

Kerberos V5 Installation Guide

Release: 1.10

Document Edition: 1.1

Last updated: 6 November 2013

1 Introduction

1.1 What is Kerberos and How Does it Work?

Kerberos V5 is based on the Kerberos authentication system developed at MIT. Under Kerberos, a client (generally either a user or a service) sends a request for a ticket to the Key Distribution Center (KDC). The KDC creates a *ticket-granting ticket* (TGT) for the client, encrypts it using the client's password as the key, and sends the encrypted TGT back to the client. The client then attempts to decrypt the TGT, using its password. If the client successfully decrypts the TGT (*i.e.*, if the client gave the correct password), it keeps the decrypted TGT, which indicates proof of the client's identity.

The TGT, which expires at a specified time, permits the client to obtain additional tickets, which give permission for specific services. The requesting and granting of these additional tickets is user-transparent.

1.2 Why Should I use Kerberos?

Since Kerberos negotiates authenticated, and optionally encrypted, communications between two points anywhere on the Internet, it provides a layer of security that is not dependent on which side of a firewall either client is on. Since studies have shown that half of the computer security breaches in industry happen from *inside* firewalls, Kerberos V5 from MIT will play a vital role in the security of your network.

1.3 Please Read the Documentation

As with any software package that uses a centralized database, the installation procedure is somewhat involved, and requires forethought and planning. MIT has attempted to make this Kerberos V5 Installation Guide as concise as possible, rather than making it an exhaustive description of the details of Kerberos. Consequently, everything in this guide appears because MIT believes that it is important. Please read and follow these instructions carefully.

This document is one piece of the document set for Kerberos V5. The documents, and their intended audiences, are:

- **Kerberos V5 Installation Guide:** a concise guide for installing Kerberos V5. Kerberos administrators (particularly whoever will be making site-wide decisions about the installation) and the system administrators who will be installing the software should read this guide.
- **Kerberos V5 System Administrator's Guide:** a sysadmin's guide to administering a Kerberos installation. The System Administrator's Guide describes the administration software and suggests policies and procedures for administering a Kerberos installation. Anyone who will have administrative access to your Kerberos database should read this guide.
- **Kerberos V5 UNIX User's Guide:** a guide to using the Kerberos UNIX client programs. All users on UNIX systems should read this guide, particularly the "Tutorial" section.

1.4 Overview of This Guide

The next chapter describes the decisions you need to make before installing Kerberos V5.

Chapter three provided instructions for building the Kerberos sources.

Chapter four describes installation procedures for each class of Kerberos machines:

1. Key Distribution Centers (KDCs).
 - A. The Master KDC.
 - B. Slave KDCs.
2. UNIX client machines
3. UNIX application server machines

Note that a machine can be both a client machine and an application server.

Chapter five describes procedure for updating previous installations of Kerberos V5.

Chapter six describes our problem reporting system.

2 Realm Configuration Decisions

Before installing Kerberos V5, it is necessary to consider the following issues:

- The name of your Kerberos realm (or the name of each realm, if you need more than one).
- How you will map your hostnames onto Kerberos realms.
- Which ports your KDC and kadmin (database access) services will use.
- How many slave KDCs you need and where they should be located.
- The hostnames of your master and slave KDCs.
- How frequently you will propagate the database from the master KDC to the slave KDCs.

2.1 Kerberos Realms

Although your Kerberos realm can be any ASCII string, convention is to make it the same as your domain name, in upper-case letters. For example, hosts in the domain example.com would be in the Kerberos realm EXAMPLE.COM.

If you need multiple Kerberos realms, MIT recommends that you use descriptive names which end with your domain name, such as BOSTON.EXAMPLE.COM and HOUSTON.EXAMPLE.COM.

2.2 Mapping Hostnames onto Kerberos Realms

Mapping hostnames onto Kerberos realms is done in one of two ways.

The first mechanism, which has been in use for years in MIT-based Kerberos distributions, works through a set of rules in the `krb5.conf` configuration file. (See Section 4.1.1.2 [krb5.conf], page 21.) You can specify mappings for an entire domain or subdomain, and/or on a hostname-by-hostname basis. Since greater specificity takes precedence, you would do this by specifying the mappings for a given domain or subdomain and listing the exceptions.

The second mechanism works by looking up the information in special TXT records in the Domain Name Service. This is currently not used by default because security holes could result if the DNS TXT records were spoofed. If this mechanism is enabled on the client, it will try to look up a TXT record for the DNS name formed by putting the prefix `_kerberos` in front of the hostname in question. If that record is not found, it will try using `_kerberos` and the host's domain name, then its parent domain, and so forth. So for the hostname BOSTON.ENGINEERING.FOOBAR.COM, the names looked up would be:

```
_kerberos.boston.engineering.foobar.com  
_kerberos.engineering.foobar.com  
_kerberos.foobar.com  
_kerberos.com
```

The value of the first TXT record found is taken as the realm name. (Obviously, this doesn't work all that well if a host and a subdomain have the same name, and different realms. For example, if all the hosts in the ENGINEERING.FOOBAR.COM domain are in the ENGINEERING.FOOBAR.COM realm, but a host named ENGINEERING.FOOBAR.COM is

for some reason in another realm. In that case, you would set up TXT records for all hosts, rather than relying on the fallback to the domain name.)

Even if you do not choose to use this mechanism within your site, you may wish to set it up anyway, for use when interacting with other sites.

2.3 Ports for the KDC and Admin Services

The default ports used by Kerberos are port 88 for the KDC¹ and port 749 for the admin server. You can, however, choose to run on other ports, as long as they are specified in each host's `/etc/services` and `krb5.conf` files, and the `kdc.conf` file on each KDC. For a more thorough treatment of port numbers used by the Kerberos V5 programs, refer to the “Configuring Your Firewall to Work With Kerberos V5” section of the *Kerberos V5 System Administrator's Guide*.

2.4 Slave KDCs

Slave KDCs provide an additional source of Kerberos ticket-granting services in the event of inaccessibility of the master KDC. The number of slave KDCs you need and the decision of where to place them, both physically and logically, depends on the specifics of your network.

All of the Kerberos authentication on your network requires that each client be able to contact a KDC. Therefore, you need to anticipate any likely reason a KDC might be unavailable and have a slave KDC to take up the slack.

Some considerations include:

- Have at least one slave KDC as a backup, for when the master KDC is down, is being upgraded, or is otherwise unavailable.
- If your network is split such that a network outage is likely to cause a network partition (some segment or segments of the network to become cut off or isolated from other segments), have a slave KDC accessible to each segment.
- If possible, have at least one slave KDC in a different building from the master, in case of power outages, fires, or other localized disasters.

2.5 Hostnames for the Master and Slave KDCs

MIT recommends that your KDCs have a predefined set of CNAME records (DNS hostname aliases), such as `kerberos` for the master KDC and `kerberos-1`, `kerberos-2`, ... for the slave KDCs. This way, if you need to swap a machine, you only need to change a DNS entry, rather than having to change hostnames.

A new mechanism for locating KDCs of a realm through DNS has been added to the MIT Kerberos V5 distribution. A relatively new record type called `SRV` has been added to DNS. Looked up by a service name and a domain name, these records indicate the hostname and port number to contact for that service, optionally with weighting and prioritization. (See RFC 2782 if you want more information. You can follow the example below for straightforward cases.)

¹ Kerberos V4 used port 750. If necessary, you can run on both ports for backward compatibility.

The use with Kerberos is fairly straightforward. The domain name used in the SRV record name is the domain-style Kerberos realm name. (It is possible to have Kerberos realm names that are not DNS-style names, but we don't recommend it for Internet use, and our code does not support it well.) Several different Kerberos-related service names are used:

`_kerberos._udp`

This is for contacting any KDC by UDP. This entry will be used the most often. Normally you should list port 88 on each of your KDCs.

`_kerberos._tcp`

This is for contacting any KDC by TCP. The MIT KDC by default will not listen on any TCP ports, so unless you've changed the configuration or you're running another KDC implementation, you should leave this unspecified. If you do enable TCP support, normally you should use port 88.

`_kerberos-master._udp`

This entry should refer to those KDCs, if any, that will immediately see password changes to the Kerberos database. This entry is used only in one case, when the user is logging in and the password appears to be incorrect; the master KDC is then contacted, and the same password used to try to decrypt the response, in case the user's password had recently been changed and the first KDC contacted hadn't been updated. Only if that fails is an "incorrect password" error given.

If you have only one KDC, or for whatever reason there is no accessible KDC that would get database changes faster than the others, you do not need to define this entry.

`_kerberos-adm._tcp`

This should list port 749 on your master KDC. Support for it is not complete at this time, but it will eventually be used by the `kadmin` program and related utilities. For now, you will also need the `admin_server` entry in `krb5.conf`. (See Section 4.1.1.2 [krb5.conf], page 21.)

`_kpasswd._udp`

This should list port 464 on your master KDC. It is used when a user changes her password.

Be aware, however, that the DNS SRV specification requires that the hostnames listed be the canonical names, not aliases. So, for example, you might include the following records in your (BIND-style) zone file:

```
$ORIGIN foobar.com.
_kerberos          TXT      "FOOBAR.COM"
kerberos           CNAME    daisy
kerberos-1         CNAME    use-the-force-luke
kerberos-2         CNAME    bunny-rabbit
_kerberos._udp     SRV      0 0 88 daisy
                   SRV      0 0 88 use-the-force-luke
                   SRV      0 0 88 bunny-rabbit
_kerberos-master._udp SRV    0 0 88 daisy
_kerberos-adm._tcp SRV      0 0 749 daisy
_kpasswd._udp      SRV      0 0 464 daisy
```

As with the DNS-based mechanism for determining the Kerberos realm of a host, we recommend distributing the information this way for use by other sites that may want to interact with yours using Kerberos, even if you don't immediately make use of it within your own site. If you anticipate installing a very large number of machines on which it will be hard to update the Kerberos configuration files, you may wish to do all of your Kerberos service lookups via DNS and not put the information (except for `admin_server` as noted above) in future versions of your `krb5.conf` files at all. Eventually, we hope to phase out the listing of server hostnames in the client-side configuration files; making preparations now will make the transition easier in the future.

2.6 Database Propagation

The Kerberos database resides on the master KDC, and must be propagated regularly (usually by a cron job) to the slave KDCs. In deciding how frequently the propagation should happen, you will need to balance the amount of time the propagation takes against the maximum reasonable amount of time a user should have to wait for a password change to take effect.

If the propagation time is longer than this maximum reasonable time (*e.g.*, you have a particularly large database, you have a lot of slaves, or you experience frequent network delays), you may wish to cut down on your propagation delay by performing the propagation in parallel. To do this, have the master KDC propagate the database to one set of slaves, and then have each of these slaves propagate the database to additional slaves.

3 Building Kerberos V5

Kerberos V5 uses a configuration system built using the Free Software Foundation's 'autoconf' program. This system makes Kerberos V5 much simpler to build and reduces the amount of effort required in porting Kerberos V5 to a new platform.

3.1 Organization of the Source Directory

Below is a brief overview of the organization of the complete source directory. More detailed descriptions follow.

appl	applications with Kerberos V5 extensions
clients	Kerberos V5 user programs
gen-manpages	manpages for Kerberos V5 and the Kerberos V5 login program
include	include files
kadmin	administrative interface to the Kerberos master database
kdc	the Kerberos V5 Authentication Service and Key Distribution Center
krb524	utilities for converting between Kerberos 4 and Kerberos 5
lib	libraries for use with/by Kerberos V5
mac	source code for building Kerberos V5 on MacOS
prototype	templates for source code files
slave	utilities for propagating the database to slave KDCs
tests	test suite
util	various utilities for building/configuring the code, sending bug reports, etc.
windows	source code for building Kerberos V5 on Windows (see windows/README)

3.1.1 The appl Directory

The *appl* directory contains sample Kerberos application client and server programs. In previous releases, it contained Kerberized versions of remote access daemons, but those have now been moved to a separate project.

