kerberos key.

The syntax for using the tool is as follows:

kerberize -t trustedcert -D userdn [-w passwd | -W] [-H ldapuri] [-f config-file] [-l logfile] [-c]

Table 6-1 *Syntax for Kerberize tool*

Parameter	Description
-t	Trusted root certificate of the LDAP Server specified in DER or B64 format, for binding to it over SSL/TLS. If the extension is missing or different from these formats, then the file will be encoded in B64 format.
-D	DN of the administrator who have sufficient rights to modify the matching DNs and to associate the newly created principals to them.
-w	Password of the admindn for binding to LDAP Server. Usage of this option is not recommended.
-W	Prompt for the password.

Parameter	Description
-H	The URI of the LDAP Server. For example, "ldaps://acme.org:636".
-f	Path of the configuration file containing the required information to create and associate kerberos principals to matching DNs. The default value is "/etc/opt/novell/kerberos/conf/kerberize.conf".
-l	The path of the file for logging the information. The default value is "/var/opt/novell/kerberos/log/kerberize.log".
-c	Continuous operation mode. Errors are reported, but kerberize will continue with operations. The default is to exit after reporting an error.

The binary is located at `/opt/novell/kerberos/sbin`.

1. Edit the `kerberize.conf` configuration file and specify the following details. The file is located at `/etc/opt/novell/kerberos/conf/kerberize.conf`.
   - Kerberos realm
   - Subtree
   - Kerberos principal and key creation rules
   - (optional) Overrides
2. Execute the kerberize tool with the configuration file as an input along with other required information to connect to LDAP server.

   For example:

   ```
   kerberize -D cn=eDir_administrator,o=org  -H ldaps://
   ldapserver.org:636 -t TurstedRootCert.der -f /etc/opt/novell/
   kerberos/conf kerberos.conf
   ```

6.3 Configuration Parameters

This section provides information on the parameters in the `kerberize.conf` configuration file. You can use the file to set principal names and their Kerberos passwords. For a sample configuration file, refer to Section 6.4, “Sample kerberize.conf File,” on page 73.

The `kerberize.conf` file parameters are listed below:

Table 6-2 *Kerberize.conf Parameters*

Parameter	Description
`general`	
noofoperations	Specifies the number of operation(s) to be performed by the kerberize tool, where each operation is defined under [operation-n] section. For example: `noofoperations = 2`

Parameter	Description
operation-1	
principal operation	Specifies the type of the principal operation. The possible values are *add* and *remove*. If the value is *add*, the operation enables Kerberos for the matching DN and if the value is *remove*, it disables the matching DN from using the Kerberos protocol. For example: `principal-operation = add`
realm	Identifies the default realm to be used for this operation. A Kerberos principal will either be added or removed from this realm, based on the type of the principal-operation specified. For example: `realm = EXAMPLE.COM`
base	Specifies the base DN, where the search of DNs should start. If this option is not provided, it searches the entire tree. For example: `base = ou=users,o=org`
scope	Specifies the scope for searching the DNs on which the operation is to be applied. The possible values are *base* (base object), *one* (one level) and *sub* (subtree). For example: `scope = sub`

filter	Specifies the filter to be used for searching the DN on an LDAP Server. Filter should conform to the string representation for search filters. If not provided, the default filter, (objectClass=*), is used.
	For example:
	`1. filter = (objectClass=*) matches all the DNs`
	`2. filter = (&(objectClass=inetOrgPerson) (cn=*)) matches all users`

principal-name	Specifies the regular expression that provides the Kerberos principal name to be added or removed, by evaluating the regular expression value. For example: `[^...]({cn}){sn} generates a principal which contains first three characters of cn attriubte and the complete sn attribute value of the eDirectory DN, in that order.`

