Full credit is given to the above companies including the OS that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'kerberos.7'

$ man kerberos.7

KERBEROS(7) MIT Kerberos KERBEROS(7)

NAME

kerberos - Overview of using Kerberos

DESCRIPTION

The Kerberos system authenticates individual users in a network envi?

ronment. After authenticating yourself to Kerberos, you can use Ker?

beros-enabled programs without having to present passwords or certifi?

cates to those programs.

If you receive the following response from kinit(1):

kinit: Client not found in Kerberos database while getting initial cre?

dentials

you haven't been registered as a Kerberos user. See your system admin?

istrator.

A Kerberos name usually contains three parts. The first is the pri?

mary, which is usually a user's or service's name. The second is the

instance, which in the case of a user is usually null. Some users may

have privileged instances, however, such as root or admin. In the case

of a service, the instance is the fully qualified name of the machine on which it runs; i.e. there can be an ssh service running on the machine ABC (ssh/ABC@REALM), which is different from the ssh service running on the machine XYZ (ssh/XYZ@REALM). The third part of a Kerberos name is the realm. The realm corresponds to the Kerberos service providing authentication for the principal. Realms are conventionally all-uppercase, and often match the end of hostnames in the realm (for instance, host01.example.com might be in realm EXAMPLE.COM).

When writing a Kerberos name, the principal name is separated from the instance (if not null) by a slash, and the realm (if not the local realm) follows, preceded by an "@" sign. The following are examples of valid Kerberos names:

```
   david
   jennifer/admin
   joeuser@BLEEP.COM
   cbrown/root@FUBAR.ORG
```

When you authenticate yourself with Kerberos you get an initial Kerberos ticket. (A Kerberos ticket is an encrypted protocol message that provides authentication.) Kerberos uses this ticket for network utilities such as ssh. The ticket transactions are done transparently, so you don't have to worry about their management.

Note, however, that tickets expire. Administrators may configure more privileged tickets, such as those with service or instance of root or admin, to expire in a few minutes, while tickets that carry more ordinary privileges may be good for several hours or a day. If your login session extends beyond the time limit, you will have to re-authenticate yourself to Kerberos to get new tickets using the kinit(1) command.

Some tickets are renewable beyond their initial lifetime. This means that kinit -R can extend their lifetime without requiring you to re-authenticate.

If you wish to delete your local tickets, use the kdestroy(1) command.

Kerberos tickets can be forwarded. In order to forward tickets, you must request forwardable tickets when you kinit. Once you have forwardable tickets, most Kerberos programs have a command line option to forward them to the remote host. This can be useful for, e.g., running kinit on your local machine and then sshing into another to do work. Note that this should not be done on untrusted machines since they will then have your tickets.

ENVIRONMENT VARIABLES

Several environment variables affect the operation of Kerberos-enabled programs. These include:

KRB5CCNAME

Default name for the credentials cache file, in the form TYPE:residual. The type of the default cache may determine the availability of a cache collection. FILE is not a collection type; KEYRING, DIR, and KCM are.

If not set, the value of default_ccache_name from configuration files (see KRB5_CONFIG) will be used. If that is also not set, the default type is FILE, and the residual is the path /tmp/krb5cc_*uid*, where uid is the decimal user ID of the user.

KRB5_KTNAME

Specifies the location of the default keytab file, in the form TYPE:residual. If no type is present, the FILE type is assumed and residual is the pathname of the keytab file. If unset, FILE:/etc/krb5.keytab will be used.

KRB5_CONFIG

Specifies the location of the Kerberos configuration file. The default is /etc/krb5.conf. Multiple filenames can be specified, separated by a colon; all files which are present will be read.

KRB5_KDC_PROFILE

Specifies the location of the KDC configuration file, which con? tains additional configuration directives for the Key Distribu? tion Center daemon and associated programs. The default is /var/kerberos/krb5kdc/kdc.conf.

KRB5RCACHENAME

(New in release 1.18) Specifies the location of the default re? play cache, in the form type:residual. The file2 type with a pathname residual specifies a replay cache file in the version-2 format in the specified location. The none type (residual is ignored) disables the replay cache. The dfl type (residual is ignored) indicates the default, which uses a file2 replay cache in a temporary directory. The default is dfl:.

KRB5RCACHETYPE

Specifies the type of the default replay cache, if KRB5RCACHENAME is unspecified. No residual can be specified, so none and dfl are the only useful types.

KRB5RCACHEDIR

Specifies the directory used by the dfl replay cache type. The default is the value of the TMPDIR environment variable, or /var/tmp if TMPDIR is not set.

KRB5_TRACE

Specifies a filename to write trace log output to. Trace logs can help illuminate decisions made internally by the Kerberos libraries. For example, env KRB5_TRACE=/dev/stderr kinit would send tracing information for kinit(1) to /dev/stderr. The de?fault is not to write trace log output anywhere.

KRB5_CLIENT_KTNAME

Default client keytab file name. If unset, FILE:/var/ker?beros/krb5/user/%{euid}/client.keytab will be used).

KPROP_PORT

kprop(8) port to use. Defaults to 754.

GSS_MECH_CONFIG

Specifies a filename containing GSSAPI mechanism module configu?ration. The default is to read /etc/gss/mech and files with a .conf suffix within the directory /etc/gss/mech.d.

Most environment variables are disabled for certain programs, such as login system programs and setuid programs, which are designed to be se?cure when run within an untrusted process environment.

SEE ALSO

kdestroy(1), kinit(1), klist(1), kswitch(1), kpasswd(1), ksu(1), krb5.conf(5), kdc.conf(5), kadmin(1), kadmind(8), kdb5_util(8), krb5kdc(8)

BUGS

AUTHORS

Steve Miller, MIT Project Athena/Digital Equipment Corporation

Clifford Neuman, MIT Project Athena

Greg Hudson, MIT Kerberos Consortium

Robbie Harwood, Red Hat, Inc.

HISTORY

The MIT Kerberos 5 implementation was developed at MIT, with contributions from many outside parties. It is currently maintained by the MIT Kerberos Consortium.

RESTRICTIONS

Copyright 1985, 1986, 1989-1996, 2002, 2011, 2018 Masachusetts Institute of Technology

AUTHOR

MIT

COPYRIGHT

1985-2022, MIT