3.1.2 The clients Directory

This directory contains the code for several user-oriented programs.

kdestroy	This program destroys the user's active Kerberos authorization tickets. MIT recommends that users kdestroy before logging out.
kinit	This program prompts users for their Kerberos principal name and password, and attempts to get an initial ticket-granting-ticket for that principal.
klist	This program lists the Kerberos principal and Kerberos tickets held in a credentials cache, or the keys held in a keytab file.
kpasswd	This program changes a user's Kerberos password.
ksu	This program is a Kerberized version of the su program that is meant to securely change the real and effective user ID to that of the target user and to create a new security context.
kvno	This program acquires a service ticket for the specified Kerberos principals and prints out the key version numbers of each.

3.1.3 The gen-manpages Directory

There are two manual pages in this directory. One is an introduction to the Kerberos system. The other describes the `.k5login` file which allows users to give access with their UID to other users authenticated by the Kerberos system.

3.1.4 The include Directory

This directory contains the *include* files needed to build the Kerberos system.

3.1.5 The kadmin Directory

In this directory is the code for the utilities `kadmin`, `kadmin.local`, `kdb5_util`, and `ktutil`. `ktutil` is the Kerberos keytab file maintenance utility from which a Kerberos administrator can read, write, or edit entries in a Kerberos V5 keytab or Kerberos V4 srvtab. `kadmin` and `kadmin.local` are command-line interfaces to the Kerberos V5 KADM5 administration system. `kadmin.local` runs on the master KDC and does not use Kerberos to authenticate to the database, while `kadmin` uses Kerberos authentication and an encrypted RPC. The two provide identical functionalities, which allow administrators to modify the database of Kerberos principals. `kdb5_util` allows administrators to perform low-level maintenance procedures on Kerberos and the KADM5 database. With this utility, databases can be created, destroyed, or dumped to and loaded from ASCII files. It can also be used to create master key stash files.

3.1.6 The kdc Directory

This directory contains the code for the `krb5kdc` daemon, the Kerberos Authentication Service and Key Distribution Center.

3.1.7 The krb524 Directory

This directory contains the code for `krb524`, a service that converts Kerberos V5 credentials into Kerberos V4 credentials suitable for use with applications that for whatever reason do not use V5 directly.

3.1.8 The lib Directory

The *lib* directory contain 10 subdirectories as well as some definition and glue files. The *crypto* subdirectory contains the Kerberos V5 encryption library. The *des425* subdirectory exports the Kerberos V4 encryption API, and translates these functions into calls to the Kerberos V5 encryption API. The *gssapi* library contains the Generic Security Services API, which is a library of commands to be used in secure client-server communication. The *kadm5* directory contains the libraries for the KADM5 administration utilities. The Kerberos 5 database libraries are contained in *kdb*. The directories *krb4* and *krb5* contain the Kerberos 4 and Kerberos 5 APIs, respectively. The *rpc* directory contains the API for the Kerberos Remote Procedure Call protocol.

3.1.9 The prototype Directory

This directory contains several template files. The `prototype.h` and `prototype.c` files contain the MIT copyright message and a placeholder for the title and description of the file. `prototype.h` also has a short template for writing `ifdef` and `ifndef` preprocessor

statements. The `getopt.c` file provides a template for writing code that will parse the options with which a program was called.

3.1.10 The slave Directory

This directory contains code which allows for the propagation of the Kerberos principal database from the master KDC to slave KDCs over an encrypted, secure channel. `kprop` is the program which actually propagates the database dump file. `kpropd` is the Kerberos V5 slave KDC update server which accepts connections from the `kprop` program. `kslave_update` is a script that takes the name of a slave server, and propagates the database to that server if the database has been modified since the last dump or if the database has been dumped since the last propagation.

3.1.11 The util Directory

This directory contains several utility programs and libraries. The programs used to configure and build the code, such as `autoconf`, `lndir`, `kbuild`, `reconf`, and `makedepend`, are in this directory. The *profile* directory contains most of the functions which parse the Kerberos configuration files (`krb5.conf` and `kdc.conf`). Also in this directory are the Kerberos error table library and utilities (*et*), the Sub-system library and utilities (*ss*), database utilities (*db2*), pseudo-terminal utilities (*pty*), bug-reporting program `send-pr`, and a generic support library `support` used by several of our other libraries.

3.2 Build Requirements

In order to build Kerberos V5, you will need approximately 60-70 megabytes of disk space. The exact amount will vary depending on the platform and whether the distribution is compiled with debugging symbol tables or not.

Your C compiler must conform to ANSI C (ISO/IEC 9899:1990, “c89”). Some operating systems do not have an ANSI C compiler, or their default compiler requires extra command-line options to enable ANSI C conformance.

If you wish to keep a separate *build tree*, which contains the compiled ‘*.o’ file and executables, separate from your source tree, you will need a ‘make’ program which supports ‘VPATH’, or you will need to use a tool such as ‘lndir’ to produce a symbolic link tree for your build tree.

3.3 Unpacking the Sources

The first step in each of these build procedures is to unpack the source distribution. The Kerberos V5 distribution comes in a tar file, generally named ‘`krb5-1.10.tar`’, which contains a compressed tar file consisting of the sources for all of Kerberos (generally ‘`krb5-1.10.tar.gz`’) and a PGP signature for this source tree (generally ‘`krb5-1.10.tar.gz.asc`’). MIT highly recommends that you verify the integrity of the source code using this signature.

Unpack the compressed tar file in some directory, such as ‘`/u1/krb5-1.10`’. (In the rest of this document, we will assume that you have chosen to unpack the Kerberos V5 source distribution in this directory. Note that the tarfiles will by default all unpack into the

‘./krb5-1.10’ directory, so that if your current directory is ‘/u1’ when you unpack the tarfiles, you will get ‘/u1/krb5-1.10/src’, etc.)

3.4 Doing the Build

You have a number of different options in how to build Kerberos. If you only need to build Kerberos for one platform, using a single directory tree which contains both the source files and the object files is the simplest. However, if you need to maintain Kerberos for a large number of platforms, you will probably want to use separate build trees for each platform. We recommend that you look at Section 3.10 [OS Incompatibilities], page 16, for notes that we have on particular operating systems.

3.4.1 Building Within a Single Tree

If you don’t want separate build trees for each architecture, then use the following abbreviated procedure.

1. `cd /u1/krb5-1.10/src`
2. `./configure`
3. `make`

That’s it!

3.4.2 Building with Separate Build Directories

If you wish to keep separate build directories for each platform, you can do so using the following procedure. (Note, this requires that your ‘make’ program support ‘VPATH’. GNU’s make will provide this functionality, for example.) If your ‘make’ program does not support this, see the next section.

For example, if you wish to create a build directory for pmax binaries you might use the following procedure:

1. `mkdir /u1/krb5-1.10/pmax`
2. `cd /u1/krb5-1.10/pmax`
3. `../src/configure`
4. `make`

3.4.3 Building Using ‘lnidir’

If you wish to keep separate build directories for each platform, and you do not have access to a ‘make’ program which supports ‘VPATH’, all is not lost. You can use the ‘lnidir’ program to create symbolic link trees in your build directory.

For example, if you wish to create a build directory for solaris binaries you might use the following procedure:

1. `mkdir /u1/krb5-1.10/solaris`
2. `cd /u1/krb5-1.10/solaris`
3. `/u1/krb5-1.10/src/util/lnidir ‘pwd’/../../src`
4. `./configure`

5. `make`

You must give an absolute pathname to `'lndir'` because it has a bug that makes it fail for relative pathnames. Note that this version differs from the latest version as distributed and installed by the XConsortium with X11R6. Either version should be acceptable.

3.5 Installing the Binaries

Once you have built Kerberos, you should install the binaries. You can do this by running:

```
% make install
```

If you want to install the binaries into a destination directory that is not their final destination, which may be convenient if you want to build a binary distribution to be deployed on multiple hosts, you may use:

```
% make install DESTDIR=/path/to/destdir
```

This will install the binaries under `DESTDIR/PREFIX`, e.g., the user programs will install into `DESTDIR/PREFIX/bin`, the libraries into `DESTDIR/PREFIX/lib`, etc.

Note that if you want to test the build (see Section 3.6 [Testing the Build], page 11), you usually do not need to do a `make install` first.

Some implementations of `'make'` allow multiple commands to be run in parallel, for faster builds. We test our Makefiles in parallel builds with GNU `'make'` only; they may not be compatible with other parallel build implementations.

3.6 Testing the Build

The Kerberos V5 distribution comes with built-in regression tests. To run them, simply type the following command while in the top-level build directory (i.e., the directory where you sent typed `'make'` to start building Kerberos; see Section 3.4 [Doing the Build], page 10.):

```
% make check
```

However, there are several prerequisites that must be satisfied first:

- Configure and build Kerberos with Tcl support. Tcl is used to drive the test suite. This often means passing `--with-tcl` to configure to tell it the location of the Tcl configuration script. (See Section 3.7 [Options to Configure], page 12.)
- On some operating systems, you have to run `'make install'` before running `'make check'`, or the test suite will pick up installed versions of Kerberos libraries rather than the newly built ones. You can install into a prefix that isn't in the system library search path, though. Alternatively, you can configure with `--disable-rpath`, which renders the build tree less suitable for installation, but allows testing without interference from previously installed libraries.
- In order to test the RPC layer, the local system has to be running the `portmap` daemon and it has to be listening to the regular network interface (not just localhost).

3.6.1 The DejaGnu Tests

Some of the built-in regression tests are setup to use the DejaGnu framework for running tests. These tests tend to be more comprehensive than the normal built-in tests as they setup test servers and test client/server activities.

DejaGnu may be found wherever GNU software is archived.

3.6.2 The KADM5 Tests

Regression tests for the KADM5 system, including the GSS-RPC, KADM5 client and server libraries, and kpasswd, are also included in this release. Each set of KADM5 tests is contained in a sub-directory called **unit-test** directly below the system being tested. For example, `lib/rpc/unit-test` contains the tests for GSS-RPC. The tests are all based on DejaGnu (but they are not actually called part of "The DejaGnu tests," whose naming predates the inclusion of the KADM5 system). In addition, they require the Tool Command Language (TCL) header files and libraries to be available during compilation and some of the tests also require Perl in order to operate. If all of these resources are not available during configuration, the KADM5 tests will not run. The TCL installation directory can be specified with the `--with-tcl` configure option. (See See Section 3.7 [Options to Configure], page 12.) The `runtest` and `perl` programs must be in the current execution path.

If you install DejaGnu, TCL, or Perl after configuring and building Kerberos and then want to run the KADM5 tests, you will need to re-configure the tree and run **make** at the top level again to make sure all the proper programs are built. To save time, you actually only need to reconfigure and build in the directories `src/kadmin/testing`, `src/lib/rpc`, `src/lib/kadm5`.

3.7 Options to Configure

There are a number of options to '**configure**' which you can use to control how the Kerberos distribution is built. The following table lists the most commonly used options to Kerberos V5's '**configure**' program.

--help

Provides help to configure. This will list the set of commonly used options for building Kerberos.

--prefix=PREFIX

By default, Kerberos will install the package's files rooted at `"/usr/local"` as in `"/usr/local/bin"`, `"/usr/local/sbin"`, etc. If you desire a different location, use this option.

--exec-prefix=EXECPREFIX

This option allows one to separate the architecture independent programs from the configuration files and manual pages.

--localstatedir=LOCALSTATEDIR

This option sets the directory for locally modifiable single-machine data. In Kerberos, this mostly is useful for setting a location for the KDC data files, as they will be installed in `LOCALSTATEDIR/krb5kdc`, which is by default `PREFIX/var/krb5kdc`.

CC=COMPILER

Use **COMPILER** as the C compiler.

CFLAGS=FLAGS

Use **FLAGS** as the default set of C compiler flags.

Note that if you use the native Ultrix compiler on a DECstation you are likely to lose if you pass no flags to cc; md4.c takes an estimated 3,469 billion years to compile if you provide neither the `-g` flag nor the `-O` flag to `cc`.

CPPFLAGS=CPPOPTS

Use **CPPOPTS** as the default set of C preprocessor flags. The most common use of this option is to select certain `#define`'s for use with the operating system's include files.