Parameter	Description
password	Specifies a default value, a random value, or a regular expression. For example: `password = {RAND}` This expression generates a four-character random value as the password.
policy	Specifies the (password) policy DN to be used by the principal. The password policy DN should be existing; if it does not, the operation does not succeed. For example: `policy = cn=passwd-policy,o=org`
tktpolicy	Specifies the ticket policy DN to be used by the principal. The ticket policy DN should be existing, failing which the operation will not succeed. For example: `tktpolicy = cn=ticket-policy,o=org`
up	Specifies whether Universal Password is enabled. The possible values are *true* and *false*. For example: `up = true` If the value is *true*, the existing Universal password is set as kerberos password. To make this work, you must enable Universal Password in eDirectory.
expire	Expiration time of the principal. Use the format yyyymmddhhmmssz.
pw expire	Expiration time of the password of the principal. Use the format yyyymmddhhmmssz.
dn	Specifies the DN of the entry that is to be overridden. For example: `dn = cn=user1,ou=users,o=org`
noofoverrides	Specifies the number of overrides to be used. Among the DNs identified based on the search filter and scope, some of them can have different values than that of the values specified in [operation-n] section. These overriding rules can be specified in "override-m" subsection. For example: `noofoverrides = 3`

Parameter	Description
override-m	Some eDirectory users can be overridden with specific values. The override section takes the following parameters: ◆ dn ◆ principal-name ◆ password ◆ expire ◆ pwexpire ◆ policy ◆ tktpolicy ◆ up

6.4 Sample kerberize.conf File

A sample `kerberize.conf` file is provided in the `/etc/opt/novell/kerberos/conf` directory. You can use this configuration file to set the principal names and Kerberos passwords for eDirectory users.

NOTE: You can perform only add principal operation or remove principal operation at a time. You cannot perform both operations at the same time. If both operation types need to be performed, run the kerberize tool twice specifying `add` and then `remove` as the operation type in the configuration file.

```
[general]
   noofoperations = 1

[operation-1]
   principal-operation = add
   realm = EXAMPLE.COM
   base = ou=users,o=org
   scope = sub
   filter = (&(objectClass=inetOrgPerson)(cn=*))
   principal-name = [^.]({cn}){sn}
   password = {RAND}
#  password = "mypasswd"
   expire = 20080523110527Z
```

```
    pwexpire = 20080124075345Z
#   up = true
    up = false
#   policy = "cn=my-password-policy,o=org"
#    tktpolicy = "cn=my-ticket-policy,o=org"
#   set the following to 3 to process all the 3 overrides, set it to
2 for processing first 2 of them, etc.
    noofoverrides = 0
    override-1 = {
         dn = cn=jsmith,ou=engineering,o=acme
         principal-name = jsmith
        password = secret1
        }
    override-2={
         dn = cn=scarl,ou=kerberos,ou=finance,o=acme
         principal-name = scarl
        password = secret2
        }
    override-3={
         dn = cn=john,u=kerberos,ou=engg,o=acme
         principal-name = john
        password = secret3
        }
[operation-2]
#   principal-operation = remove
#   realm = EXAMPLE.COM
#   base = ou=users,o=org
#   scope = sub
#   filter = (cn=*)
#   principal-name = [^.]({cn}){sn}
```

```
[operation-n]
#  principal-operation = add
#  realm = TESTREALM
#  base = ou=engr-unit,o=org
#  scope = base
#  filter = (&(objectClass=inetOrgPerson)(cn=*))
#  principal-name = {sn}
#  password = {RAND}
```

Deployment Notes

7

Use the following information when you deploy the Novell® Kerberos KDC:

7.1 Optimizing the Performance

The LDAP search performance on principal name attributes impacts the authentication performance of the Novell Kerberos KDC.

Before deploying the Novell Kerberos KDC, review and consider the following guidelines to optimize the LDAP search:

- **Create a DS replica indexed on the krbPrincipalName attribute:** Kerberos authentication by the Novell Kerberos KDC searches for the principal name attribute within the specified subtree containers. The search is faster when the container is small and flat. The search time increases as the size and nesting increase.

  To increase the search performance, create separate DS replicas and implement value indexing on the krbPrincipalName attribute. Use this replica server as the LDAP server for KDC, Administration, and Password server access. This indexing on the principal name improves the speed of the search.
- **Create aliases for identities in large trees:** If a large eDirectory™ tree has users with Kerberos identities spread all over the tree, we recommend creating Kerberos alias objects for those eDirectory users and keeping all the Kerberos alias objects under the realm container. This simplifies the search and increases the speed of the Kerberos authentication performance.

