NAME

krb5_keytab - kerberos keytab management client

SYNOPSIS

krb5_keytab [**-AFUZfvw**] [**-p** *user*] [**-L** *lib*] [**-X** *xrealm*] [**-W** *winprinc*] [*princ* ...]
krb5_keytab -c [**-AFUZfv**] [**-p** *user*] [**-L** *lib*] [*princ* ...]
krb5_keytab -Ug [**-v**] [**-p** *user*] [*princ* ...]
krb5_keytab -l [**-v**] [**-p** *user*]
krb5_keytab -q [**-Uv**] [**-p** *user*] [*princ* ...]
krb5_keytab -t [**-Uv**] [**-p** *user*] [**-L** *lib*] [*princ* ...]

DESCRIPTION

krb5_keytab will create, fetch or rotate the Kerberos keys and ensure that the correct keys are installed.

In general, **krb5_keytab** should be called in the start scripts of the server that requires the keytab. If run as the user in question, it requires no arguments and will simply fetch all of the keys necessary. **krb5_keytab** can also be run as root but in this case, it is necessary to specify the **-p** *user* or the **-A** flag.

The *princ* arguments are Kerberos principals with the following completion rules:

1. If the realm is missing, then the default realm is appended.

2. If the instance is missing then all hostname instances are appended, i.e.: if one supplies ''foo'' or ''foo@REALM'' on a machine with hostnames bar and baz then it will expand to the principals foo/bar@REALM and foo/baz@REALM. Note that this will not append the cluster names as not all services are expected to run on all cluster addresses. If **-U** is specified then the username is prepended to the hostname.

krb5_keytab will then construct a connexion to a KDC and request the appropriate keys, creating or rotating them if necessary. If the keys need to be created or rotated then **krb5_keytab** will need to connect to the master KDC.

krb5_keytab is designed to be run in the start logic of applications that require keytabs. There is no attempt made to persist keys across rebuilds or reboots and in fact, the assumption is that if a host is rebuilt and does not request a service key again then said service key is no longer being used and is eligible for removal.

krb5_keytab creates keys that are compatible with the Kerberos library specified by the optional **-L** *library* argument. The library should be specified as the Kerberos libraries that are used by the server which will use the keytab. If an incorrect library is specified then the keys created or generated may not

be compatible with your server. The default library is specified in the configuration file. Specifying certain legacy libraries requires administrative privileges.

Please note that although, **krb5_keytab** will create stronger keys, it will not [currently] modify existing keys if called on a machine which already has keys provisioned for a service. It will simply retrieve the existing keys from the KDC.

The options are as follows:

-A specifies that **krb5_keytab** will fetch the host keys for the machine. This requires that **krb5_keytab** is run as root and will prompt for Kerberos admin credentials.

-F will force the keytab to be rewritten and also implies **-f**.

-L *lib* specifies that the Kerberos libraries that the server uses are *lib*. This information will be used to determine what encryption types are supported, so it is essential that this is correct. The default is mitkrb5/1.4 when creating new keys. When changing keys, the current encryption types will be preserved unless the **-L** *library* option is specified.

-U prepends the username to the hostname used to construct default instances where they are not specified. I.e. host.example.com becomes username.host.example.com.

-W *winprinc* specifies that **krb5_keytab** will use the Windows principal *winprinc* as a client when contacting the KDC. The principal must be fully qualified including the realm and must exist in */etc/krb5.keytab*.

-X *xrealm* specifies that **krb5_keytab** will use the realm *xrealm* to construct the client principal in preference to the realm of the requested principal.

-Z operate on a local Kerberos DB rather than talking to the network. This mode is useful for startup scripts on the master KDC but it should only be used on the master KDC and not on any slave KDCs as that would cause the Kerberos DB on that slave to be out of synchronisation with the master with perhaps less than desired results.

-c change the keys.

-f force contact with a KDC. By default, **krb5_keytab** will not contact a KDC if the keytab tests correctly. This flag overrides this behaviour. Do not use this flag in any automated use, it is designed for human intervention only.

-g will cause **krb5_keytab** to read the current keytab and issue a list of commands that would generate an equivalent keytab under certain assumptions.

-l lists the current keytab. This flag is for administrators only.

-p *user* the user for which keys are to be obtained. This option is only relevant for administrative users.

-q will describe the keytab in a human readable format.

-t tests if the specified keys in are compatible with the current library version as specified by the **-L** *library* option.

-v increase the verbose level.

-w will look for a Windows principal during a xrealm bootstrap operation. This flag must be specified in conjunction with **-X** *xrealm*. The Windows principal selected will be the first valid principal which is in the realm *xrealm* containing only one component which is terminated with a dollar sign.

EXIT STATUS

krb5_keytab exits 0 on success and >0 if an error occurred.

FILES

/etc/krb5_keytab.conf is the configration file for krb5_keytab.

EXAMPLES

If running as an ID, to create a keytab for the default version of the Kerberos libraries containing all possible service principals configured for the host:

```
$ krb5_keytab
```

If only a certain principal needs to be created:

```
$ krb5_keytab service/hostname
```

To fetch Kerberos credentials host/<hostname>@BAR.EXAMPLE.COM using host/<hostname>@FOO.EXAMPLE.COM:

```
$ krb5_keytab -X FOO.EXAMPLE.COM host@BAR.EXAMPLE.COM
```

To test if a keytab contains keys that are compatible with a certain version of the Kerberos libraries:

```
$ krb5_keytab -t -L mitkrb5/1.3
```

Or to test if just a single principal has keys that are compatible with a certain version of the Kerberos libraries:

```
$ krb5_keytab -t -L sunjdk/1.6 HTTP/host.example.com
```

To generate a list of commands that would create a functionally equivalent keytab for a particular user:

```
$ krb5_keytab -g -p user
```

SEE ALSO

knc(1), krb5_keytabd(8).

BUGS

Key rotation is not yet implemented.