LD=LINKER

Use **LINKER** as the default loader if it should be different from C compiler as specified above.

LDFLAGS=LDOPTS

This option allows one to specify optional arguments to be passed to the linker. This might be used to specify optional library paths.

--with-krb4

This option enables Kerberos V4 backwards compatibility using the builtin Kerberos V4 library.

--with-krb4=KRB4DIR

This option enables Kerberos V4 backwards compatibility using a pre-existing Kerberos V4 installation. The directory specified by **KRB4DIR** specifies where the V4 header files should be found (`'KRB4DIR/include'`) as well as where the V4 Kerberos library should be found (`'KRB4DIR/lib'`).

--without-krb4

Disables Kerberos V4 backwards compatibility. This prevents Kerberos V4 clients from using the V5 services including the KDC. This would be useful if you know you will never install or need to interact with V4 clients.

--with-netlib[=libs]

Allows for suppression of or replacement of network libraries. By default, Kerberos V5 configuration will look for `-lnsl` and `-lsocket`. If your operating system has a broken resolver library (see Section 3.10.7 [Solaris versions 2.0 through 2.3], page 18) or fails to pass the tests in `'src/tests/resolv'` you will need to use this option.

--with-tcl=TCLPATH

Some of the unit-tests in the build tree rely upon using a program in Tcl. The directory specified by **TCLPATH** specifies where the Tcl header file (`'TCLPATH/include/tcl.h'`) as well as where the Tcl library should be found (`'TCLPATH/lib'`).

--enable-shared

This option will turn on the building and use of shared library objects in the Kerberos build. This option is only supported on certain platforms.

`--enable-dns`

`--enable-dns-for-kdc`

`--enable-dns-for-realm`

Enable the use of DNS to look up a host's Kerberos realm, or a realm's KDCs, if the information is not provided in `krb5.conf`. See Section 2.5 [Hostnames for the Master and Slave KDCs], page 4 for information about using DNS to locate the KDCs, and Section 2.2 [Mapping Hostnames onto Kerberos Realms], page 3 for information about using DNS to determine the default realm. By default, DNS lookups are enabled for the former but not for the latter.

`--disable-kdc-lookaside-cache`

Disables the cache in the KDC which detects retransmitted client requests and resends the previous responses to them.

`--with-system-et`

Use an installed version of the error-table support software, the `'compile_et'` program, the `'com_err.h'` header file and the `'com_err'` library. If these are not in the default locations, you may wish to specify `CPPFLAGS=-I/some/dir` and `LDLFLAGS=-L/some/other/dir` options at configuration time as well.

If this option is not given, a version supplied with the Kerberos sources will be built and installed along with the rest of the Kerberos tree, for Kerberos applications to link against.

`--with-system-ss`

Use an installed version of the subsystem command-line interface software, the `'mk_cmds'` program, the `'ss/ss.h'` header file and the `'ss'` library. If these are not in the default locations, you may wish to specify `CPPFLAGS=-I/some/dir` and `LDLFLAGS=-L/some/other/dir` options at configuration time as well. See also the `'SS_LIB'` option.

If this option is not given, the `'ss'` library supplied with the Kerberos sources will be compiled and linked into those programs that need it; it will not be installed separately.

`SS_LIB=libs...`

If `'-lss'` is not the correct way to link in your installed `'ss'` library, for example if additional support libraries are needed, specify the correct link options here. Some variants of this library are around which allow for Emacs-like line editing, but different versions require different support libraries to be explicitly specified.

This option is ignored if `'--with-system-ss'` is not specified.

`--with-system-db`

Use an installed version of the Berkeley DB package, which must provide an API compatible with version 1.85. This option is *unsupported* and untested. In particular, we do not know if the database-rename code used in the dumpfile load operation will behave properly.

If this option is not given, a version supplied with the Kerberos sources will be built and installed. (We are not updating this version at this time because of licensing issues with newer versions that we haven't investigated sufficiently yet.)

DB_HEADER=headername.h

If 'db.h' is not the correct header file to include to compile against the Berkeley DB 1.85 API, specify the correct header file name with this option. For example, 'DB_HEADER=db3/db_185.h'.

DB_LIB=libs...

If '-ldb' is not the correct library specification for the Berkeley DB library version to be used, override it with this option. For example, 'DB_LIB=-ldb-3.3'.

--with-crypto-impl=IMPL

Use specified crypto implementation in lieu of the default builtin. Currently only one alternative crypto-system openssl is available and it requires version 1.0.0 or higher of OpenSSL.

For example, in order to configure Kerberos on a Solaris machine using the 'suncc' compiler with the optimizer turned on, run the configure script with the following options:

```
% ./configure CC=suncc CFLAGS=-O
```

For a slightly more complicated example, consider a system where several packages to be used by Kerberos are installed in '/usr/foobar', including Berkeley DB 3.3, and an 'ss' library that needs to link against the 'curses' library. The configuration of Kerberos might be done thus:

```
% ./configure CPPFLAGS=-I/usr/foobar/include LDFLAGS=-L/usr/foobar/lib \
    --with-system-et --with-system-ss --with-system-db \
    SS_LIB='-lss -lcurses' \
    DB_HEADER=db3/db_185.h DB_LIB=-ldb-3.3
```

In previous releases, --with- options were used to specify the compiler and linker and their options.

3.8 'osconf.h'

There is one configuration file which you may wish to edit to control various compile-time parameters in the Kerberos distribution: 'include/stock/osconf.h'. The list that follows is by no means complete, just some of the more interesting variables.

Please note: The former configuration file 'config.h' no longer exists as its functionality has been merged into the auto-configuration process. See Section 3.7 [Options to Configure], page 12.

DEFAULT_PROFILE_PATH

The pathname to the file which contains the profiles for the known realms, their KDCs, etc. The default value is /etc/krb5.conf.

The profile file format is no longer the same format as Kerberos V4's 'krb.conf' file.

DEFAULT_KEYTAB_NAME

The type and pathname to the default server keytab file (the equivalent of Kerberos V4's '/etc/srvtab'). The default is /etc/krb5.keytab.

DEFAULT_KDC_ENCTYPE

The default encryption type for the KDC. The default value is des3-cbc-sha1.

KDCRCACHE

The name of the replay cache used by the KDC. The default value is krb5kdc'rcache.

RCTMPDIR

The directory which stores replay caches. The default is to try /var/tmp, /usr/tmp, /var/usr/tmp, and /tmp.

DEFAULT_KDB_FILE

The location of the default database. The default value is /usr/local/var/krb5kdc/principal.

3.9 Shared Library Support

Shared library support is provided for a few operating systems. There are restrictions as to which compiler to use when using shared libraries. In all cases, executables linked with the shared libraries in this build process will have built in the location of the libraries, therefore obliterating the need for special LD_LIBRARY_PATH, et al environment variables when using the programs. Except where noted, multiple versions of the libraries may be installed on the same system and continue to work.

Currently the supported platforms are Solaris 2.6-2.9 (aka SunOS 5.6-5.9), Irix 6.5, Redhat Linux, MacOS 8-10, and Microsoft Windows (using DLLs).

Shared library support has been tested on the following platforms but not exhaustively (they have been built but not necessarily tested in an installed state): Tru64 (aka Alpha OSF/1 or Digital Unix) 4.0, and HP/UX 10.20.

Platforms for which there is shared library support but not significant testing include FreeBSD, OpenBSD, AIX (4.3.3), Linux, NetBSD 1.4.x (i386).

To enable shared libraries on the above platforms, run the configure script with the option '--enable-shared'.

3.10 Operating System Incompatibilities

This section details operating system incompatibilities with Kerberos V5 which have been reported to the developers at MIT. If you find additional incompatibilities, and/or discover workarounds to such problems, please send a report via the `krb5-send-pr` program. Thanks!

3.10.1 AIX

The AIX 3.2.5 linker dumps core trying to build a shared '`libkrb5.a`' produced with the GNU C compiler. The native AIX compiler works fine. This problem is fixed using the AIX 4.1 linker.

3.10.2 Alpha OSF/1 V1.3

Using the native compiler, compiling with the `-O` compiler flag causes the `asn.1` library to be compiled incorrectly.

Using GCC version 2.6.3 or later instead of the native compiler will also work fine, both with or without optimization.

3.10.3 Alpha OSF/1 V2.0

There used to be a bug when using the native compiler in compiling `md4.c` when compiled without either the `-O` or `-g` compiler options. We have changed the code and there is no problem under V2.1, but we do not have access to V2.0 to test and see if the problem would exist there. (We welcome feedback on this issue). There was never a problem in using GCC version 2.6.3.

In version 3.2 and beyond of the operating system, we have not seen this sort of problem with the native compiler.

3.10.4 Alpha OSF/1 (Digital UNIX) V4.0

The C compiler provided with Alpha OSF/1 V4.0 (a.k.a. Digital UNIX) defaults to an extended K&R C mode, not ANSI C. You need to provide the `-std` argument to the compiler (i.e., `./configure CC='cc -std'`) to enable extended ANSI C mode. More recent versions of the operating system, such as 5.0, seem to have C compilers which default to `-std`.

3.10.5 BSDI

BSDI versions 1.0 and 1.1 reportedly has a bad `sed` which causes it to go into an infinite loop during the build. The work around is to use a `sed` from somewhere else, such as GNU. (This may be true for some versions of other systems derived from BSD 4.4, such as NetBSD and FreeBSD.)

3.10.6 HPUX

The native (bundled) compiler for HPUX currently will not work, because it is not a full ANSI C compiler. The optional ANSI C compiler should work as long as you give it the `-Ae` flag (i.e. `./configure CC='cc -Ae'`). This is equivalent to `./configure CC='c89 -D_HPUX_SOURCE'`, which was the previous recommendation. This has only been tested recently for HPUX 10.20.

You will need to configure with `--disable-shared --enable-static`, because as of 1.4 we don't have support for HPUX shared library finalization routines, nor the option (yet) to ignore that lack of support (which means repeated `dlopen/dlclose` cycles on the Kerberos libraries may not be safe) and build the shared libraries anyways.

You will also need to configure the build tree with `--disable-thread-support` if you are on HPUX 10 and do not have the DCE development package installed, because that's where the `pthread.h` header file is found. (We don't know if our code will work with such a package installed, because according to some HP documentation, their `pthread.h` has to be included before any other header files, and our code doesn't do that.)

If you use GCC, it may work, but some versions of GCC have omitted certain important preprocessor defines, like `__STDC_EXT__` and `__hpux`.

3.10.7 Solaris versions 2.0 through 2.3

The `gethostbyname()` routine is broken; it does not return a fully qualified domain name, even if you are using the Domain Name Service routines. Since Kerberos V5 uses the fully qualified domain name as the second component of a service principal (i.e., `'host/tsx-11.mit.edu@ATHENA.MIT.EDU'`), this causes problems for servers who try to figure out their own fully qualified domain name.

Workarounds:

1. Supply your own resolver library. (such as `bind-4.9.3pl1` available from `ftp.vix.com`)
2. Upgrade to Solaris 2.4
3. Make sure your `/etc/nsswitch.conf` has `'files'` before `'dns'` like:

```
hosts:      files dns
```

and then in `/etc/hosts`, make sure there is a line with your workstation's IP address and hostname, with the fully qualified domain name first. Example:

```
18.172.1.4      dcl.mit.edu dcl
```

Note that making this change may cause other programs in your environment to break or behave differently.

3.10.8 Solaris 2.X

You **must** compile Kerberos V5 without the UCB compatibility libraries. This means that `'/usr/ucblib'` must not be in the `LD_LIBRARY_PATH` environment variable when you compile it. Alternatively you can use the `-i` option to `'cc'`, by using the specifying `CFLAGS=-i` option to `'configure'`.

If you are compiling for a 64-bit execution environment, you may need to configure with the option `CFLAGS="-D_XOPEN_SOURCE=500 -D__EXTENSIONS__"`. This is not well tested; at MIT we work primarily with the 32-bit execution environment.

3.10.9 Solaris 9

Solaris 9 has a kernel race condition which causes the final output written to the slave side of a pty to be lost upon the final `close()` of the slave device. This causes the dejagnum-based tests to fail intermittently. A workaround exists, but requires some help from the scheduler, and the "make check" must be executed from a shell with elevated priority limits.