7.2 LDAP Server Failover

The Novell Kerberos KDC uses LDAP to access eDirectory. This means that whenever the eDirectory or LDAP services are down or are restarted for maintenance purposes, the Novell Kerberos KDC services are affected. Additionally, the Novell Kerberos KDC services need to be restarted manually whenever the eDirectory or LDAP services are restored.

The Novell Kerberos KDC provides a mechanism to overcome this problem. It establishes LDAP connections with multiple LDAP servers. If any of the servers are not responding, the LDAP connections with the other servers are utilized. If all the LDAP servers are down, the Novell Kerberos KDC services do not abort, but handle the requests appropriately by returning an error. The LDAP module attempts to reconnect with all the LDAP servers until it gets a connection.

The list of LDAP servers can be set in the `/etc/krb5.conf` file.

7.2.1 Configuring Multiple Servers for Failover

The Novell Kerberos KDC services read the database-specific parameters from the `/etc/krb5.conf` configuration file, or you can provide these parameters at the command line. Using the command line helps you avoid frequent modification of the configuration file, and you can modify the options even without Write permissions for the configuration file. Additionally, many server requests with different parameter values on a single machine are also possible.

Setting the LDAP Servers List

You can use any of the following methods to set up the LDAP servers:

The list of the LDAP servers that the Novell Kerberos KDC server tries to connect is defined by the ldap_servers parameter in the `/etc/krb5.conf` file.

- **Configuration File**

  Use the ldap_servers parameter in the `/etc/krb5.conf` file as follows:

  ```
  ldap_servers = ldaps://ldap-server1.mit.edu   ldaps://ldap-server2.mit.edu:1636
  ```

- **Command Line**

  Use the following command line option to set the list of LDAP servers that the Kerberos service (KDC, Administration, and Password) should connect to.

  ```
  -x host=ldaps://hostname:port
  ```

If you have specified multiple LDAP servers, you must configure Kerberos LDAP extensions on all the LDAP servers. The Password Agent needs to be configured only on a single LDAP server, which has a writeable replica.

7.3 Bulkloading Principals

While bulkloading user principals with LDIF files, include `krbPrincipalAux` and `krbTicketPolicyAux` in the objectClass.

For example:

```
version: 1

dn: cn=jsmith,ou=engineering,o=acme
changetype: add
objectclass: User
objectclass: krbPrincipalAux
objectclass: krbTicketPolicyAux
cn: jsmith
Surname: smith
krbPrincipalName: jsmith@ACME.COM
```

Although none of the attributes in the `krbTicketPolicyAux` object class are specified in the LDIF file for the creation of the user principals, failing to include this object class makes the Kerberos administration utilities fail, because they refer to this class.

7.4 Configuring Subtrees for a Realm

If subtrees configured for the realm span across partitions, and if the search scope for the realm is set to subtree, do one of the following:

- Set the referral option to *Enabling Chaining* on the LDAP server configured for Kerberos services.
- Make sure that all the partitions holding the subtree data are present on the LDAP server.

Interoperability with MIT and Microsoft KDCs

8

This section describes how to achieve interoperability with the MIT and the Microsoft KDC cross-realm setup:

The procedures use the following terminology:

- MIT realm: mitrealm
- Win2K domain: w2kdomain
- Novell® Kerberos KDC realm: novlrealm

Replace the realm names specified above with the names chosen by the Kerberos administrator.

8.1 Interoperability with the MIT KDC

8.1.1 Accessing Services in mitrealm from novlrealm

To access services, set up a cross-realm authentication between novlrealm and mitrealm as follows:

1. In novlrealm, create a principal named krbtgt/mitrealm@novlrealm.
2. In mitrealm, create a principal named krbtgt/mitrealm@novlrealm.
3. In the appropriate Kerberos configuration file (`/etc/krb5.conf`), create entries for `novlrealm` and `mitrealm`.

IMPORTANT: Make sure that in both realms the password or key for krbtgt/mitrealm@novlrealm is the same.

8.1.2 Accessing Services in novlrealm from mitrealm

To access services, set up cross-realm authentication between novlrealm and mitrealm:

1. In mitrealm, create a principal named krbtgt/novlrealm@mitrealm.
2. In novlrealm, create a principal named krbtgt/novlrealm@mitrealm.
3. In the appropriate Kerberos configuration file (`/etc/krb5.conf`), create entries for novlrealm and mitrealm.

IMPORTANT: Make sure that in both realms the password or key for krbtgt/novlrealm@mitrealm is same.

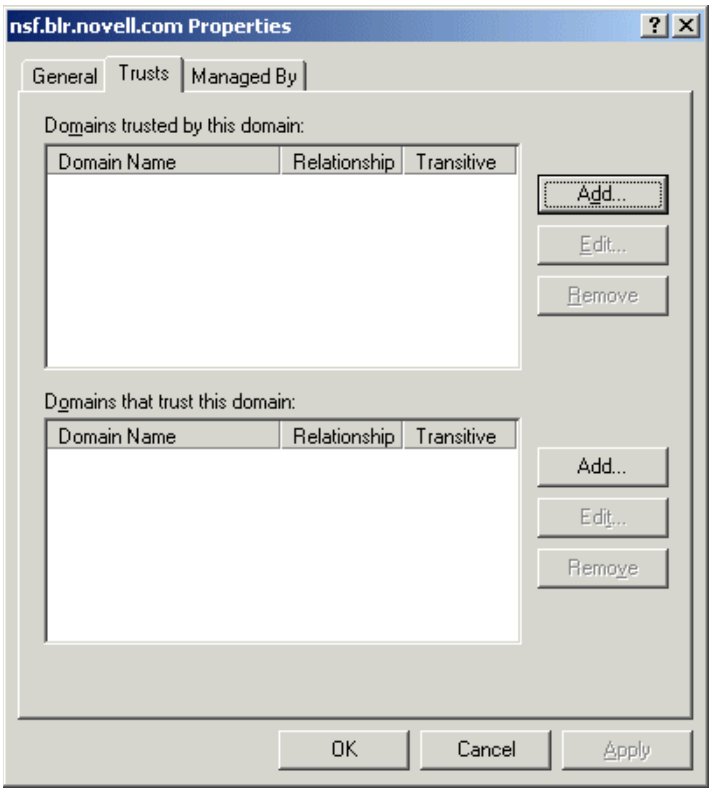
8.2 Interoperability with the Microsoft KDC

To set up cross-realm authentication between novlrealm and w2kdomain:

1. (Conditional) If a user object does not already exist for a user in Active Directory, create a user object.

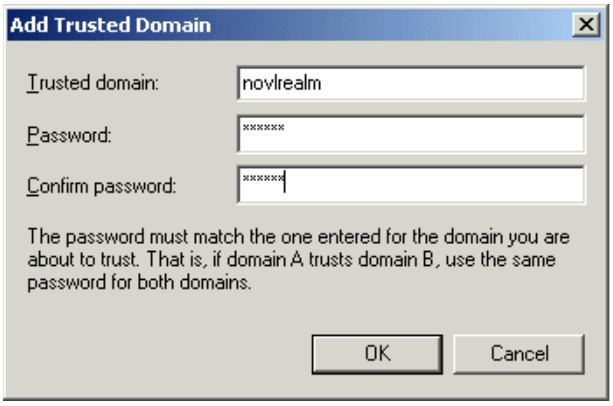
   User creation is required in order to get tickets containing PAC (authorization data honored by application services in w2kdomain) from Microsoft Active Directory or KDC.

2. Map the user's principal in novlrealm to this user object:
   - **2a** Click *Start* > *Programs* > *Administrative Tools* > *Active Directory Users* and *Computers*.
   - **2b** Right-click the user object > *Name Mappings*.
   - **2c** Click *Kerberos Names* > *Add*.
   - **2d** Specify the user's principal name.
3. Set up a trust between w2kdomain and novlrealm:
   - **3a** Click *Start* > *Programs* > *Administrative Tools* > *Active Directory Domains and Trusts*.
   - **3b** Click *win2kdomain* > *Properties* > *Trusts*.
   - **3c** Click *Add* in the *Domains trusted by this domain* section to display the *Add Trusted Domain* dialog box.