Run something like

```
priocntl -s -c FX -m 30 -p 30 -i pid nnnn
```

as root, where `nnnn` is the pid of the shell whose priority limit you wish to raise.

Sun has released kernel patches for this race condition. Apply patch 117171-11 for sparc, or patch 117172-11 for x86. Later revisions of the patches should also work. It is not necessary to run "make check" from a shell with elevated priority limits once the patch has been applied.

3.10.10 SGI Irix 5.X

If you are building in a tree separate from the source tree, the vendors version of make does not work properly with regards to 'VPATH'. It also has problems with standard inference rules in 5.2 (not tested yet in 5.3) so one needs to use GNU's make.

Under 5.2, there is a bug in the optional System V `-lsocket` library in which the routine `gethostbyname()` is broken. The system supplied version in `-lc` appears to work though so one may simply specify `--with-netlib` option to 'configure'.

In 5.3, `gethostbyname()` is no longer present in `-lsocket` and is no longer an issue.

3.10.11 Ultrix 4.2/3

The DEC MIPS platform currently will not support the native compiler, since the Ultrix compiler is not a full ANSI C compiler. You should use GCC instead.

3.11 Using 'Autoconf'

(If you are not a developer, you can skip this section.)

In most of the Kerberos V5 source directories, there is a 'configure' script which automatically determines the compilation environment and creates the proper Makefiles for a particular platform. These 'configure' files are generated using 'autoconf', which can be found in the 'src/util/autoconf' directory in the distribution.

Normal users will not need to worry about running 'autoconf'; the distribution comes with the 'configure' files already prebuilt. Developers who wish to modify the 'configure.in' files should see section "Overview" in *The Autoconf Manual*.

Note that in order to run 'autoconf', you must have GNU 'm4' in your path. Before you use the 'autoconf' in the Kerberos V5 source tree, you may also need to run 'configure', and then run 'make' in the 'src/util/autoconf' directory in order to properly set up 'autoconf'.

One tool which is provided for the convenience of developers can be found in 'src/util/reconf'. This program should be run while the current directory is the top source directory. It will automatically rebuild any 'configure' files which need rebuilding. If you know that you have made a change that will require that all the 'configure' files need to be rebuilt from scratch, specify the `--force` option:

```
% cd /u1/krb5-1.10/src
% ./util/reconf --force
```

The developmental sources are a raw source tree (before it's been packaged for public release), without the pre-built 'configure' files. In order to build from such a source tree, you must do:

```
% cd krb5/util/autoconf
% ./configure
% make
% cd ../..
% util/reconf
```

Then follow the instructions for building packaged source trees (above). To install the binaries into a binary tree, do:

```
% cd /u1/krb5-1.10/src  
% make all  
% make install DESTDIR=somewhere-else
```

4 Installing Kerberos V5

The sections of this chapter describe procedures for installing Kerberos V5 on:

1. The KDCs
2. UNIX client machines
3. UNIX Application Servers

4.1 Installing KDCs

The Key Distribution Centers (KDCs) issue Kerberos tickets. Each KDC contains a copy of the Kerberos database. The master KDC contains the master copy of the database, which it propagates to the slave KDCs at regular intervals. All database changes (such as password changes) are made on the master KDC.

Slave KDCs provide Kerberos ticket-granting services, but not database administration. This allows clients to continue to obtain tickets when the master KDC is unavailable.

MIT recommends that you install all of your KDCs to be able to function as either the master or one of the slaves. This will enable you to easily switch your master KDC with one of the slaves if necessary. (See Section 4.1.7 [Switching Master and Slave KDCs], page 30.) This installation procedure is based on that recommendation.

4.1.1 Install the Master KDC

This installation procedure will require you to go back and forth a couple of times between the master KDC and each of the slave KDCs. The first few steps must be done on the master KDC.

4.1.1.1 Edit the Configuration Files

Modify the configuration files, `/etc/krb5.conf` and `/usr/local/var/krb5kdc/kdc.conf` to reflect the correct information (such as the hostnames and realm name) for your realm. MIT recommends that you keep `krb5.conf` in `/etc`.

Most of the tags in the configuration have default values that will work well for most sites. There are some tags in the `krb5.conf` file whose values must be specified, and this section will explain those as well as give an overview of all of the sections in both configuration files. For more information on changing defaults with the configuration files, see the Kerberos V5 System Administrator's Guide sections on configuration files.

4.1.1.2 `krb5.conf`

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. Normally, you should install your `krb5.conf` file in the directory `/etc`. You can override the default location by setting the environment variable `'KRB5_CONFIG'`.

The `krb5.conf` file is set up in the style of a Windows INI file. Sections are headed by the section name, in square brackets. Each section may contain zero or more relations, of the form:

```

foo = bar

```

or

```

fubar = {
    foo = bar
    baz = quux
}

```

Placing a ‘*’ at the end of a line indicates that this is the *final* value for the tag. This means that neither the remainder of this configuration file nor any other configuration file will be checked for any other values for this tag.

For example, if you have the following lines:

```

foo = bar*
foo = baz

```

then the second value of foo (baz) would never be read.

The `krb5.conf` file can include other files using either of the following directives at the beginning of a line:

```

include FILENAME
includedir DIRNAME

```

FILENAME or *DIRNAME* should be an absolute path. The named file or directory must exist and be readable. Including a directory includes all files within the directory whose names consist solely of alphanumeric characters, dashes, or underscores. Included profile files are syntactically independent of their parents, so each included file must begin with a section header.

The `krb5.conf` file may contain any or all of the following sections:

- libdefaults** Contains default values used by the Kerberos V5 library.
- login** Contains default values used by the Kerberos V5 login program.
- appdefaults** Contains default values that can be used by Kerberos V5 applications.
- realms** Contains subsections keyed by Kerberos realm names. Each subsection describes realm-specific information, including where to find the Kerberos servers for that realm.
- domain_realm** Contains relations which map domain names and subdomains onto Kerberos realm names. This is used by programs to determine what realm a host should be in, given its fully qualified domain name.
- logging** Contains relations which determine how Kerberos programs are to perform logging.
- capaths** Contains the authentication paths used with direct (nonhierarchical) cross-realm authentication. Entries in this section are used by the client to determine the intermediate realms which may be used in cross-realm authentication. It is also used by the end-service when checking the transited field for trusted intermediate realms.
- plugins** Contains tags to register dynamic plugin modules and to turn modules on and off.

If you are not using DNS TXT records, you must specify the `default_realm` in the `libdefaults` section. If you are not using DNS SRV records, you must include the `kdc` tag for each realm in the `realms` section. To communicate with the `kadmin` server in each realm, the `admin_server` tag must be set in the `realms` section. If your domain name and realm name are not the same, you must provide a translation in `domain_realm`. It is also highly recommended that you create a `[logging]` stanza if the computer will be functioning as a KDC so that the KDC and `kadmin` will generate logging output.

An example `krb5.conf` file:

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

[realms]
    ATHENA.MIT.EDU = {
        kdc = kerberos.mit.edu
        kdc = kerberos-1.mit.edu
        kdc = kerberos-2.mit.edu
        admin_server = kerberos.mit.edu
    }

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

4.1.1.3 kdc.conf

The `kdc.conf` file contains KDC configuration information, including defaults used when issuing Kerberos tickets. Normally, you should install your `kdc.conf` file in the directory `/usr/local/var/krb5kdc`. You can override the default location by setting the environment variable `'KRB5_KDC_PROFILE'`.

The `kdc.conf` file is set up in the same format as the `krb5.conf` file. (See Section 4.1.1.2 [krb5.conf], page 21.) The `kdc.conf` file may contain any or all of the following three sections:

kdcdefaults

Contains default values for overall behavior of the KDC.

realms

Contains subsections keyed by Kerberos realm names. Each subsection describes realm-specific information, including where to find the Kerberos servers for that realm.

logging

Contains relations which determine how Kerberos programs are to perform logging.

4.1.1.4 Create the Database

You will use the `kdb5_util` command *on the Master KDC* to create the Kerberos database and the optional stash file. The *stash file* is a local copy of the master key that resides in encrypted form on the KDC's local disk. The stash file is used to authenticate the KDC to itself automatically before starting the `kadmin` and `krb5kdc` daemons (*e.g.*, as part of the machine's boot sequence). The stash file, like the keytab file (see Section 4.3.1 [The Keytab File], page 35, for more information) is a potential point-of-entry for a break-in, and

if compromised, would allow unrestricted access to the Kerberos database. If you choose to install a stash file, it should be readable only by root, and should exist only on the KDC's local disk. The file should not be part of any backup of the machine, unless access to the backup data is secured as tightly as access to the master password itself.

If you choose not to install a stash file, the KDC will prompt you for the master key each time it starts up. This means that the KDC will not be able to start automatically, such as after a system reboot.

Note that `kdb5_util` will prompt you for the master key for the Kerberos database. This key can be any string. A good key is one you can remember, but that no one else can guess. Examples of bad keys are words that can be found in a dictionary, any common or popular name, especially a famous person (or cartoon character), your username in any form (*e.g.*, forward, backward, repeated twice, *etc.*), and any of the sample keys that appear in this manual. One example of a key which might be good if it did not appear in this manual is “MITiys4K5!”, which represents the sentence “MIT is your source for Kerberos 5!” (It's the first letter of each word, substituting the numeral “4” for the word “for”, and includes the punctuation mark at the end.)

The following is an example of how to create a Kerberos database and stash file on the master KDC, using the `kdb5_util` command. (The line that begins with \Rightarrow is a continuation of the previous line.) Replace *ATHENA.MIT.EDU* with the name of your Kerberos realm.

```
shell% /usr/local/sbin/kdb5_util create -r ATHENA.MIT.EDU -s
Initializing database '/usr/local/var/krb5kdc/principal' for
 $\Rightarrow$  realm 'ATHENA.MIT.EDU',
master key name 'K/M@ATHENA.MIT.EDU'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.

Enter KDC database master key:  $\Leftarrow$  Type the master password.
Re-enter KDC database master key to verify:  $\Leftarrow$  Type it again.
shell%
```

This will create five files in the directory specified in your `kdc.conf` file: two Kerberos database files, `principal.db`, and `principal.ok`; the Kerberos administrative database file, `principal.kadm5`; the administrative database lock file, `principal.kadm5.lock`; and the stash file, `.k5stash`. (The default directory is `/usr/local/var/krb5kdc`.) If you do not want a stash file, run the above command without the `-s` option.

4.1.1.5 Add Administrators to the Acl File

Next, you need create an Access Control List (`acl`) file, and put the Kerberos principal of at least one of the administrators into it. This file is used by the `kadmind` daemon to control which principals may view and make privileged modifications to the Kerberos database files. The filename should match the value you have set for “`acl_file`” in your `kdc.conf` file. The default file name is `'/usr/local/var/krb5kdc/kadm5.acl'`.

The format of the file is:

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

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. `target_principal` can also include backreferences to `Kerberos_principal`, in which “**number*” matches the component *number* in the `Kerberos_principal`.

Note: a common use of an *admin* instance is so you can 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.

The permissions are represented by single letters; UPPER-CASE 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
*	All privileges (admcil).
x	All privileges (admcil); identical to “*”.

The restrictions are a string of flags. Allowed restrictions are:

[+ -]*flagname*

flag is forced to indicated value. The permissible flags are the same as the + and - flags for the `kadmin addprinc` and `modprinc` commands.

-clearpolicy

policy is forced to clear

-policy *pol* policy is forced to be *pol*

expire *time*

pwexpire *time*

maxlife *time*

maxrenewlife *time*

associated value will be forced to MIN(*time*, requested value)

The above flags act as restrictions on any add or modify operation which is allowed due to that ACL line.