3d In the *Add Trusted Domain* dialog box, specify novlrealm as the trusted domain.

Figure 8-1 *Adding Trusted Domain*



3e Enter the password and re-enter it to confirm the password.

> **IMPORTANT:** Make sure that in both realms the password or key of krbtgt/w2kdomain@novlrealm is the same.

3f Click *OK* to ignore the warning message about non-Windows Kerberos realms.

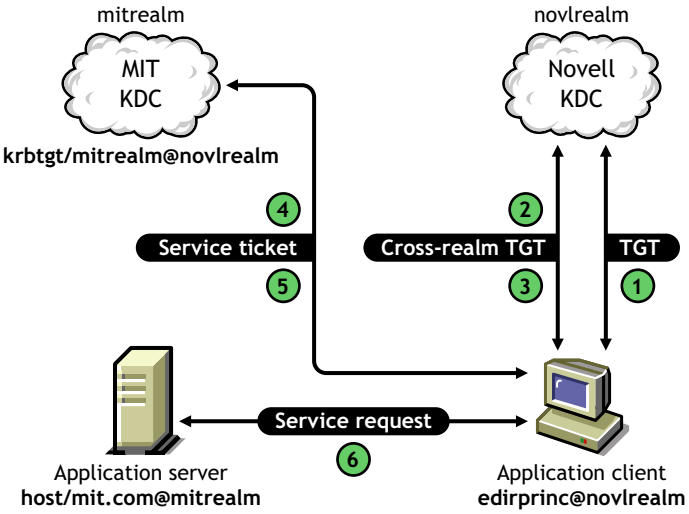
4 In novlrealm, create a principal named krbtgt/w2kdomain@novlrealm.

5 In the appropriate Kerberos configuration file (`/etc/krb5.conf`), create entries for novlrealm and mitrealm.

8.3 How Cross-Realm Setup Works

Figure 5 uses the example of accessing a service in the MIT KDC realm from a Novell KDC realm.

Figure 8-2 *Cross-realm Setup*



The activity listed below uses the following terminology:

eDirectory user:ediruser.novell

User principal:edirprinc@novlrealm
Service principal: host/mit.com@mitrealm

The background activity in a cross-realm setup is explained below:

1. An eDirectory™ user authenticates to novlrealm as edirprinc@novlrealm.
2. The application client requests a service ticket for the principal, host/mit.com@mitrealm, from KDC server hosting novlrealm.
3. The KDC server sends a service ticket for the principal, krbtgt/mitrealm@novlrealm, to the client.
4. The client sends this cross-realm ticket to MIT KDC hosting mitrealm, along with a request for a service ticket for the principal, host/mit.com@mitrealm.
5. MIT KDC sends the service ticket for host/mit.com@mitrealm to the application client.
6. The client sends this service ticket to the application server.

Security Considerations

9

This section provides information on the security considerations for Novell® Kerberos KDC:

- Use SSL mutual authentication or SASL EXTERNAL bind for authenticating the Kerberos services.
- Secure the connection between your Web browser and the iManager server with SSL and the connection between iManager and Novell eDirectory™. Failing to do so causes the Kerberos sensitive data, such as the master key and principal key, to be sniffed during the creation of the realm and principals.
- Protect the following files with appropriate file system rights:
  - Configuration file (`/etc/krb5.conf`)
  - Service password stash file (specified with the ldap_service_password_file parameter in `/etc/krb5.conf`)
  - ACL file for administration (specified with the acl_file parameter in `/etc/krb5.conf`)
  - Password dictionary file (specified with the dict_file parameter in `/etc/krb5.conf`)
  - Certificate files for the Kerberos service.
  - Trusted root certificates for the LDAP servers (specified with the ldap_root_certificate_file parameter in `/etc/krb5.conf`)
  - Log files of the KDC, Administration, and Password servers, because they contain auditing information.
  - Kerberos keytab files (the default location is `/etc/krb5.keytab`)
  - Configuration and log files of the Kerberization utility.
- All of these files must be stored only on the local storage device and not on remotely mounted devices. The recommended file permissions for these files are RW for root. Additionally, protect these files during backup and restore operations.
- Use the strongest cryptographic algorithm for the master and principal keys. Use DES and RC4 only for interoperability with other Kerberos distributions.
- Keep the Kerberos servers in a physically secure location with the access only to the authorized personnel.
- The TGS (krbtgt/REALM@REALM), Administration service (kadmin/admin@REALM), and Password service (kadmin/changepw@REALM) principal keys must be randomly generated and periodically reset.