Here is an example of a `kadm5.acl` file. Note that order is important; permissions are determined by the first matching entry.

```
*/admin@ATHENA.MIT.EDU *
joadmin@ATHENA.MIT.EDU ADMCIL
joadmin/*@ATHENA.MIT.EDU il */root@ATHENA.MIT.EDU
*@ATHENA.MIT.EDU cil */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 `joadmin` has all permissions with his `admin` instance, `joadmin/admin@ATHENA.MIT.EDU` (matches the first line). He has no permissions at all with his `null` instance, `joadmin@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 their `admin` instance, but not any other `admin` instance. Any principal in the realm `ATHENA.MIT.EDU` (except for `joadmin@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 they create or modify will not be able to get postdateable tickets or tickets with a life of longer than 9 hours.

4.1.1.6 Add Administrators to the Kerberos Database

Next you need to add administrative principals to the Kerberos database. (You must add at least one now.) To do this, use `kadmin.local` on the master *KDC*. The administrative principals you create should be the ones you added to the ACL file. (See Section 4.1.1.5 [Add Administrators to the Acl File], page 24.) In the following example, the administration principal `admin/admin` is created:

```
shell% /usr/local/sbin/kadmin.local
kadmin.local: addprinc admin/admin@ATHENA.MIT.EDU
NOTICE: no policy specified for "admin/admin@ATHENA.MIT.EDU";
assigning "default".

Enter password for principal admin/admin@ATHENA.MIT.EDU: ⇐ Enter a password.
Re-enter password for principal admin/admin@ATHENA.MIT.EDU: ⇐ Type it again.
Principal "admin/admin@ATHENA.MIT.EDU" created.
kadmin.local:
```

4.1.1.7 Create a kadmind Keytab (optional)

The `kadmind` keytab is the key that the legacy administration daemons `kadmind4` and `v5passwd` will use to decrypt administrators' or clients' Kerberos tickets to determine whether or not they should have access to the database. You need to create the `kadmin` keytab with entries for the principals `kadmin/admin` and `kadmin/changepw`. (These principals are placed in the Kerberos database automatically when you create it.) To create the `kadmin` keytab, run `kadmin.local` and use the `ktadd` command, as in the following example. (The line beginning with \Rightarrow is a continuation of the previous line.):

```

shell% /usr/local/sbin/kadmin.local
kadmin.local: ktadd -k /usr/local/var/krb5kdc/kadm5.keytab
⇒ kadmin/admin kadmin/changepw
Entry for principal kadmin/admin with kvno 5, encryption
type Triple DES cbc mode with HMAC/sha1 added to keytab
WRFILE:/usr/local/var/krb5kdc/kadm5.keytab.
Entry for principal kadmin/admin with kvno 5, encryption type DES cbc mode
with CRC-32 added to keytab
WRFILE:/usr/local/var/krb5kdc/kadm5.keytab.
Entry for principal kadmin/changepw with kvno 5, encryption
type Triple DES cbc mode with HMAC/sha1 added to keytab
WRFILE:/usr/local/var/krb5kdc/kadm5.keytab.
Entry for principal kadmin/changepw with kvno 5,
encryption type DES cbc mode with CRC-32 added to keytab
WRFILE:/usr/local/var/krb5kdc/kadm5.keytab.
kadmin.local: quit
shell%

```

As specified in the ‘-k’ argument, `ktadd` will save the extracted keytab as `/usr/local/var/krb5kdc/kadm5.keytab`. The filename you use must be the one specified in your `kdc.conf` file.

4.1.1.8 Start the Kerberos Daemons on the Master KDC

At this point, you are ready to start the Kerberos daemons on the Master KDC. To do so, type:

```

shell% /usr/local/sbin/krb5kdc
shell% /usr/local/sbin/kadmind

```

Each daemon will fork and run in the background. Assuming you want these daemons to start up automatically at boot time, you can add them to the KDC’s `/etc/rc` or `/etc/inittab` file. You need to have a stash file in order to do this.

You can verify that they started properly by checking for their startup messages in the logging locations you defined in `/etc/krb5.conf`. (See Section 4.1.1.1 [Edit the Configuration Files], page 21.) For example:

```

shell% tail /var/log/krb5kdc.log
Dec 02 12:35:47 beebledbrox krb5kdc[3187](info): commencing operation
shell% tail /var/log/kadmind.log
Dec 02 12:35:52 beebledbrox kadmind[3189](info): starting

```

Any errors the daemons encounter while starting will also be listed in the logging output.

4.1.2 Install the Slave KDCs

You are now ready to start configuring the slave KDCs. Assuming you are setting the KDCs up so that you can easily switch the master KDC with one of the slaves, you should perform each of these steps on the master KDC as well as the slave KDCs, unless these instructions specify otherwise.

4.1.2.1 Create Host Keys for the Slave KDCs

Each KDC needs a host principal in the Kerberos database. You can enter these from any host, once the `kadmind` daemon is running. For example, if your master KDC were called

kerberos.mit.edu, and you had two KDC slaves named kerberos-1.mit.edu and kerberos-2.mit.edu, you would type the following:

```
shell% /usr/local/sbin/kadmin
kadmin: addprinc -randkey host/kerberos.mit.edu
NOTICE: no policy specified for "host/kerberos.mit.edu@ATHENA.MIT.EDU";
assigning "default"
Principal "host/kerberos.mit.edu@ATHENA.MIT.EDU" created.
kadmin: addprinc -randkey host/kerberos-1.mit.edu
NOTICE: no policy specified for "host/kerberos-1.mit.edu@ATHENA.MIT.EDU";
assigning "default"
Principal "host/kerberos-1.mit.edu@ATHENA.MIT.EDU" created.
kadmin: addprinc -randkey host/kerberos-2.mit.edu
NOTICE: no policy specified for "host/kerberos-2.mit.edu@ATHENA.MIT.EDU";
assigning "default"
Principal "host/kerberos-2.mit.edu@ATHENA.MIT.EDU" created.
kadmin:
```

It is not actually necessary to have the master KDC server in the Kerberos database, but it can be handy if:

- anyone will be logging into the machine as something other than root
- you want to be able to swap the master KDC with one of the slaves if necessary.

4.1.2.2 Extract Host Keytabs for the KDCs

Each KDC (including the master) needs a keytab to decrypt tickets. Ideally, you should extract each keytab locally on its own KDC. If this is not feasible, you should use an encrypted session to send them across the network. To extract a keytab on a KDC called kerberos.mit.edu, you would execute the following command:

```
kadmin: ktadd host/kerberos.mit.edu
kadmin: Entry for principal host/kerberos.mit.edu@ATHENA.MIT.EDU with
kvno 1, encryption type DES-CBC-CRC added to keytab
WRFILE:/etc/krb5.keytab.
kadmin:
```

Note that the principal must exist in the Kerberos database in order to extract the keytab.

4.1.2.3 Set Up the Slave KDCs for Database Propagation

The database is propagated from the master KDC to the slave KDCs via the kpropd daemon. To set up propagation, create a file on each KDC, named /usr/local/var/krb5kdc/kpropd.acl, containing the principals for each of the KDCs.

For example, if the master KDC were kerberos.mit.edu, the slave KDCs were kerberos-1.mit.edu and kerberos-2.mit.edu, and the realm were ATHENA.MIT.EDU, then the file's contents would be:

```
host/kerberos.mit.edu@ATHENA.MIT.EDU
host/kerberos-1.mit.edu@ATHENA.MIT.EDU
host/kerberos-2.mit.edu@ATHENA.MIT.EDU
```

Then, add the following line to `/etc/inetd.conf` file on each KDC:

```
krb5_prop stream tcp nowait root /usr/local/sbin/kpropd kpropd
```

You also need to add the following lines to `/etc/services` on each KDC:

```
kerberos      88/udp      kdc           # Kerberos authentication (udp)
kerberos      88/tcp      kdc           # Kerberos authentication (tcp)
krb5_prop     754/tcp      kpropd        # Kerberos slave propagation
kerberos-adm  749/tcp      kadmin        # Kerberos 5 admin/changepw (tcp)
kerberos-adm  749/udp      kadmin        # Kerberos 5 admin/changepw (udp)
```

4.1.3 Back on the Master KDC

Now that the slave KDCs are able to accept database propagation, you'll need to propagate the database to each of them.

4.1.3.1 Propagate the Database to Each Slave KDC

First, create a dump of the database on the master KDC, as follows:

```
shell% /usr/local/sbin/kdb5_util dump /usr/local/var/krb5kdc/slave_datatrans
shell%
```

Next, you need to manually propagate the database to each slave KDC, as in the following example. (The lines beginning with `⇒` are continuations of the previous line.):

```
/usr/local/sbin/kprop -f /usr/local/var/krb5kdc/slave_datatrans
⇒ kerberos-1.mit.edu
/usr/local/sbin/kprop -f /usr/local/var/krb5kdc/slave_datatrans
⇒ kerberos-2.mit.edu
```

You will need a script to dump and propagate the database. The following is an example of a bourne shell script that will do this. (Note that the line that begins with `⇒` is a continuation of the previous line. Remember that you need to replace `/usr/local` with the name of the directory in which you installed Kerberos V5.)

```
#!/bin/sh

kdclist = "kerberos-1.mit.edu kerberos-2.mit.edu"

/usr/local/sbin/kdb5_util "dump
⇒ /usr/local/var/krb5kdc/slave_datatrans"

for kdc in $kdclist
do
/usr/local/sbin/kprop -f /usr/local/var/krb5kdc/slave_datatrans $kdc
done
```

You will need to set up a cron job to run this script at the intervals you decided on earlier (See Section 2.6 [Database Propagation], page 6.)

4.1.4 Finish Installing the Slave KDCs

Now that the slave KDCs have copies of the Kerberos database, you can create stash files for them and start the `krb5kdc` daemon.

4.1.4.1 Create Stash Files on the Slave KDCs

Create stash files, by issuing the following commands on each slave KDC:

```
shell% kdb5_util stash
kdb5_util: Cannot find/read stored master key while reading master key
kdb5_util: Warning: proceeding without master key

Enter KDC database master key:  ← Enter the database master key.
shell%
```

As mentioned above, the stash file is necessary for your KDCs to be able to authenticate to themselves, such as when they reboot. You could run your KDCs without stash files, but you would then need to type in the Kerberos database master key by hand every time you start a KDC daemon.

4.1.4.2 Start the krb5kdc Daemon on Each KDC

The final step in configuring your slave KDCs is to run the KDC daemon:

```
shell% /usr/local/sbin/krb5kdc
```

As with the master KDC, you will probably want to add this command to the KDCs' `/etc/rc` or `/etc/inittab` files, so they will start the `krb5kdc` daemon automatically at boot time.

4.1.5 Add Kerberos Principals to the Database

Once your KDCs are set up and running, you are ready to use `kadmin` to load principals for your users, hosts, and other services into the Kerberos database. This procedure is described fully in the “Adding or Modifying Principals” section of the Kerberos V5 System Administrator’s Guide. (See Section 4.1.2.1 [Create Host Keys for the Slave KDCs], page 27, for a brief description.) The keytab is generated by running `kadmin` and issuing the `ktadd` command.

4.1.6 Limit Access to the KDCs

To limit the possibility that your Kerberos database could be compromised, MIT recommends that each KDC be a dedicated host, with limited access. If your KDC is also a file server, FTP server, Web server, or even just a client machine, someone who obtained root access through a security hole in any of those areas could gain access to the Kerberos database.

4.1.7 Switching Master and Slave KDCs

You may occasionally want to use one of your slave KDCs as the master. This might happen if you are upgrading the master KDC, or if your master KDC has a disk crash.

Assuming you have configured all of your KDCs to be able to function as either the master KDC or a slave KDC (as this document recommends), all you need to do to make the changeover is:

If the master KDC is still running, do the following on the *old* master KDC:

1. Kill the `kadmind` process.
2. Disable the cron job that propagates the database.

3. Run your database propagation script manually, to ensure that the slaves all have the latest copy of the database. (See Section 4.1.3.1 [Propagate the Database to Each Slave KDC], page 29.) If there is a need to preserve per-principal policy information from the database, you should do a “`kdb5_util dump -ov`” in order to preserve that information and propagate that dump file securely by some means to the slave so that its database has the correct state of the per-principal policy information.

On the *new* master KDC:

1. Create a database keytab. (See Section 4.1.1.7 [Create a `kadmind` Keytab (optional)], page 26.)
2. Start the `kadmind` daemon. (See Section 4.1.1.8 [Start the Kerberos Daemons], page 27.)
3. Set up the cron job to propagate the database. (See Section 4.1.3.1 [Propagate the Database to Each Slave KDC], page 29.)
4. Switch the CNAMEs of the old and new master KDCs. (If you don’t do this, you’ll need to change the `krb5.conf` file on every client machine in your Kerberos realm.)

4.1.8 Incremental Database Propagation

At some very large sites, dumping and transmitting the database can take more time than is desirable for changes to propagate from the master KDC to the slave KDCs. The incremental propagation support added in the 1.7 release is intended to address this.

With incremental propagation enabled, all programs on the master KDC that change the database also write information about the changes to an “update log” file, maintained as a circular buffer of a certain size. A process on each slave KDC connects to a service on the master KDC (currently implemented in the `kadmind` server) and periodically requests the changes that have been made since the last check. By default, this check is done every two minutes. If the database has just been modified in the previous several seconds (currently the threshold is hard-coded at 10 seconds), the slave will not retrieve updates, but instead will pause and try again soon after. This reduces the likelihood that incremental update queries will cause delays for an administrator trying to make a bunch of changes to the database at the same time.

Incremental propagation uses the following entries in the per-realm data in the KDC config file:

`iprop_enable` (boolean)

If this is set to `true`, then incremental propagation is enabled, and (as noted below) normal `kprop` propagation is disabled. The default is `false`.

`iprop_master_ulogsize` (integer)

This indicates the number of entries that should be retained in the update log. The default is 1000; the maximum number is 2500.

`iprop_slave_poll` (time interval)

This indicates how often the slave should poll the master KDC for changes to the database. The default is two minutes.

iprop_port (integer)

This specifies the port number to be used for incremental propagation. This is required in both master and slave configuration files.

iprop_logfile (file name)

This specifies where the update log file for the realm database is to be stored. The default is to use the **database_name** entry from the **realms** section of the config file, with **.u`log`** appended. (NOTE: If **database_name** isn't specified in the **realms** section, perhaps because the LDAP database back end is being used, or the file name is specified in the **dbmodules** section, then the hard-coded default for **database_name** is used. Determination of the **iprop_logfile** default value will not use values from the **dbmodules** section.)

Both master and slave sides must have principals named **kiprop/hostname** (where *hostname* is, as usual, the lower-case, fully-qualified, canonical name for the host) registered and keys stored in the default keytab file (**/etc/krb5.keytab**).

On the master KDC side, the **kiprop/hostname** principal must be listed in the **kadmind** ACL file **kadm5.acl**, and given the **p** privilege.

On the slave KDC side, **kpropd** should be run. When incremental propagation is enabled, it will connect to the **kadmind** on the master KDC and start requesting updates.

The normal **kprop** mechanism is disabled by the incremental propagation support. However, if the slave has been unable to fetch changes from the master KDC for too long (network problems, perhaps), the log on the master may wrap around and overwrite some of the updates that the slave has not yet retrieved. In this case, the slave will instruct the master KDC to dump the current database out to a file and invoke a one-time **kprop** propagation, with special options to also convey the point in the update log at which the slave should resume fetching incremental updates. Thus, all the keytab and ACL setup previously described for **kprop** propagation is still needed.

There are several restrictions in the current implementation:

- Changes to password policy objects are not propagated incrementally. Changes to which policy applies to a principal are propagated.
- The master and slave must be able to initiate TCP connections in both directions, without an intervening NAT.
- If the slave has an IPv6 interface address but needs to accept connections over IPv4, the operating system needs “dual stack” support (i.e. the ability to accept IPv6 and IPv4 connections on a single IPv6 listener socket). At this time, all modern Unix-like operating systems have dual stack support except OpenBSD.

4.1.8.1 Sun/MIT Incremental Propagation Differences

Sun donated the original code for supporting incremental database propagation to MIT. Some changes have been made in the MIT source tree that will be visible to administrators. (These notes are based on Sun's patches. Changes to Sun's implementation since then may not be reflected here.)

The Sun config file support looks for `sunw_dbprop_enable`, `sunw_dbprop_master_ologsize`, and `sunw_dbprop_slave_poll`.

The incremental propagation service is implemented as an ONC RPC service. In the Sun implementation, the service is registered with `rpcbind` (also known as `portmapper`) and the client looks up the port number to contact. In the MIT implementation, where interaction with some modern versions of `rpcbind` doesn't always work well, the port number must be specified in the config file on both the master and slave sides.

The Sun implementation hard-codes pathnames in `‘/var/krb5’` for the update log and the per-slave `kprop` dump files. In the MIT implementation, the pathname for the update log is specified in the config file, and the per-slave dump files are stored in `/usr/local/var/krb5kdc/slave_datatrans_hostname`.

4.2 Installing and Configuring UNIX Client Machines

Client machine installation is much more straightforward than installation of the KDCs.

4.2.1 Client Programs

The Kerberized client programs are `kinit`, `klist`, `kdestroy`, `kpasswd`, and `ksu`. All of these programs are in the directory `/usr/local/bin`.

MIT recommends that you use `login.krb5` in place of `/bin/login` to give your users a single-sign-on system. You will need to make sure your users know to use their Kerberos passwords when they log in.

You will also need to educate your users to use the ticket management programs `kinit`, `klist`, `kdestroy`, and to use the Kerberos programs `ksu` and `kpasswd` in place of their non-Kerberos counterparts `su` and `passwd`.

4.2.2 Client Machine Configuration Files

Each machine running Kerberos must have a `/etc/krb5.conf` file. (See Section 4.1.1.2 [krb5.conf], page 21.)

Also, for most UNIX systems, you must add the appropriate Kerberos services to each client machine's `/etc/services` file. If you are using the default configuration for Kerberos V5, you should be able to just insert the following code:

```
kerberos      88/udp    kdc      # Kerberos V5 KDC
kerberos      88/tcp    kdc      # Kerberos V5 KDC
kerberos-adm  749/tcp    kdc      # Kerberos 5 admin/changepw
kerberos-adm  749/udp    kdc      # Kerberos 5 admin/changepw
krb5_prop     754/tcp    kdc      # Kerberos slave propagation
krb524        4444/tcp   kdc      # Kerberos 5 to 4 ticket translator
```

4.2.2.1 Mac OS X Configuration

To install Kerberos V5 on Mac OS X and Mac OS X Server, follow the directions for generic Unix-based OS's, except for the `/etc/services` updates described above.

Mac OS X and Mac OS X Server use a database called NetInfo to store the contents of files normally found in `/etc`. Instead of modifying `/etc/services`, you should run the following commands to add the Kerberos service entries to NetInfo:

```
$ niutil -create . /services/kerberos
$ niutil -createprop . /services/kerberos name kerberos kdc
$ niutil -createprop . /services/kerberos port 750
$ niutil -createprop . /services/kerberos protocol tcp udp
$ niutil -create . /services/krbupdate
$ niutil -createprop . /services/krbupdate name krbupdate kreg
$ niutil -createprop . /services/krbupdate port 760
$ niutil -createprop . /services/krbupdate protocol tcp
$ niutil -create . /services/kpasswd
$ niutil -createprop . /services/kpasswd name kpasswd kpwd
$ niutil -createprop . /services/kpasswd port 761
$ niutil -createprop . /services/kpasswd protocol tcp
$ niutil -create . /services/klogin
$ niutil -createprop . /services/klogin port 543
$ niutil -createprop . /services/klogin protocol tcp
$ niutil -create . /services/eklogin
$ niutil -createprop . /services/eklogin port 2105
$ niutil -createprop . /services/eklogin protocol tcp
$ niutil -create . /services/kshell
$ niutil -createprop . /services/kshell name kshell krcmd
$ niutil -createprop . /services/kshell port 544
$ niutil -createprop . /services/kshell protocol tcp
```

In addition to adding services to NetInfo, you must also modify the resolver configuration in NetInfo so that the machine resolves its own hostname as a FQDN (fully qualified domain name). By default, Mac OS X and Mac OS X Server machines query NetInfo to resolve hostnames before falling back to DNS. Because NetInfo has an unqualified name for all the machines in the NetInfo database, the machine's own hostname will resolve to an unqualified name. Kerberos needs a FQDN to look up keys in the machine's keytab file.

Fortunately, you can change the `lookupd` caching order to query DNS first. Run the following NetInfo commands and reboot the machine:

```
$ niutil -create . /locations/lookupd/hosts
$ niutil -createprop . /locations/lookupd/hosts LookupOrder CacheAgent DNSAgent
  NIAGENT NILAGENT
```

Once you have rebooted, you can verify that the resolver now behaves correctly. Compile the Kerberos 5 distribution and run:

```
$ cd .../src/tests/resolve
$ ./resolve
```

This will tell you whether or not your machine returns FQDNs on name lookups. If the test still fails, you can also try turning off DNS caching. Run the following commands and reboot:

```
$ niutil -create . /locations/lookupd/hosts
$ niutil -createprop . /locations/lookupd/hosts LookupOrder DNSAgent
CacheAgent NIAgent NILAgent
```

The remainder of the setup of a Mac OS X client machine or application server should be the same as for other UNIX-based systems.

4.3 UNIX Application Servers

An application server is a host that provides one or more services over the network. Application servers can be “secure” or “insecure.” A “secure” host is set up to require authentication from every client connecting to it. An “insecure” host will still provide Kerberos authentication, but will also allow unauthenticated clients to connect.

If you have Kerberos V5 installed on all of your client machines, MIT recommends that you make your hosts secure, to take advantage of the security that Kerberos authentication affords. However, if you have some clients that do not have Kerberos V5 installed, you can run an insecure server, and still take advantage of Kerberos V5’s single sign-on capability.

4.3.1 The Keytab File

All Kerberos server machines need a *keytab* file, called `/etc/krb5.keytab`, to authenticate to the KDC. The keytab file is an encrypted, local, on-disk copy of the host’s key. The keytab file, like the stash file (Section 4.1.1.4 [Create the Database], page 23) is a potential point-of-entry for a break-in, and if compromised, would allow unrestricted access to its host. The keytab file should be readable only by root, and should exist only on the machine’s local disk. The file should not be part of any backup of the machine, unless access to the backup data is secured as tightly as access to the machine’s root password itself.

In order to generate a keytab for a host, the host must have a principal in the Kerberos database. The procedure for adding hosts to the database is described fully in the “Adding or Modifying Principals” section of the *Kerberos V5 System Administrator’s Guide*. See Section 4.1.2.1 [Create Host Keys for the Slave KDCs], page 27. for a brief description.) The keytab is generated by running `kadmin` and issuing the `ktadd` command.

For example, to generate a keytab file to allow the host `trillium.mit.edu` to authenticate for the services `host`, `ftp`, and `pop`, the administrator `joadmin` would issue the command (on `trillium.mit.edu`):

```
trillium% /usr/local/sbin/kadmin
kadmin5: ktadd host/trillium.mit.edu ftp/trillium.mit.edu
⇒ pop/trillium.mit.edu
kadmin: Entry for principal host/trillium.mit.edu@ATHENA.MIT.EDU with
kvno 3, encryption type DES-CBC-CRC added to keytab
WRFILE:/etc/krb5.keytab.
kadmin: Entry for principal ftp/trillium.mit.edu@ATHENA.MIT.EDU with
kvno 3, encryption type DES-CBC-CRC added to keytab
WRFILE:/etc/krb5.keytab.
kadmin: Entry for principal pop/trillium.mit.edu@ATHENA.MIT.EDU with
kvno 3, encryption type DES-CBC-CRC added to keytab
WRFILE:/etc/krb5.keytab.
kadmin5: quit
trillium%
```

If you generate the keytab file on another host, you need to get a copy of the keytab file onto the destination host (`trillium`, in the above example) without sending it unencrypted over the network.

4.3.2 Some Advice about Secure Hosts

Kerberos V5 can protect your host from certain types of break-ins, but it is possible to install Kerberos V5 and still leave your host vulnerable to attack. Obviously an installation guide is not the place to try to include an exhaustive list of countermeasures for every possible attack, but it is worth noting some of the larger holes and how to close them.

We recommend that backups of secure machines exclude the keytab file (`/etc/krb5.keytab`). If this is not possible, the backups should at least be done locally, rather than over a network, and the backup tapes should be physically secured.

The keytab file and any programs run by root, including the Kerberos V5 binaries, should be kept on local disk. The keytab file should be readable only by root.

5 Upgrading Existing Kerberos V5 Installations

If you already have an existing Kerberos database that you created with a prior release of Kerberos 5, you can upgrade it to work with the current release with the `kdb5_util` command. It is only necessary to perform this dump/undump procedure if you were running a `krb5-1.0.x` KDC and are migrating to a `krb5-1.1.x` or newer KDC or if you were running a `krb5-1.1.x` KDC and are migrating to a `krb5-1.2.x` or newer KDC. The process for upgrading a Master KDC involves the following steps:

1. Stop your current KDC and administration server processes, if any.
2. Dump your existing Kerberos database to an ASCII file with `kdb5_util`'s “dump” command:

```
shell% cd /usr/local/var/krb5kdc
shell% kdb5_util dump old-kdb-dump
shell% kdb5_util dump -ov old-kdb-dump.ov
shell%
```

3. Create a new Master KDC installation (See Section 4.1.1 [Install the Master KDC], page 21.). If you have a stash file for your current database, choose any new master password but then copy your existing stash file to the location specified by your `kdc.conf`; if you do not have a stash file for your current database, you must choose the same master password.
4. Load your old Kerberos database into the new system with `kdb5_util`'s “load” command:

```
shell% cd /usr/local/var/krb5kdc
shell% kdb5_util load old-kdb-dump
shell% kdb5_util load -update old-kdb-dump.ov
shell%
```

The “dump -ov” and “load -update” commands are necessary in order to preserve per-principal policy information, since the default dump format filters out that information. If you omit those steps, the loaded database database will lose the policy information for each principal that has a policy.

To update a Slave KDC, you must stop the old server processes on the Slave KDC, install the new server binaries, reload the most recent slave dump file, and re-start the server processes.

5.1 Upgrading to Triple-DES Encryption Keys

Beginning with the 1.2 release from MIT, Kerberos includes a stronger encryption algorithm called “triple DES” – essentially, three applications of the basic DES encryption algorithm, greatly increasing the resistance to a brute-force search for the key by an attacker. This algorithm is more secure, but encryption is much slower.

Release 1.1 had some support for triple-DES service keys, but with release 1.2 we have added support for user keys and session keys as well. Release 1.0 had very little support for multiple cryptosystems, and some of that software may not function properly in an environment using triple-DES as well as plain DES.

In the 1.3 release from MIT, Kerberos also includes the RC4 encryption algorithm, a stream cipher symmetric key algorithm developed in 1987 by Ronald Rivest at RSA Data Security. Please note that RC4 is not part of the IETF standard.

Because of the way the MIT Kerberos database is structured, the KDC will assume that a service supports only those encryption types for which keys are found in the database. Thus, if a service has only a single-DES key in the database, the KDC will not issue tickets for that service that use triple-DES or RC4 session keys; it will instead issue only single-DES session keys, even if other services are already capable of using triple-DES or RC4. So if you make sure your application server software is updated before adding a triple-DES or RC4 key for the service, clients should be able to talk to services at all times during the updating process.

Normally, the listed `supported_enctypes` in `kdc.conf` are all used when a new key is generated. You can control this with command-line flags to `kadmin` and `kadmin.local`. You may want to exclude triple-DES and RC4 by default until you have updated a lot of your application servers, and then change the default to include triple-DES and RC4. We recommend that you always include `des-cbc-crc` in the default list.

6 Bug Reports for Kerberos V5

In any complex software, there will be bugs. If you have successfully built and installed Kerberos V5, please use the `krb5-send-pr` program to fill out a Problem Report should you encounter any errors in our software.

Bug reports that include proposed fixes are especially welcome. If you do include fixes, please send them using either context diffs or unified diffs (using ‘`diff -c`’ or ‘`diff -u`’, respectively). Please be careful when using “cut and paste” or other such means to copy a patch into a bug report; depending on the system being used, that can result in converting TAB characters into spaces, which makes applying the patches more difficult.

The `krb5-send-pr` program is installed in the directory `/usr/local/sbin`.

The `krb5-send-pr` program enters the problem report into our Problem Report Management System (PRMS), which automatically assigns it to the engineer best able to help you with problems in the assigned category.

The `krb5-send-pr` program will try to intelligently fill in as many fields as it can. You need to choose the *category*, *class*, *severity*, and *priority* of the problem, as well as giving us as much information as you can about its exact nature.

The PR **category** will be one of:

<code>krb5-admin</code>	<code>krb5-appl</code>	<code>krb5-build</code>	<code>krb5-clients</code>
<code>krb5-doc</code>	<code>krb5-kdc</code>	<code>krb5-libs</code>	<code>krb5-misc</code>
<code>pty</code>	<code>telnet</code>	<code>test</code>	

Choose the category that best describes the area under which your problem falls.

The **class** can be *sw-bug*, *doc-bug*, *change-request*, or *support*. The first two are exactly as their names imply. Use *change-request* when the software is behaving according to specifications, but you want to request changes in some feature or behavior. The *support* class is intended for more general questions about building or using Kerberos V5.

The **severity** of the problem indicates the problem’s impact on the usability of Kerberos V5. If a problem is *critical*, that means the product, component or concept is completely non-operational, or some essential functionality is missing, and no workaround is known. A *serious* problem is one in which the product, component or concept is not working properly or significant functionality is missing. Problems that would otherwise be considered *critical* are rated *serious* when a workaround is known. A *non-critical* problem is one that is indeed a problem, but one that is having a minimal effect on your ability to use Kerberos V5. *E.g.*, The product, component or concept is working in general, but lacks features, has irritating behavior, does something wrong, or doesn’t match its documentation. The default severity is *serious*.

The **priority** indicates how urgent this particular problem is in relation to your work. Note that low priority does not imply low importance. A priority of *high* means a solution is needed as soon as possible. A priority of *medium* means the problem should be solved no later than the next release. A priority of *low* means the problem should be solved in a future release, but it is not important to your work how soon this happens. The default priority is *medium*.

Note that a given severity does not necessarily imply a given priority. For example, a non-critical problem might still have a high priority if you are faced with a hard deadline. Conversely, a serious problem might have a low priority if the feature it is disabling is one that you do not need.

It is important that you fill in the *release* field and tell us what changes you have made, if any.

A sample filled-out form from a company named “Toasters, Inc.” might look like this:

```
To: krb5-bugs@mit.edu
Subject: misspelled "Kerberos" in title of installation guide
From: jcb
Reply-To: jcb
Cc:
X-send-pr-version: 3.99

>Submitter-Id: mit
>Originator: Jeffrey C. Gilman Bigler
>Organization:
mit
>Confidential: no
>Synopsis: Misspelled "Kerberos" in title of installation guide
>Severity: non-critical
>Priority: low
>Category: krb5-doc
>Class: doc-bug
>Release: 1.0-development
>Environment:
<machine, os, target, libraries (multiple lines)>
System: ULTRIX imbrum 4.2 0 RISC
Machine: mips
>Description:
    Misspelled "Kerberos" in title of "Kerboros V5 Installation Guide"
>How-To-Repeat:
    N/A
>Fix:
    Correct the spelling.
```

If the `krb5-send-pr` program does not work for you, or if you did not get far enough in the process to have an installed and working `krb5-send-pr`, you can generate your own form, using the above as an example.

Appendix A Copyright

Copyright © 1985-2013 by the Massachusetts Institute of Technology.

All rights reserved.

Export of software employing encryption 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 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 M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore if you modify this software you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original MIT software. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided “as is” without express or implied warranty.

Documentation components of this software distribution are licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. (<http://creativecommons.org/licenses/by-sa/3.0/>)

Individual source code files are copyright MIT, Cygnus Support, Novell, OpenVision Technologies, Oracle, Red Hat, Sun Microsystems, FundsXpress, and others.

Project Athena, Athena, Athena MUSE, Discuss, Hesiod, Kerberos, Moira, and Zephyr are trademarks of the Massachusetts Institute of Technology (MIT). No commercial use of these trademarks may be made without prior written permission of MIT.

“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).

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`:

Copyright, OpenVision Technologies, Inc., 1993-1996, All Rights Reserved

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 <crowdad@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.

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

Copyright © 1998 by the FundsXpress, INC.

All rights reserved.

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 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The implementation of the AES encryption algorithm in `src/lib/crypto/builtin/aes` has the following copyright:

Copyright © 2001, Dr Brian Gladman <brg@gladman.uk.net>, Worcester, UK.

All rights reserved.

LICENSE TERMS

The free distribution and use of this software in both source and binary form is allowed (with or without changes) provided that:

1. distributions of this source code include the above copyright notice, this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other associated materials;
3. the copyright holder’s name is not used to endorse products built using this software without specific written permission.

DISCLAIMER

This software is provided ‘as is’ with no explicit or implied warranties in respect of any properties, including, but not limited to, correctness and fitness for purpose.

Portions contributed by Red Hat, including the pre-authentication plug-in framework and the NSS crypto implementation, contain the following copyright:

Copyright © 2006 Red Hat, Inc.
Portions copyright © 2006 Massachusetts Institute of Technology
All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Red Hat, Inc., nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED 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 bundled verito source code is subject to the following license:

Copyright 2011 Red Hat, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

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

```

and the initial implementation of incremental propagation, including the following new or changed files:

```

include/iprop_hdr.h
kadmin/server/ipropd_svc.c
lib/kdb/iprop.x
lib/kdb/kdb_convert.c
lib/kdb/kdb_log.c
lib/kdb/kdb_log.h
lib/krb5/error_tables/kdb5_err.et
slave/kpropd_rpc.c
slave/kproplog.c

```

are subject to the following license:

Copyright © 2004 Sun Microsystems, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Kerberos V5 includes documentation and software developed at the University of California at Berkeley, which includes this copyright notice:

Copyright © 1983 Regents of the University of California.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS 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.

Portions contributed by Novell, Inc., including the LDAP database backend, are subject to the following license:

Copyright © 2004-2005, Novell, Inc.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- The copyright holder's name is not used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED 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.

Portions funded by Sandia National Laboratory and developed by the University of Michigan's Center for Information Technology Integration, including the PKINIT implementation, are subject to the following license:

COPYRIGHT © 2006-2007
THE REGENTS OF THE UNIVERSITY OF MICHIGAN
ALL RIGHTS RESERVED

Permission is granted to use, copy, create derivative works and redistribute this software and such derivative works for any purpose, so long as the name of The University of Michigan is not used in any advertising or publicity pertaining to the use of distribution of this software without specific, written prior authorization. If the above copyright notice or any other identification of the University of Michigan is included in any copy of any portion of this software, then the disclaimer below must also be included.

THIS SOFTWARE IS PROVIDED AS IS, WITHOUT REPRESENTATION FROM THE UNIVERSITY OF MICHIGAN AS TO ITS FITNESS FOR ANY PURPOSE, AND WITHOUT WARRANTY BY THE UNIVERSITY OF MICHIGAN OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE REGENTS OF THE UNIVERSITY OF MICHIGAN SHALL NOT BE LIABLE FOR ANY DAMAGES, INCLUDING SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WITH RESPECT TO ANY CLAIM ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SOFTWARE, EVEN IF IT HAS BEEN OR IS HEREAFTER ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The pkcs11.h file included in the PKINIT code has the following license:

Copyright 2006 g10 Code GmbH
Copyright 2006 Andreas Jellinghaus

This file is free software; as a special exception the author gives unlimited permission to copy and/or distribute it, with or without modifications, as long as this notice is preserved.

This file is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY, to the extent permitted by law; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Portions contributed by Apple Inc. are subject to the following license:

Copyright 2004-2008 Apple Inc. All Rights Reserved.