IMPORTANT: We do not recommend the use of an Administration server, because it needs almost supervisor rights. Instead, we recommend using kadmin.local directly to communicate with eDirectory using LDAP over SSL. We also recommend that you use the Novell Kerberos KDC iManager plug-ins.

Troubleshooting Kerberos KDC

10

This section provides troubleshooting information that you can use to resolve some of the common problems with the Novell® Kerberos KDC.

10.1 Starting the Services

The Server stops functioning, with the error message "File Size Exceeded"

Possible Cause: The log file size cannot exceed 2 GB.

Action: To restart KDC, delete the log file manually. To avoid this situation, we recommend that you back up and delete the log file on a regular basis.

The Server fails to start, with the error message “master key read failed : [-603] - while fetching master key K/M for realm”

Possible Cause: The Service object is not assigned rights to the realm container.

Action: Modify the realm or service by specifying that the corresponding Service object pertains to this realm as follows:

```
kdb5_ldap_util -D cn=admin,o=org modify_service -
realm ATHENA.MIT.EDU cn=service-kdc,o=org
```

For more information, refer to Section 3.4.2, “Modifying a Service,” on page 39.

Server fails to start, with the error message "Invalid credentials"

Possible Cause: Problem in stashing service passwords.

Action: Check to see if the entry corresponding to the service object is correct in the stash file. If there are multiple entries, delete all entries except the last one, then restart the server.

10.2 KDC

KDC fails to start, with the error message "krb5kdc: master key type mismatch"

Possible Cause: The master key type specified with the -k command line option does not match the one in eDirectory™.

Possible Cause: The realm was created with the default value for the master key type but the -k command line option with appropriate type was not specified while invoking the KDC.

Action: Specify the correct master key type with the -k option. If you have created the realm with the default master key type, specify DES3-HMAC-SHA1 as the value for the -k option.

10.3 kdb5_ldap_util

On creating a realm, the error message displayed is create: Realm creation FAILED:[2] Set Master key failed while creating realm 'ATHENA.MIT.EDU'

Possible Cause: The LDAP extension was not added

Action: Using kdb5_ldap_util to add LDAP extension. For example,

```
kdb5_ldap_util -D cn=admin,o=org -w secret
ldapxtn_info -add
```

Possible Cause: The LDAP extension was not loaded

Action: Load the LDAP extension by restarting the LDAP server

create_service or setsrvpw commands fail to set the service object password, with the error message, "FAILED: DSA is unwilling to perform. Failed to set password for service object"

Possible Cause: The password might be violating the password policy configured for the container in the which the service object is created.

Action: If the password is specified manually, ensure that the password adheres to the policy that is configured.

If -randpw option is used, ensure that the password policy allows a password of 128 characters.

destroy: Realm Delete FAILED: Operation not allowed on nonleaf deleting database of ‘ATHENA.MIT.EDU’

Possible Cause: The realm name is case sensitive. Specify the realm name in the same case that was specified during its creation. The destroy realm operation deletes all the principals in the realm before the realm is deleted. If the realm name case specified during the destroy operation is different from that specified during creation, the principals under the realm are not deleted because the principal names includes the realm names, which are case sensitive. If the principals under the realm object are not deleted, the realm destroy operation fails.

Action: Make sure you specify the realm name in the same case as specified during creation.

10.4 kadmin

- “Change password operation fails, with the error message "change_password: Password history principal key version mismatch while changing password for "systest@ATHENA.MIT.EDU"".” on page 89

Change password operation fails, with the error message "change_password: Password history principal key version mismatch while changing password for "systest@ATHENA.MIT.EDU"".

Possible Cause: The history value is greater than 1 in the password policy associated with the principal.

Action: The History option is not supported in this release. Modify the password policy and set the history value to 1.

10.5 Kadmin.local

- “Unable to load requested database module ‘kldap’: plug-in symbol 'kdb_function_table' not found while initializing kadmin.local interface” on page 89

Unable to load requested database module ‘kldap’: plug-in symbol 'kdb_function_table' not found while initializing kadmin.local interface

Possible Cause: Not able to locate the kldap LDAP plug-in.

Action: Add the path where the plug-in is located under [dbmodules] section of the Kerberos krb5.conf configuration file.

```
[dbmodules]
      db_module_dir = /opt/novell/kerberos/lib
```

10.6 iManager Plug-in

- “Failure to extend schema or create realm” on page 89

Failure to extend schema or create realm

Possible Cause: A trusted root certificate was not imported into the keystore.

Action: Import the trusted root certificate from the LDAP server into the keystore and restart the Tomcat server.

Sample krb5.conf File

A

A sample `krb5.conf` file is provided in the `/opt/novell/kerberos/` directory. You can use the `/etc/krb5.conf` configuration file to set the default values. If you do not specify any of the mandatory parameters while managing the Novell® Kerberos KDC, the values are taken from the `/etc/krb5.conf` file. This file looks similar to the following:

```
[libdefaults]
      default_realm = ATHENA.MIT.EDU

[realms]
        ATHENA.MIT.EDU = {
                max_life = 10h 0m 0s
                max_renewable_life = 7d 0h 0m 0s
                acl_file = /opt/novell/kerberos/kadm5.acl
                dict_file = /opt/novell/kerberos/kadm5.dict
                kdc = kerberos.mit.edu
                admin_server = kerberos-1.mit.edu
                kpasswd_server = kerberos-1.mit.edu
                database_module = ldapconf
        }

[domain_realm]
      .mit.edu = ATHENA.MIT.EDU
      mit.edu = ATHENA.MIT.EDU

[logging]
      kdc = FILE:/var/log/krb5kdc.log
      admin_server = FILE:/var/log/kadmind.log
      kpasswd_server = FILE:/var/log/kpasswdd.log

[dbdefaults]
      database_module = ldapconf

[dbmodules]
      db_module_dir = /opt/novell/kerberos/lib
      ldapconf = {
                db_library = kldap
                ldap_kdc_dn = "cn=KDC Server - kerberos.mit.edu,o=mit"
                ldap_kadmind_dn = "cn=Admin Server -
kerberos.mit.edu,o=mit"
                ldap_kpasswdd_dn = "cn=Passwd Server -
kerberos.mit.edu,o=mit"
                ldap_root_certificate_file = /opt/novell/kerberos/
TrustedRoot-ldap-server1.mit.edu.der
                ldap_service_password_file = /opt/novell/kerberos/
keyfile
                ldap_servers = ldaps://dap-server1.mit.edu  ldaps://
ldap-server2.mit.edu:1636
                ldap_conns_per_server = 5
        }
```

Supported Encryption Types and Salt Types

B

This section lists the supported encryption and salt types supported in the Novell® Kerberos KDC.

B.1 Supported Encryption Types

The following encryption types are supported by the Novell Kerberos KDC components:

- des-cbc-crc: DES cbc mode with CRC-32
- des-cbc-md4: DES cbc mode with RSA-MD4
- des-cbc-md5: DES cbc mode with RSA-MD5
- des3-cbc-sha1
- des3-hmac-sha1
- des3-cbc-sha1-kd: triple DES cbc mode with HMAC/sha1
- aes256-cts-hmac-sha1-96
- aes256-cts: AES-256 CTS mode with 96-bit SHA-1 HMAC
- aes128-cts-hmac-sha1-96
- aes128-cts: AES-128 CTS mode with 96-bit SHA-1 HMAC
- arcfour-hmac
- rc4-hmac
- arcfour-hmac-md5: RC4 with HMAC/MD5

B.2 Supported Salt Types

Your Kerberos key is derived from your password. To ensure that users who happen to have the same password do not have the same key, Kerberos 5 incorporates more information into the key using something called a salt. The supported values for salts are as follows.