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 Apple Inc. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Apple Inc. 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 MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The implementations of UTF-8 string handling in `src/util/support` and `src/lib/krb5/unicode` are subject to the following copyright and permission notice:

The OpenLDAP Public License
Version 2.8, 17 August 2003

Redistribution and use of this software and associated documentation (“Software”), with or without modification, are permitted provided that the following conditions are met:

1. Redistributions in source form must retain copyright statements and notices,
2. Redistributions in binary form must reproduce applicable copyright statements and notices, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution, and
3. Redistributions must contain a verbatim copy of this document.

The OpenLDAP Foundation may revise this license from time to time. Each revision is distinguished by a version number. You may use this Software under terms of this license revision or under the terms of any subsequent revision of the license.

THIS SOFTWARE IS PROVIDED BY THE OPENLDAP FOUNDATION AND ITS CONTRIBUTORS “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OPENLDAP FOUNDATION, ITS CONTRIBUTORS, OR THE AUTHOR(S) OR OWNER(S) OF THE SOFTWARE 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 names of the authors and copyright holders must not be used in advertising or otherwise to promote the sale, use or other dealing in this Software without specific, written prior permission. Title to copyright in this Software shall at all times remain with copyright holders.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Copyright 1999-2003 The OpenLDAP Foundation, Redwood City, California, USA. All Rights Reserved. Permission to copy and distribute verbatim copies of this document is granted.

Marked test programs in `src/lib/krb5/krb` have the following copyright:

Copyright © 2006 Kungliga Tekniska Högskolan
(Royal Institute of Technology, Stockholm, Sweden).
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of KTH nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY KTH AND ITS CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL KTH OR ITS 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.

Portions of the RPC implementation in `src/lib/rpc` and `src/include/gssrpc` have the following copyright and permission notice:

Copyright © 2010, Oracle America, Inc.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the “Oracle America, Inc.” nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER 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.

Copyright © 2006,2007,2009 NTT (Nippon Telegraph and Telephone Corporation). All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY NTT “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL NTT 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.

Copyright 2000 by Carnegie Mellon University

All Rights Reserved

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 Carnegie Mellon University not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

CARNEGIE MELLON UNIVERSITY DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CARNEGIE MELLON UNIVERSITY BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright © 2002 Naval Research Laboratory (NRL/CCS)

Permission to use, copy, modify and distribute this software and its documentation is hereby granted, provided that both the copyright notice and this permission notice appear in all copies of the software, derivative works or modified versions, and any portions thereof.

NRL ALLOWS FREE USE OF THIS SOFTWARE IN ITS “AS IS” CONDITION AND DISCLAIMS ANY LIABILITY OF ANY KIND FOR ANY DAMAGES WHATSOEVER RESULTING FROM THE USE OF THIS SOFTWARE.

Portions extracted from Internet RFCs have the following copyright notice:

Copyright © The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an “AS IS” basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright © 1991, 1992, 1994 by Cygnus Support.

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. Cygnus Support makes no representations about the suitability of this software for any purpose. It is provided “as is” without express or implied warranty.

Copyright © 2006 Secure Endpoints Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Portions of the implementation of the Fortuna-like PRNG are subject to the following notice:

Copyright © 2005 Marko Kreen
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR 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.

Copyright © 1994 by the University of Southern California

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 copy, modify, and distribute this software and its documentation in source and binary forms is hereby granted, provided that any documentation or other materials related to such distribution or use acknowledge that the software was developed by the University of Southern California.

DISCLAIMER OF WARRANTY. THIS SOFTWARE IS PROVIDED “AS IS”. The University of Southern California MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. By way of example, but not limitation, the University of Southern California MAKES NO REPRESENTATIONS OR WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. The University of Southern California shall not be held liable for any liability nor for any direct, indirect, or consequential damages with respect to any claim by the user or distributor of the ksu software.

Copyright © 1995

The President and Fellows of Harvard University

This code is derived from software contributed to Harvard by Jeremy Rassen.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

This product includes software developed by the University of California, Berkeley and its contributors.
4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS

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.

Copyright © 2008 by the Massachusetts Institute of Technology.
 Copyright 1995 by Richard P. Basch. All Rights Reserved.
 Copyright 1995 by Lehman Brothers, Inc. All Rights Reserved.

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 Richard P. Basch, Lehman Brothers and M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Richard P. Basch, Lehman Brothers and M.I.T. make no representations about the suitability of this software for any purpose. It is provided “as is” without express or implied warranty.

The following notice applies to `src/lib/krb5/krb/strptime.c`:

Copyright © 1997, 1998 The NetBSD Foundation, Inc.
 All rights reserved.

This code was contributed to The NetBSD Foundation by Klaus Klein.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
 This product includes software developed by the NetBSD Foundation, Inc.
 and its contributors.
4. Neither the name of The NetBSD Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION 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 following notice applies to Unicode library files in `src/lib/krb5/unicode`:

Copyright 1997, 1998, 1999 Computing Research Labs,
New Mexico State University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following notice applies to `src/util/support/strlcpy.c`:

Copyright © 1998 Todd C. Miller <Todd.Miller@courtesan.com>

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The following notice applies to `src/util/profile/argv_parse.c` and `src/util/profile/argv_parse.h`:

Copyright 1999 by Theodore Ts'o.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS" AND THEODORE TS'O (THE AUTHOR) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN

NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. (Isn't it sick that the U.S. culture of lawsuit-happy lawyers requires this kind of disclaimer?)

The following notice applies to SWIG-generated code in `src/util/profile/profile_tcl.c`:

Copyright © 1999-2000, The University of Chicago

This file may be freely redistributed without license or fee provided this copyright message remains intact.

The following notice applies to portions of `src/lib/rpc` and `src/include/gssrpc`:

Copyright © 2000 The Regents of the University of Michigan. All rights reserved.

Copyright © 2000 Dug Song <dugsong@UMICH.EDU>. All rights reserved, all wrongs reversed.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS 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.

Implementations of the MD4 algorithm are subject to the following notice:

Copyright © 1990, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD4 Message Digest Algorithm" in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD4 Message Digest Algorithm" in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided “as is” without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

Implementations of the MD5 algorithm are subject to the following notice:

Copyright © 1990, RSA Data Security, Inc. All rights reserved.

License to copy and use this software is granted provided that it is identified as the “RSA Data Security, Inc. MD5 Message-Digest Algorithm” in all material mentioning or referencing this software or this function.

License is also granted to make and use derivative works provided that such works are identified as “derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm” in all material mentioning or referencing the derived work.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided “as is” without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

The following notice applies to `src/lib/crypto/crypto_tests/t_mddriver.c`:

Copyright © 1990-2, RSA Data Security, Inc. Created 1990. All rights reserved.

RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided “as is” without express or implied warranty of any kind.

These notices must be retained in any copies of any part of this documentation and/or software.

Portions of `src/lib/krb5` are subject to the following notice:

Copyright © 1994 CyberSAFE Corporation.

Copyright 1990,1991,2007,2008 by the Massachusetts Institute of Technology.

All Rights Reserved.

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 M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. Furthermore if you modify this software you must label your software as modified software and not distribute it in such a fashion that it might be confused with the original M.I.T. software. Neither M.I.T., the Open Computing Security Group, nor CyberSAFE Corporation make any representations about the suitability of this software for any purpose. It is provided “as is” without express or implied warranty.

Portions contributed by PADL Software are subject to the following license:

Copyright (c) 2011, PADL Software Pty Ltd. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of PADL Software nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY PADL SOFTWARE AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL PADL SOFTWARE 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 bundled libev source code is subject to the following license:

All files in libev are Copyright (C)2007,2008,2009 Marc Alexander Lehmann.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED 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.

Alternatively, the contents of this package may be used under the terms of the GNU General Public License ("GPL") version 2 or any later version, in which case the provisions of the GPL are applicable instead of the above. If you wish to allow the use of your version of this package only under the terms of the GPL and not to allow others to use your version of this

file under the BSD license, indicate your decision by deleting the provisions above and replace them with the notice and other provisions required by the GPL in this and the other files of this package. If you do not delete the provisions above, a recipient may use your version of this file under either the BSD or the GPL.

Table of Contents

1	Introduction	1
1.1	What is Kerberos and How Does it Work?	1
1.2	Why Should I use Kerberos?	1
1.3	Please Read the Documentation	1
1.4	Overview of This Guide	2
2	Realm Configuration Decisions	3
2.1	Kerberos Realms	3
2.2	Mapping Hostnames onto Kerberos Realms	3
2.3	Ports for the KDC and Admin Services	4
2.4	Slave KDCs	4
2.5	Hostnames for the Master and Slave KDCs	4
2.6	Database Propagation	6
3	Building Kerberos V5	7
3.1	Organization of the Source Directory	7
3.1.1	The appl Directory	7
3.1.2	The clients Directory	7
3.1.3	The gen-manpages Directory	8
3.1.4	The include Directory	8
3.1.5	The kadmin Directory	8
3.1.6	The kdc Directory	8
3.1.7	The krb524 Directory	8
3.1.8	The lib Directory	8
3.1.9	The prototype Directory	8
3.1.10	The slave Directory	9
3.1.11	The util Directory	9
3.2	Build Requirements	9
3.3	Unpacking the Sources	9
3.4	Doing the Build	10
3.4.1	Building Within a Single Tree	10
3.4.2	Building with Separate Build Directories	10
3.4.3	Building Using 'lndir'	10
3.5	Installing the Binaries	11
3.6	Testing the Build	11
3.6.1	The DejaGnu Tests	12
3.6.2	The KADM5 Tests	12
3.7	Options to Configure	12
3.8	'osconf.h'	15
3.9	Shared Library Support	16
3.10	Operating System Incompatibilities	16
3.10.1	AIX	16

3.10.2	Alpha OSF/1 V1.3	17
3.10.3	Alpha OSF/1 V2.0	17
3.10.4	Alpha OSF/1 (Digital UNIX) V4.0	17
3.10.5	BSDI	17
3.10.6	HPUX	17
3.10.7	Solaris versions 2.0 through 2.3	18
3.10.8	Solaris 2.X	18
3.10.9	Solaris 9	18
3.10.10	SGI Irix 5.X	19
3.10.11	Ultrix 4.2/3	19
3.11	Using 'Autoconf'	19

4 Installing Kerberos V5 21

4.1	Installing KDCs	21
4.1.1	Install the Master KDC	21
4.1.1.1	Edit the Configuration Files	21
4.1.1.2	krb5.conf	21
4.1.1.3	kdc.conf	23
4.1.1.4	Create the Database	23
4.1.1.5	Add Administrators to the Acl File	24
4.1.1.6	Add Administrators to the Kerberos Database	26
4.1.1.7	Create a kadmind Keytab (optional)	26
4.1.1.8	Start the Kerberos Daemons on the Master KDC	27
4.1.2	Install the Slave KDCs	27
4.1.2.1	Create Host Keys for the Slave KDCs	27
4.1.2.2	Extract Host Keytabs for the KDCs	28
4.1.2.3	Set Up the Slave KDCs for Database Propagation ...	28
4.1.3	Back on the Master KDC	29
4.1.3.1	Propagate the Database to Each Slave KDC	29
4.1.4	Finish Installing the Slave KDCs	29
4.1.4.1	Create Stash Files on the Slave KDCs	30
4.1.4.2	Start the krb5kdc Daemon on Each KDC	30
4.1.5	Add Kerberos Principals to the Database	30
4.1.6	Limit Access to the KDCs	30
4.1.7	Switching Master and Slave KDCs	30
4.1.8	Incremental Database Propagation	31
4.1.8.1	Sun/MIT Incremental Propagation Differences	32
4.2	Installing and Configuring UNIX Client Machines	33
4.2.1	Client Programs	33
4.2.2	Client Machine Configuration Files	33
4.2.2.1	Mac OS X Configuration	34
4.3	UNIX Application Servers	35
4.3.1	The Keytab File	35
4.3.2	Some Advice about Secure Hosts	36

5 Upgrading Existing Kerberos V5 Installations 37

5.1	Upgrading to Triple-DES Encryption Keys	37
-----	---	----

6	Bug Reports for Kerberos V5	39
Appendix A	Copyright	43