- normal: The default for Kerberos Version 5
- v4: The only type used by Kerberos Version 4, no salt
- norealm: The same as the default, without using realm information
- onlyrealm: Uses only realm information as the salt
- special: Only used in very special cases; not fully supported

Administrative Privileges for the Kerberos Database

C

You need to set administrative privileges for the Kerberos database in the `kadm5.acl` file.

The format of the file is as follows:

```
Kerberos_principal  permissions  [target_principal][restrictions]
```

Table C-1 *kadm5.acl File*

Field	Description	
Kerberos_principal	The Kerberos principal (and optional target principal) can include the "*" wildcard, so if you want any principal with the instance "admin" to have full permissions on the database, you could use the principal "*/admin@REALM" where "REALM" is your Kerberos realm.	
	NOTE: A common use of an admin instance is to grant separate permissions (such as administrator access to the Kerberos database) to a separate Kerberos principal. For example, the user *joeadmin* might have a principal for his administrative use, called *joeadmin/admin*. This way, *joeadmin* would obtain *joeadmin/admin* tickets only when he actually needs to use those permissions.	
target_principal	Can also include backreferences to Kerberos_principal, in which "*number" matches the component number in the Kerberos_principal.	
permissions	The permissions are represented by single letters; UPPERCASE letters represent negative permissions. The permissions are:	
	a	Allows the addition of principals or policies in the database.
	A	Disallows the addition of principals or policies in the database.
	d	Allows the deletion of principals or policies in the database.
	D	Disallows the deletion of principals or policies in the database.
	m	Allows the modification of principals or policies in the database.
	M	Disallows the modification of principals or policies in the database.
	c	Allows the changing of passwords for principals in the database.
	C	Disallows the changing of passwords for principals in the database.
	i	Allows inquiries to the database.
	I	Disallows inquiries to the database.
	l	Allows the listing of principals or policies in the database.
	L	Disallows the listing of principals or policies in the database.
	s	Allows the explicit setting of the key for a principal.
	S	Disallows the explicit setting of the key for a principal.

Field	Description	
	*	All privileges (admcil).
	X	All privileges (admcil); identical to "*".
restrictions	The restrictions are a string of flags. Allowed restrictions are:	
	[+ -]flagname	The flag is forced to the indicated value. The permissible flags are the same as the + and - flags for the kadmin addprinc and modprinc commands.
	-clearpolicy	The policy is forced to clear.
	-policy pol	The policy is forced to be pol.
	expire time	
	pwexpire time	
	maxlife time	
	maxrenewlife time	The associated value is forced to MIN (time, requested value).

The above flags act as restrictions on any add or modify operation that is allowed because of that ACL line.

An example of a `kadm5.acl` file is as follows:

NOTE: Order is important; permissions are determined by the first matching entry.

```
*/admin@ATHENA.MIT.EDU  *
joeadmin@ATHENA.MIT.EDU  ADMCIL
joeadmin/*@ATHENA.MIT.EDU il */root@ATHENA.MIT.EDU
*@ATHENA.MIT.EDU cil *1/admin@ATHENA.MIT.EDU
*/*@ATHENA.MIT.EDU  i
*/admin@EXAMPLE.COM * -maxlife 9h -postdateable
```

In the above file, any principal in the ATHENA.MIT.EDU realm with an admin instance has all administrative privileges.

The user joeadmin has all permissions with his admin instance, joeadmin/admin@ATHENA.MIT.EDU (matches the first line).

He has no permissions at all with his null instance, joeadmin@ATHENA.MIT.EDU (matches the second line). His root instance has inquire and list permissions with any other principal that has the instance root.

Any principal in ATHENA.MIT.EDU can inquire, list, or change the password of his or her admin instance, but not any other admin instance. Any principal in the realm ATHENA.MIT.EDU (except for joeadmin@ATHENA.MIT.EDU, as mentioned above) has inquire privileges.

Finally, any principal with an admin instance in EXAMPLE.COM has all permissions, but any principal that he or she creates or modifies cannot get postdateable tickets or tickets with a life of longer than 9 hours.